รายงานฉบับสมบูรณ์

โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนา
กิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ

# โครงการ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1)

สัญญารับทุนเลขที่ B 64-3-(2)-004

ดร. มลธิดา ภัทรนันทกุล (หัวหน้าโครงการ)
ดร. ชาลี วรกุลพิพัฒน์
ดร. กลิกา สุขสมบูรณ์
ดร. เฉลิมพล ชาญศรีภิญโญ
นายติณณ์ธิษณ์ ใจกล้า
นายอภิวัฒน์ จันทวิบูลย์
นางสาวปารย์ ศิริมนพร
นายธฤต ทองเปลว

เมษายน พ.ศ. 2568

ได้รับทุนอุดหนุนจาก
กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

**รายงานฉบับสมบูรณ์**

**โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนา**
**กิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ**

# โครงการ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1)

**สัญญารับทุนเลขที่ B 64-3-(2)-004**


**คณะผู้วิจัย**

| | | |
|---|---|---|
| นางสาว มลธิดา ภัทรนันทกุล | นักวิจัย (ป.เอก) | หัวหน้าโครงการ |
| นาย ชาลี วรกุลพิพัฒน์ | นักวิจัย (ป.เอก) | ผู้ร่วมโครงการ |
| นางสาว กลิกา สุขสมบูรณ์ | นักวิจัย (ป.เอก) | ผู้ร่วมโครงการ |
| นาย เฉลิมพล ชาญศรีภิญโญ | นักวิจัย (ป.เอก) | ผู้ร่วมโครงการ |
| นาย ติณณ์ธิษณ์ ใจกล้า | นักวิจัย (ป.โท) | ผู้ร่วมโครงการ |
| นาย อภิวัฒณ์ จันทวิบูลย์ | นักวิจัย (ป.โท) | ผู้ร่วมโครงการ |
| นางสาว ปารย์ ศิริมนพร | นักวิจัย (ป.โท) | ผู้ร่วมโครงการ |
| นาย ธฤต ทองเปลว | นักวิจัย (ป.ตรี) | ผู้ร่วมโครงการ |


**เมษายน พ.ศ. 2568**


**ได้รับทุนอุดหนุนจาก**
**กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ**
**(สำนักงาน กสทช.)**

# บทสรุปผู้บริหาร

โครงการ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1)

เมษายน พ.ศ. 2568

ในยุคดิจิทัลที่โครงสร้างพื้นฐานด้าน ICT มีความซับซ้อนและต้องการความยืดหยุ่นสูง การประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization technology) กลายเป็นเทคโนโลยีที่มีบทบาทสำคัญต่อการสร้างระบบนิเวศดิจิทัลที่เข้มแข็งต่อการขับเคลื่อนประเทศไปสู่สภาพแวดล้อม 5G ในอนาคต เพื่อเพิ่มประสิทธิภาพการบริหารและจัดการเครือข่ายอย่างยั่งยืน สำหรับในบริบทประเทศไทย แม้ว่าเทคโนโลยีดังกล่าวฯ นำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการบริหารเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่าย อีกทั้งคุณประโยชน์มากมายต่อการสร้างโอกาสทางธุรกิจใหม่ๆ และเป็นเสาหลักที่สำคัญต่อการวางรากฐานการขับเคลื่อนโครงสร้างพื้นฐาน ICT ของประเทศไปสู่ยุคดิจิทัลอย่างยั่งยืนในอนาคต ในอีกมุมหนึ่ง องค์ความรู้เกี่ยวกับเทคโนโลยีนี้ในประเทศไทยแทบจะมีน้อยมาก โดยเฉพาะอย่างยิ่งผู้ให้บริการเครือข่ายส่วนใหญ่ต่างหันมาพึ่งพาการนำเข้าเทคโนโลยีและนวัตกรรมจากต่างประเทศที่มีต้นทุนสูง ปัจจัยเหล่านี้นำมาซึ่งการเผชิญต่อความท้าทายด้านองค์ความรู้ ความพร้อมของบุคลากร และการสร้างกรณีศึกษาที่ใช้สามารถใช้งานจริงและปรับใช้ได้อย่างเหมาะสมกับองค์กร ผลกระทบดังล่าวฯ ส่งผลให้ประเทศไทยไม่สามารถก้าวข้ามพ้นจากการเป็นประเทศที่กำลังพัฒนาสู่การเติบโตอย่างยั่งยืนและระยะยาว

โครงการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริงเพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) จึงถือเป็นโครงการนำร่องที่สำคัญที่แสดงให้เห็นถึงการขับเคลื่อนโครงสร้างพื้นฐานเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบดั้งเดิม (Traditional networking model) ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดด้วยซอฟต์แวร์ (Software-driven infrastructure and services) นำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการให้บริการเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่ายรูปแบบใหม่ ที่สามารถช่วยลดข้อจำกัดการให้บริการเครือข่ายแบบดั้งเดิมได้ อาทิเช่น (1) การลดค่าใช้จ่ายด้านการลงทุนและการดำเนินงาน (2) ความสามารถในการตอบสนองการให้บริการและการจำลองเน็ตเวิร์คเซอร์วิสได้อย่างรวดเร็ว (3) ประสิทธิภาพในการประมวลผลและการบริหารทรัพยากร (4) ความยืดหยุ่นในการจัดสรรหรือขยายทรัพยากร ตลอดจน (5) การให้อิสระต่อการตั้งค่า ปรับใช้ และจำลองเน็ตเวิร์คเซอร์วิสต่างๆ ตามความต้องการเฉพาะของแต่ละองค์กรได้

โดยโครงการนี้มีวัตถุประสงค์หลัก ดังนี้

1. เชิงการวิจัยและพัฒนา: เพื่อวิจัยและพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง สนับสนุนให้เกิดการสร้างองค์ความรู้และนวัตกรรมใหม่ๆ ด้วยการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ให้มีขีดความสามารถในการจัดสรรและบริหารทรัพยากรได้อย่างยืดหยุ่นและอัตโนมัติ และสามารถให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน โดยเฉพาะ

อย่างยิ่ง เน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (security related VNFaaS) สำหรับการรักษาความปลอดภัยไซเบอร์ให้กับองค์กร

2. เชิงเศรษฐกิจ: เพื่อส่งเสริมให้เกิดการลดสัดส่วนการพึ่งพาเทคโนโลยีจากต่างประเทศ โดยการใช้องค์ความรู้ เครื่องมือ และบุคลากรที่มีในการพัฒนานวัตกรรมขึ้นมาใช้เอง สร้างโอกาสให้การเพิ่มขีดความสามารถด้านการแข่งขันและการพึ่งพาตัวเอง
3. เชิงสังคม: เพื่อส่งเสริมการต่อยอดและขยายผลนวัตกรรมไปสู่การใช้งานจริงอย่างแพร่หลายในวงกว้าง ไปสู่ภาคส่วนต่างๆ ทั้งภาคอุตสาหกรรม ภาคอุดมศึกษา ภาครัฐ และภาคเอกชน

สำหรับขอบเขตการดำเนินงานของโครงการจะแบ่งออกเป็น 4 ระยะสำคัญ ดังนี้

1. ศึกษาเชิงลึกด้านเทคโนโลยีที่เกี่ยวข้อง

ในระยะนี้ ทีมวิจัยให้ความสำคัญต่อการศึกษาเทคโนโลยีที่เกี่ยวข้อง ได้แก่ เทคโนโลยีการจำลองฟังก์ชันเครือข่ายเสมือน (Network Functions Virtualization: NFV) การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) และการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) เพื่อวิเคราะห์ถึงสถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้องอันเป็นประโยชน์ต่อการสร้างองค์ความรู้ในเทคโนโลยีการจำลองระบบเสมือนจริง

จากผลการศึกษาพบว่า การเกิดขึ้นของเทคโนโลยีดังกล่าวฯ นำมาซึ่งการเปลี่ยนแปลงมากมายต่อการจัดการโครงสร้างพื้นฐานเครือข่าย โดยมีจุดเด่นหลักๆ คือ การลดลงด้านต้นทุนและการดำเนินการ ความรวดเร็วในการให้บริการ ความยืดหยุ่นในการปรับขยายทรัพยากร รวมถึงการให้อิสระต่อการเลือกใช้งานเน็ตเวิร์คเซอร์วิสแต่ละประเภทเพื่อให้มีความเหมาะสมกับบริบทของแต่ละองค์กร ในอีกมุมหนึ่ง การเกิดขึ้นของเทคโนโลยีดังกล่าวฯ มีทั้งคุณประโยชน์ และความเสี่ยงด้านความมั่นคงปลอดภัยไซเบอร์ที่ต้องคำนึงถึง โดยภัยคุกคามสามารถเกิดขึ้นได้ทุกระดับชั้นของสถาปัตยกรรม ได้แก่ (1) ชั้นโครงสร้างพื้นฐาน (Infrastructure layer): ที่มีความเสี่ยงจากเครื่องเสมือนจริง (Virtual Machine: VM) เซิร์ฟเวอร์ และอุปกรณ์เครือข่าย; (2) ชั้นฟังก์ชันเครือข่ายเสมือน (VNF layer): ความเสี่ยงจากซอฟต์แวร์ที่ใช้บริการเน็ตเวิร์คเสมือน เช่น firewall, DPI, NAT เป็นต้น; และ (3) ชั้นจัดการและประสานงาน (Management and orchestration layer): ความเสี่ยงจากการโจมตีระบบควบคุมหรือความล้มเหลวในการกำหนดนโยบาย ด้วยเหตุนี้ การเตรียมความพร้อมและการวางมาตรการด้านความปลอดภัยในทุกระดับชั้นจึงเป็นสิ่งจำเป็นอย่างยิ่ง เพื่อป้องกันและลดโอกาสของการถูกโจมตีจากภัยคุกคามทางไซเบอร์

2. พัฒนาและพิสูจน์แนวคิด (Proof of Concept)

ในระยะนี้ ทีมวิจัยให้ความสำคัญกับการออกแบบและพัฒนาแพลตฟอร์มต้นแบบ (Prototype) สำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือน โดยเริ่มจากดำเนินการศึกษาโครงการที่เกี่ยวข้องกับการพัฒนาแพลตฟอร์ม ดังเช่น โครงการ T-NOVA, ANASTACIA, OSM, ONAP, PALANTIR, และ KDDI & NEC เพื่อทำความเข้าใจหลักการออกแบบของแต่ละโครงการ แนวคิดการออกแบบสถาปัตยกรรมของระบบ คุณสมบัติและคอมโพเนนท์ต่างๆ เพื่อประโยชน์ต่อการนำมาปรับใช้และคัดเลือกเครื่องมือ (open source tools) ที่เหมาะสมต่อการพัฒนาแพลตฟอร์มต่อไป

สำหรับแนวทางการออกแบบสถาปัตยกรรมแพลตฟอร์ม ทีมวิจัยให้ความสำคัญกับกรณีศึกษาด้านความปลอดภัย (security use case) เป็นหลัก โดยมุ่งเน้นในการพัฒนาระบบให้มีขีดความสามารถในการจัดสรรทรัพยากรอัตโนมัติ การควบคุมระบบ การจัดเตรียมและการให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย ดังเช่น Deep Packet Inspection (e.g., Suricata), Intrusion Detection System (e.g., Snort) และ Traffic analyzer (e.g., Zeek) การวิเคราะห์ความเสี่ยงด้านภัยคุกคามและความผิดปกติของเครือข่าย เพื่อเป็นฐานในการขยายผลสู่การใช้งานจริง โดยแนวคิดของการออกแบบจะแบ่งการทำงานออกเป็น 3 ระดับชั้น ดังนี้ (1) System plane ส่วนเชื่อมต่อระหว่างเครือข่ายผู้รับบริการกับแพลตฟอร์ม เพื่อทำหน้าที่รวบรวมข้อมูล logs การทำงานต่างๆ ที่สนใจ และส่งต่อไปยังแพลตฟอร์มเพื่อทำการตรวจสอบความเสี่ยงด้านความปลอดภัยที่อาจจะเกิดขึ้น; (2) Orchestration and enforcement plane ทำหน้าที่ในการบริหารจัดการทรัพยากรและจำลองเน็ตเวิร์คเซอร์วิส รวมไปถึงการประสานการทำงานต่างๆ ภายในระบบให้มีความเป็นอัตโนมัติ อีกทั้งทำหน้าที่ในการตรวจสอบความเสี่ยงหรือช่องโหว่ที่อาจจะเกิดขึ้นด้วยการวิเคราะห์ข้อมูล logs ที่ได้รับมา; และ (3) User plane เพื่ออำนวยความสะดวกให้แก่ผู้ดูแลระบบในการติดต่อกับระบบสำหรับสร้างและออกแบบพิมพ์เขียวของแต่ละเน็ตเวิร์คเซอร์ และอำนวยความสะดวกให้แก่ผู้ใช้งานในการตรวจสอบและเรียกดูผลวิเคราะห์ภัยคุกคามผ่าน dashboard ในรูปแบบกราฟแสดงผลต่างๆ

3. ทดสอบและประเมินผล

ในระยะนี้จะเกี่ยวข้องกับการทดสอบต้นแบบเพื่อประเมินประสิทธิภาพของระบบ โดยแบ่งเกณฑ์การวัดตามคุณสมบัติ 3 ด้าน ดังนี้ (1) การทำงานเชิงหน้าที่ เพื่อตรวจสอบความถูกต้องของการทำงานต่างๆ ตั้งแต่ระยะเริ่มต้น (pre-configuration and deployment time) ว่าดำเนินการอย่างไร; (2) ด้านการใช้งาน เพื่อตรวจสอบความถูกต้องของข้อมูลที่นำมาแสดงผลผ่าน dashboard ประกอบด้วย ข้อมูลที่แสดงสถานะการทำงานของระบบหลังบ้าน (เช่น สถานะการใช้งานทรัพยากร สถานะการทำงานของเครื่อง VM) ข้อมูลแสดงสถานะการแจ้งเตือนของระบบ และข้อมูลแสดงผลวิเคราะห์ภัยคุกคาม; (3) ด้านประสิทธิภาพ โดยมีเกณฑ์ชี้วัด 3 ด้านหลัก ดังนี้ เวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิส (deployment time) เวลาที่ใช้ในการตั้งค่าข้อมูลเบื้องต้น (configuration time) เวลาแฝงโดยรวมที่ใช้จากต้นทางถึงปลายทาง (end-to-end latency) จากผลการทดสอบพบว่าระยะเวลาเฉลี่ยที่ใช้ในการวิเคราะห์และประมวลผลภัยคุกคามในแต่ละเน็ตเวิร์คเซอร์วิสอยู่ในเกณฑ์ที่น่าพอใจ กล่าวคือ 0.02, 3.33, และ 9.91 วินาทีสำหรับการประมวลผลของ Suricata, Snort และ Zeek ตามลำดับ

4. ถ่ายทอดองค์ความรู้และขยายนวัตกรรมไปสู่การใช้งานจริง

ระยะนี้มุ่งเน้นการเผยแพร่ประชาสัมพันธ์ผลงาน การถ่ายทอดองค์ความรู้จากงานวิจัยไปสู่ผู้มีส่วนเกี่ยวข้อง และสนับสนุนให้เกิดการต่อยอดและขยายผลนวัตกรรมไปสู่การใช้งานจริงอย่างแพร่หลายในวงกว้างไปสู่ภาคส่วนต่างๆ ทั้งภาคอุตสาหกรรม ภาคอุดมศึกษา ภาครัฐ และภาคเอกชน เพื่อสร้างผลกระทบเชิงบวกต่อการสนับสนุนนวัตกรรมที่พัฒนาขึ้นใช้เองในประเทศระยะยาว โดยทางทีมวิจัยได้กำหนดแนวทางการดำเนินงานเอาไว้ ดังนี้ (1) การเข้าร่วมและจัดกิจกรรมสัมมนาเชิงวิชาการ

จำนวน 2 ครั้ง ได้แก่ กิจกรรมสัมมนาเพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ จัดโดยสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) เพื่อแลกเปลี่ยนองค์ความรู้และสร้างเครือข่ายความร่วมมือระหว่างนักวิจัย หน่วยงานรัฐ และภาคเอกชน และกิจกรรมเสวนาในหัวข้อซีโร่ทัชเซอร์วิส ผลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ ในงานการประชุมวิชาการประจำปีสวทช. ซึ่งมุ่งเน้นให้ผู้เข้าร่วมสามารถเข้าถึงแนวคิดการออกแบบและการทำงานแพลตฟอร์มที่พัฒนาขึ้น; (2) การออกบูธจัดแสดงนิทรรศการ จำนวน 2 ครั้ง ได้แก่ งานประชุมวิชากรและนิทรรศการเนคเทค ประจำปี 2567 ซึ่งเป็นเวทีสำคัญในการนำเสนอผลงานวิจัยให้กับบุคคลทั่วไป หน่วยงานราชการ และภาคเอกชน และงานจัดแสดงนิทรรศการ BTFP Showcase 2024 จัดโดยกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนา เพื่อประโยชน์สาธารณะ (กทปส.) ซึ่งเป็นเวทีที่เปิดโอกาสให้ทีมวิจัยได้แสดงศักยภาพของแพลตฟอร์มต่อสาธารณชนและผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้อง; และ (3) การสร้างความร่วมมือกับหน่วยงานที่เกี่ยวข้อง จำนวน 2 หน่วยงาน ได้แก่ สถาบันดนตรีกัลยานิวัฒนา และสำนักงานสาธารณสุขลพบุรี โดยมีเป้าหมายเพื่อต่อยอดและขยายผลการใช้งานแพลตฟอร์มสู่การใช้งานจริง

นอกเหนือจากกิจกรรมถ่ายทอดองค์ความรู้ โครงการฯ ยังประสบความสำเร็จอีกขั้นด้วยการได้รับรางวัล “ความเป็นเลิศด้านความคิดสร้างสรรค์” ประเภทต้นแบบผลิตภัณฑ์หรือบริการ (Prototype) ด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ จากสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) เวทีการประกวดดังกล่าวถือเป็นเวทีระดับชาติที่ได้รับการยอมรับอย่างกว้างขวาง และมีบทบาทสำคัญในการส่งเสริม สนับสนุน และเปิดโอกาสให้ผลงานวิชาการด้านความมั่นคงปลอดภัยไซเบอร์ได้รับการต่อยอดสู่การใช้งานจริงทั้งในภาครัฐและเอกชน การได้รับรางวัลนี้ไม่เพียงสะท้อนถึง ศักยภาพของนักวิจัยไทยในการพัฒนาเทคโนโลยีด้านไซเบอร์ที่ทันสมัยและมีความคิดสร้างสรรค์ หากยังตอกย้ำความสอดคล้องของผลงานกับบริบทและความต้องการของประเทศ ตลอดจนเป็นแรงผลักดันสำคัญในการขยายผลสู่องค์กรเป้าหมาย และสนับสนุนการพึ่งพาเทคโนโลยีภายในประเทศอย่างยั่งยืนในอนาคต

(บทคัดย่อ ภาษาไทย จำนวน 1 หน้ากระดาษ)

**โครงการ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1)**

**ดร. มลธิดา ภัทรนันทกุล**

**เมษายน พ.ศ. 2568**

ในยุคที่เทคโนโลยีดิจิทัลมีการพัฒนาไปอย่างรวดเร็ว หลายประเทศทั้งในยุโรปและเอเชียต่างให้ความสำคัญกับการผลักดันเทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization technology) ได้แก่ เทคโนโลยีการจำลองฟังก์ชันเครือข่ายเสมือน (Network Functions Virtualization: NFV) การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) การจัดเก็บและประมวลผลข้อมูลบนคลาวด์ (Cloud computing) ไปสู่การใช้งานจริง เนื่องจากเทคโนโลยีเหล่านี้มีบทบาทสำคัญในการผลิกโฉมโครงสร้างพื้นฐานเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบดั้งเดิม (Traditional networking model) ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดด้วยซอฟต์แวร์ (Software-driven infrastructure and services) ซึ่งสอดคล้องกับบริบทของ 5G ในอนาคต (5G services) ดังเช่น การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนสนับสนุนงานด้าน smart city, mission critical application, condition monitoring, network-security monitoring ฯลฯ นำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการให้บริการเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่ายรูปแบบใหม่ที่สามารถช่วยลดข้อจำกัดการให้บริการเครือข่ายแบบดั้งเดิมได้ ทั้งในเรื่องการช่วยลดต้นทุนการลงทุนและการดำเนินการงาน ความรวดเร็วในการตอบสนองต่อการให้บริการและการจำลองเน็ตเวิร์คเซอร์วิสอย่างทันถ่วงที ประสิทธิในการประมวลผล ความยืดหยุ่นในการบริหารและจัดสรรทรัพยากร ตลอดจนการให้อิสระต่อการกำหนดการตั้งค่าและปรับใช้งานเน็ตเวิร์คเซอร์วิสประเภทต่างๆ เพื่อให้มีความสอดคล้องตามบริบทเฉพาะของแต่ละองค์กร (on-demand network services) อย่างไรก็ตาม สำหรับบริบทของประเทศไทย เส้นทางการพัฒนานวัตกรรมที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริงพบว่ายังอยู่ในช่วงเริ่มต้น ปัญหาหลัก คือ การเผชิญกับความท้าทายด้านองค์ความรู้ ความพร้อมของบุคลากร และการสร้างกรณีศึกษาที่ใช้สามารถใช้งานจริงและเหมาะสมกับองค์กร ด้วยเหตุนี้ โครงการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริงเพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) จึงถือเป็นโครงการนำร่องที่สำคัญ โดยมีเป้าหมายหลักคือ (1) การออกแบบสถาปัตยกรรมระบบเพื่อจัดเตรียมระบบนิเวศเสมือนจริง (management and orchestration framework) และ (2) การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs as a Service: VNFaaS) โดยเฉพาะเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย เช่น Deep Packet Inspection (e.g., Suricata), Intrusion Detection System (e.g., Snort) และ Traffic analyzer (e.g., Zeek) เพื่อให้องค์กรสามารถใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสดังกล่าวฯ ต่อตรวจสอบ ป้องกัน และช่วยรักษาความปลอดภัยภายในองค์กรได้อย่างเหมาะสม โครงการนี้นับเป็นก้าวสำคัญในการวางรากฐานความมั่นคงปลอดภัยไซเบอร์ของประเทศ เป็นการสร้างความพร้อมต่อการจัดเตรียมระบบนิเวศ 5G ในอนาคต อันนำไปสู่การยกระดับขีดความสามารถในการแข่งขัน การพึ่งพาตนเอง และการปรับใช้ประโยชน์จากเทคโนโลยีดิจิทัลได้อย่างยั่งยืน

(บทคัดย่อ ภาษาอังกฤษ จำนวน 1 หน้ากระดาษ)

**The Development of Orchestration Platform for Managing and Providing VNFs as a Service (VNFaaS) Towards Supporting 5G Network (Phase 1)**

**Dr. Montida Pattaranantakul**

**April 2025**

In the era of rapid digital transformation, countries across Europe and Asia are embracing virtualization technologies – such as Network Functions Virtualization (NFV), Software-Defined Networking (SDN), and Cloud Computing, are considered – as key enablers for transitioning from traditional networking model to software-driven infrastructures and services. These technologies are vital for supporting emerging 5G applications, including smart cities, mission-critical applications, condition monitoring, and network security monitoring. They redefine a new paradigm of network service delivery by reducing costs, accelerating deployment, improving processing efficiency, and enabling flexible resource management, and on-demand network services tailored to specific organizational needs. In Thailand, however, the adoption remains at an early stage, hindered challenging by limited expertise, readiness, and practical use cases. To address this, the development of orchestration platform for managing and providing VNFs as a Service (VNFaaS) Towards Supporting 5G Network (Phase 1) is proposed, aiming to lay the foundation for a virtualized 5G ecosystem. In particular, the project focuses on two key objectives: (1) designing a management and orchestration framework to support the efficient, scalable, and automated resource allocation, and (2) delivering Virtual Network Functions as a Service (VNFaaS), particularly security-related services such as Deep Packet Inspection (e.g., Suricata), Intrusion Detection Systems (e.g., Snort), and Traffic Analyzers (e.g., Zeek) – to help organizations strengthen cybersecurity through proactive monitoring and rapid response. This initiative represents a significant step towards enhancing Thailand's cybersecurity resilience, 5G readiness, and long-term digital competitiveness.

**สารบัญ**

**บทสรุปผู้บริหาร**.....1
**บทคัดย่อภาษาไทย**.....5
**บทคัดย่อภาษาอังกฤษ**.....6
**สารบัญภาพ**.....9

**บทที่ 1 -- สาระสำคัญของโครงการ (Project highlight)**.....15
1.1 หลักการและเหตุผล
1.2 วัตถุประสงค์ของโครงการ
1.3 ขอบเขตและวิธีการดำเนินงาน
1.4 ผลที่คาดว่าจะได้รับ
1.5 ตัวชี้วัดความสำเร็จ
1.6 คณะผู้วิจัยในการดำเนินโครงการ
1.7 แผนการดำเนินโครงการ

**บทที่ 2 -- กิจกรรมที่ 1: ดำเนินการศึกษา สำรวจ และรวบรวมข้อมูลที่เกี่ยวข้อง เช่น สถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้อง อันเป็นประโยชน์ต่อการสร้างองค์ความรู้และต่อยอดในเทคโนโลยีการจำลองระบบเสมือนจริงเชิงลึก**.....23
2.1 บทนำ
2.2 ข้อจำกัดของโครงสร้างเครือข่ายแบบดั้งเดิม
2.3 เทคโนโลยีที่เกี่ยวข้อง
2.4 ช่องโหว่และภัยคุกคาม
2.5 แนวทางป้องกัน
2.6 บทสรุป

**บทที่ 3 -- กิจกรรมที่ 2: ดำเนินการวิจัย พัฒนา และออกแบบสถาปัตยกรรมแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง**.....39
3.1 บทนำ
3.2 โครงการที่เกี่ยวข้องกับการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเสมือนจริงในปัจจุบัน
3.3 ดำเนินการศึกษา สำรวจ และวิจัยความเป็นไปได้ในการเลือกใช้เครื่องมือ opensource software ต่างๆ ที่เกี่ยวข้อง
3.4 วิจัย พัฒนา และออกแบบสถาปัตยกรรมแพลตฟอร์มเชิงภาพรวม
3.5 บทสรุป

**บทที่ 4 -- กิจกรรมที่ 3: ทดสอบประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน**.....86

4.1 บทนำ

4.2 การทำงานเชิงหน้าที่ (funcitonality)

4.3 ด้านการใช้งาน (feasibility)

4.4 ด้านประสิทธิภาพ (performance)

545 ประโยชน์ที่ผู้ใช้งานจะได้รับ

4.6 บทสรุป

**บทที่ 5 -- กิจกรรมที่ 4: ถ่ายทอดความรู้เชิงวิจัยและนวัตกรรม**.....114

5.1 บทนำ

5.2 การเผยแพร่และประชาสัมพันธ์ผลงานผ่านเวทีสัมมนา

5.3 การเผยแพร่และประชาสัมพันธ์ผลงานผ่านออกจัดแสดงนิทรรศการ

5.4 การร่วมหารือ ต่อยอด ขยายผลนวัตกรรมร่วมกับหน่วยงานพันธมิตร

5.5 ความสำเร็จที่ผ่านมา

5.6 บทสรุป

**บรรณานุกรม**.....136

**ภาคผนวก**

**ภาคผนวก ก:** หนังสือเชิญเข้าร่วมและเป็นวิทยากรในกิจกรรมสัมมนา เพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายด้านความมั่นคงปลอดภัยไซเบอร์.....140

**ภาคผนวก ข:** รายละเอียดกิจกรรมเสวนาหัวข้อเรื่อง Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ.....142

**ภาคผนวก ค:** หนังสือเชิญเข้าร่วมงาน BTFP Showcase 2024.....144

**ภาคผนวก ง:** จดหมายขอความอนุเคราะห์ร่วมทดลองใช้ระบบจากสถาบันดนตรีกัลยานิวัฒนา.....148

**ภาคผนวก จ:** จดหมายขอความอนุเคราะห์ร่วมทดลองใช้ระบบจากสำนักงานสาธารณสุขลพบุรี.....149

**ภาคผนวก ฉ:** วารสารวิชาการที่ได้รับการตีพิมพ์ในฐานข้อมูลวารสารระดับนานาชาติ Computer Networks.....150

**ภาคผนวก ช:** รายงานผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสารสำนักงานกสทช.....177

**ภาคผนวก ซ:** การจัดซื้อครุภัณฑ์ (คอมพิวเตอร์ส่วนบุคคล และจอแสดงผล).....195

**ประวัติผู้วิจัย**.....199

**สารบัญภาพ**

**หน้า**


รูปที่ 1.1 แผนผังการแบ่งภาระหน้าที่การดำเนินงานของผู้วิจัย 21

รูปที่ 2.1.1 ผลสำรวจ Gartner Hype Cycle ปี ๒๕๖๓ – ถึงการวิเคราะห์ทิศทางและแนวโน้มของเทคโนโลยีโครงข่าย (enterprise networking) ที่เกิดใหม่และกำลังสร้างความได้เปรียบทางการแข่งขันในตลาดระดับสูงในอีก 5-10 ปีข้างหน้า 24

รูปที่ 2.1.2: : แสดงโครงสร้างภาพรวมของเทคโนโลยี NFV และ SDN กับการทำงานเกื้อหนุนกัน โดยมี (1) ส่วนจัดการและประสานงาน (NFV Management and Orchestration: NFV MANO Stack) ทำหน้าที่จัดสรรทรัพยากรเครือข่าย (network resources) และจำลอง เน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ประเภทต่างๆ เช่น virtual firewalls, virtual switches เป็นต้น และ (2) ส่วนควบคุม (SDN controller) ทำหน้าที่กำหนดเส้นทางการส่งต่อแพ็กเก็ตบนเครือข่ายไปยัง VNFs ที่ได้ถูกจำลองไว้ให้ตรงตามข้อกำหนดและนโยบายการรักษาความปลอดภัยที่ได้กำหนดไว้ 25

รูปที่ 2.3.1: กลุ่มเทคโนโลยีที่ช่วยในการขับเคลื่อนการจำลองระบบเสมือนจริง (Virtualization technology) 27

รูปที่ 2.3.2: รูปแบบการให้บริการเทคโนโลยีคลาวด์ 28

รูปที่ 2.3.3: การจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริงด้วยเทคโนโลยี NFV 29

รูปที่ 2.3.4: การทำงานของเทคโนโลยี SDN สำหรับควบคุมการรับส่งข้อมูลบนเครือข่ายและกำหนดเส้นทางการส่งต่อแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ประเภทต่างๆ เช่น Firewall, DPI, Load balancer 31

รูปที่ 2.3.5: สถาปัตยกรรมการออกแบบเครื่องเสมือนจริง (VMs) เมื่อเทียบกับการทำงานของเครื่องเซิร์ฟเวอร์ทั่วไป [26] 32

รูปที่ 2.3.6: สถาปัตยกรรมการออกแบบ Container (ด้านซ้าย) เมื่อเทียบกับเครื่อง VMs (ด้านขวา) [13] 32

รูปที่ 2.4.1: แสดงตัวอย่างรูปแบบการโจมตีและภัยคุกคามที่เกี่ยวข้อง 33

รูปที่ 3.2.1: แนวคิดการออกแบบโครงการ T-NOVA ในรูปแบบ Network Functions as-a-Service (NFaaS) 40

รูปที่ 3.2.2: สถาปัตยกรรม T-NOVA 41

รูปที่ 3.2.3: สถาปัตยกรรม ANASTACIA 43

รูปที่ 3.2.4: สถาปัตยกรรม OSM 45

รูปที่ 3.2.5: โมดูลต่างๆ ภายในระบบ Run-time components 47

รูปที่ 3.2.6: สถาปัตยกรรม ONAP 50

รูปที่ 3.2.7: แนวคิดโดยรวมของสถาปัตยกรรม PALANTIR 51

รูปที่ 3.2.8: สถาปัตยกรรมโดยรวมของระบบ PALANTIR 53

รูปที่ 3.2.9: แนวคิดโดยรวมของสถาปัตยกรรม OpenStack Tacker 55

รูปที่ 3.3.1: สถาปัตยกรรมการออกแบบระบบจำลองเครือข่ายเสมือนจริง (NFV) 56

รูปที่ 3.3.2: แผนภาพรวมสถาปัตยกรรม OpenStack 57
รูปที่ 3.3.3: โมดูลต่างๆ บนแพลตฟอร์ม GCP 59
รูปที่ 3.3.4: เครื่องมือของ AWS ที่สนับสนุนการให้บริการด้าน cloud-native NFV orchestratration [36] 60
รูปที่ 3.3.5: การให้บริการของ Microsoft Azure 61
รูปที่ 3.3.6: รูปแบบการให้บริการโครงสร้างพื้นฐานคลาวด์ของ VMware 62
รูปที่ 3.3.7: โครงสร้างของระบบ Snort 64
รูปที่ 3.3.8: โครงสร้างสถาปัตยกรรมของ Suricata 65
รูปที่ 3.3.9: ตัวอย่างการจำลองและติดตั้ง Honeypot ในระบบเครือข่าย 67
รูปที่ 3.3.10: โครงสร้างสถาปัตยกรรมของ Wazuh 68
รูปที่ 3.3.11: โครงสร้างสถาปัตยกรรมของ Zeek 69
รูปที่ 3.3.12: โครงสร้างสถาปัตยกรรม OSM เชิงภาพรวม 70
รูปที่ 3.3.13: โครงสร้างสถาปัตยกรรมของ OpenStack Tacker 71
รูปที่ 3.4.1: โครงสร้างสถาปัตยกรรมการออกแบบของระบบ 72
รูปที่ 3.4.2: แสดงตัวอย่างการแผนภาพกระแสข้อมูลการบริหาร จัดการ และตรวจสอบความปลอดภัยบนเครือข่าย 73
รูปที่ 3.4.3: ตัวอย่าง network service descriptor สำหรับใช้ในการจำลองเครื่องเสมือน (VM) 74
รูปที่ 3.4.4: แสดงตัวอย่างผลการจำลองเครื่องเสมือน (VM) ตาม network service descriptor ที่ได้กำหนดขึ้น 75
รูปที่ 3.4.5: แสดงการเชื่อมต่อระหว่างชั้น NFV MANO (OSM) และชั้น NFVI (OpenStack) 76
รูปที่ 3.4.6: แสดงตัวอย่างการจำลองและใช้งานโครงสร้างพื้นฐาน (NFV infrastructure) ด้วยแพลตฟอร์ม OpenStack ในหลายๆ ที่พร้อมกัน 77
รูปที่ 3.4.7: หน้า CLI แสดงรายละเอียดและข้อมูล OpenStack (cloud-based infrastructure) ที่ระบบได้ดำเนินการเชื่อมต่อ 78
รูปที่ 3.4.8: แสดงตัวอย่างการจำลองเน็ตเวิร์คเซอร์วิส (VNFs) ด้านความปลอดภัยประเภทต่างๆ และบริบทความต้องการใช้งานของผู้รับบริการที่ต้องการใช้งาน VNFs แต่ละประเภทแตกต่างกันออกไป 79
รูปที่ 3.4.9: แสดงโครงสร้างการทำงานที่เกี่ยวข้องกับ web portal 83
รูปที่ 4.2.1: แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้อง ซึ่งประกอบด้วย (1) Admin ผู้ที่รับผิดชอบในการออกแบบพิมพ์เขียว (network service templates) ด้านความปลอดภัยแต่ละประเภท ในที่นี้คือ Suricata, Snort และ Zeek (2) NFV orchestrator ทำหน้าที่รับและประมวลผลพิมพ์เขียวที่ได้ออกแบบ และอัพโหลดพิมพ์เขียวที่ถูกต้องไว้ใน repository กลาง (security service catalog) (3) Policy interpreter ทำหน้าที่ตีนโยบายความปลอดภัยที่ได้กำหนดหรือการตั้งค่าต่างๆ ที่เกี่ยวข้อง (4) Security service catalog ทำหน้าที่เสมือนแหล่งจัดเก็บกลาง (repository) ที่รวบรวมพิมพ์เขียวของแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกสร้างขึ้นและพร้อมใช้งาน (5) OpenStack cloud infrastructure provider รับผิดชอบใน 87

การจัดสรรทรัพยากรโครงสร้างพื้นฐาน ดังเช่น VM, CPU, RAM, ฯลฯ ที่จำเป็นเพื่อรอบรับการจำลองเน็ตเวิร์คเซอร์วิส
รูปที่ 4.2.2: แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้องกับการตรวจสอบและวิเคราะห์ความเสี่ยงด้านภัยคุกคาม ซึ่งประกอบด้วย (1) Log collection agent ซึ่งได้ถูกติดตั้งที่ network gateway ของหน่วยงานผู้รับบริการเพื่อรวบรวมข้อมูล logs การทำงานต่างๆ (2) OpenStack network gateway ทำหน้าที่ในการจัดการเส้นทางและส่งต่อข้อมูล logs ที่ได้รับมาไปยังเน็ตเวิร์คเซอร์วิสด้านความที่กำหนด (3) NS instances คือเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ได้ถูกจำลองขึ้น ได้แก่ Suricata, Snort และ Zeek ซึ่งทำหน้าที่ในการประมวล วิเคราะห์ และตรวจหาความผิดปกติหรือความเสี่ยงที่ก่อให้เกิดภัยคุกคามจากข้อมูล logs (4) Log processor ทำหน้าที่รวบรวมข้อมูลและส่งผลลัพธ์ที่ได้จากการวิเคราะห์ไปแสดงผลผ่านเว็บ และ (5) Security monitoring dashboard ทำหน้าที่ในการนำผลลัพธ์มาจัดจำแนกและแสดงผลในรูปแบบกราฟรายงานต่างๆ เพื่อให้ง่ายต่อการทำความเข้าใจ 88
รูปที่ 4.2.3: แสดงการกำหนดข้อมูลเบื้องต้นสำหรับการจำลอง Suricata 89
รูปที่ 4.2.4: แสดงการตั้งค่า Suricata และการสร้าง metrics สำหรับ monitoring โดย NFV orchestrator จะตรวจสอบโดยอ้างอิงค่า threshold ที่กำหนดในการปรับลดหรือขยายเน็ตเวิร์คเซอร์วิส (Suricata) 90
รูปที่ 4.2.5: แสดงการตั้งค่าเน็ตเวิร์คให้กับ Suricata 91
รูปที่ 4.2.6: ภาพแสดง network topology ของ Suricata, Snort และ Zeek โดย -ext แสดง external link connection 92
รูปที่ 4.2.7: ภาพแสดง network topology ของ Suricata, Snort และ Zeek ซึ่งทั้งสามประเภทจะถูกเชื่อมต่อเข้ากับ Network วงเดียวกันคือ ‘client2’ 93
รูปที่ 4.2.8: ตัวอย่างการตั้งค่าพอร์ตที่จะใช้สำหรับการฟังข้อมูล traffic 94
รูปที่ 4.2.9: ตัวอย่างการเพิ่ม rules ใหม่และกำหนด path ไปยัง default rule 94
รูปที่ 4.2.10: แสดงสถานะการเปิดใช้งาน Suricata service 94
รูปที่ 4.2.11: แสดงตัวอย่างการตั้งค่า log file บน Snort service 95
รูปที่ 4.2.12: แสดงตัวอย่างการตั้งค่า log file บน Zeek 95
รูปที่ 4.2.13: แสดงตัวอย่างการตั้งค่า interface บน Zeek service 95
รูปที่ 4.2.14: โครงสร้างการเชื่อมต่อเครือข่าย (network topology) ของแต่ละเน็ตเวิร์คเซอร์วิสบน OpenStack 96
รูปที่ 4.2.15: ผลลัพธ์หลังจากการจำลองข้อมูลทราฟฟิกจาก public dataset 97
รูปที่ 4.2.16: ผลลัพธ์ของ fast.log หลังการจำลองข้อมูลทราฟฟิกจาก public dataset 98
รูปที่ 4.2.17: ผลลัพธ์ของ eve.json หลังจากจำลองข้อมูลทราฟฟิกจาก public dataset 98
รูปที่ 4.2.18: ผลลัพธ์ของ alert_json.txt หลังการจำลองข้อมูลทราฟฟิกจาก public dataset 99
รูปที่ 4.2.19: ผลลัพธ์ของ http.log หลังจากจำลองข้อมูลทราฟฟิกจาก public dataset 99
รูปที่ 4.2.20: ไฟล์ log หลังจากการจำลองข้อมูลทราฟฟิกจาก public dataset 99

รูปที่ 4.3.1: หน้าต่าง dashboard แสดงข้อมูลแจ้งสถานะทำงานของ instances ต่างๆ ได้แก่ CPU utilization และ memory utilization 100
รูปที่ 4.3.2: หน้าต่าง dashboard แสดงข้อมูลแจ้งสถานะทำงานของ instances ต่างๆ ได้แก่ Resident memory usage, Process virtual memory, VM status และ VNF status 101
รูปที่ 4.3.3: หน้าต่าง dashboard แสดงผลการแจ้งเตือน (alerts) 102
รูปที่ 4.3.4: หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Snort 103
รูปที่ 4.3.5: หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Suricata 105
รูปที่ 4.3.6: หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Zeek 106
รูปที่ 4.4.1: ภาพแสดงข้อมูลที่เกี่ยวข้องกับระยะเวลาที่ใช้ในการจำลองและติดตั้งเน็ตเวิร์คเซอร์วิส (NS instances) 107
รูปที่ 4.4.2: กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในการติดตั้งเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ได้แก่ Suricata, Snort และ Zeek (**หมายเหตุ:เนื่องจากว่าการติดตั้งนั้นต้องทำเป็นลำดับขั้นจึงทำให้มีความแตกต่างของเวลา ซึ่งในความเป็นจริงแล้วเวลาที่ใช้ควรจะใกล้เคียงกัน และก็ขึ้นอยู่กับการใช้งานทรัพยากรของเครื่องเซิฟเวอร์ในขณะนั้นด้วย) 108
รูปที่ 4.4.3: ภาพแสดงข้อมูลเกี่ยวกับระยะเวลาที่ใช้ในการตั้งค่าเบื้องต้นจากไฟล์ cloud-init 108
รูปที่ 4.4.4: กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในตั้งค่าตั้งค่าเบื้องต้นในแต่ละเน็ตเวิร์คเซอร์วิส (**หมายเหตุ (1) สาเหตุที่ Zeek ใช้เวลานานนั้นเกิดจากการที่ต้องมีการเปิดใช้งานโดยใช้คำสั่ง sudo zeekctl deploy; (2) สาเหตุที่ Suricata ใช้เวลานานกว่า Snort ก็มาจากการ configuration ใน Suricata.yaml ที่มีการใช้คำสั่ง sudo systemctl restart suricata.service ส่งผลให้ใช้เวลาเพิ่มขึ้นในส่วนนี้) 109
รูปที่ 4.4.5: แสดงข้อมูลเกี่ยวกับ log alert ที่แสดงผ่าน Wazuh dashboard 109
รูปที่ 4.4.6: กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาแฝงเฉลี่ย (end-to-end latency) ของแต่ละเน็ตเวิร์คเซอร์วิส (**หมายเหตุ: (1) จากผลการทดลองพบข้อสันนิฐานว่าสาเหตุที่ Suricata มีการส่งที่รวดเร็วอาจจะเป็นเพราะมีการรองรับ JSON log ไฟล์อยู่แล้ว ซึ่งแตกต่างจาก Zeek และ Snort ทีจะต้องมีการแปลงข้อมูลหรือเขียน log ให้อยู่ในรูป JSON) (2) สาเหตุที Zeek ส่งข้อมูลช้าอาจจะเป็นผลมาจากการที่ Zeek นั้นมีหลาย log ที่ต้องส่งข้อมูลจึงอาจจะทำให้เกิด bottle neck ที่ตัว fluent-bit ซึ่งทำหน้าที่เป็น log processor จึงทำให้เกิดความล่าช้าในส่วนนี้ 110
รูปที่ 4.5.1: คุณลักษณะสำคัญของแพลตฟอร์มฯ 111
รูปที่ 5.2.1: กล่าวรายงานการจัดกิจกรรมโดยผู้ช่วยศาสตราจารย์ ดร. วราภรณ์ พรหมวิกร ผู้ช่วยเลขาธิการคณะกรรการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) 115
รูปที่ 5.2.2: กล่าวเปิดงานและบรรยายพิเศษโดยพลอากาศตรีอมร ชมเชย ผู้ช่วยเลขาธิการคณะกรรการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) 116
รูปที่ 5.2.3: ภาพบรรยากาศกิจกรรมสัมมนา 116

รูปที่ 5.2.4: ดร มลธิดา ภัทรนันทกุล นักวิจัยจากสวทช. สังกัดเนคเทค ได้เสนองานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ ในหัวข้อเรื่อง ซีโร่ทัชเซอร์วิส ระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยอัตโนมัติสำหรับตรวจสอบและเฝ้าระวังภัยคุกคามไซเบอร์ 117
รูปที่ 5.2.5: โปสเตอร์ประชาสัมพันธ์กิจกรรมเสวนา Zero-Touch Services ที่จัดขึ้น 118
รูปที่ 5.2.6: สถิติผู้สนใจลงทะเบียนเข้าร่วมกิจกรรมเสวนา 119
รูปที่ 5.2.7: บรรยากาศจุดรับลงทะเบียนหน้าห้องจัดสัมมนา 119
รูปที่ 5.2.8: แนะนำวิทยากร 120
รูปที่ 5.2.9: คุณเอกฉันท์ รัตนเลิศนุสรณ์ หัวหน้าทีมวิจัยความมั่นคงปลอดภัยไซเบอร์ กลุ่มวิจัยการสื่อสารและเครือข่าย เนคเทค กล่าวต้อนรับและเปิดงาน 120
รูปที่ 5.2.10: ดร. มลธิดา ภัทรนันทกุล (หัวหน้าโครงการฯ) ทีมวิจัยความมั่นคงปลอดภัยไซเบอร์ กลุ่มวิจัยการสื่อสารและเครือข่าย เนคเทค ได้กล่าวถึงภูมิทัศน์ภัยคุกคามทางไซเบอร์ในประเทศไทย 120
รูปที่ 5.2.11: มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคอุดมศึกษา โดยรองศาสตราจารย์ปานใจ ธารทัศนวงศ์ ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร 121
รูปที่ 5.2.12: มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาครัฐ โดยนายแพทย์ภาณุพงศ์ ตันติรัตน์ รองผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงสาธารณสุข 121
รูปที่ 5.2.13: มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคเอกชน โดยคุณนรินทร์ รุณภัย ที่ปรึกษาอิสระ ด้านความมั่นคงปลอดภัยไซเบอร์ ภาคเอกชน 121
รูปที่ 5.2.14: มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคเอกชน โดยคุณอังคาร รองกรรมการผู้จัดการ บริษัททีที-เน็ต ไอที โซลูชัน จำกัด 122
รูปที่ 5.2.15: แนะนำแพลตฟอร์มโดยดร. มลธิดา ภัทรนันทกุล 122
รูปที่ 5.2.16: บรรยากาศกิจกรรมเสวนา 123
รูปที่ 5.2.17: ช่วงถามตอบ 124
รูปที่ 5.3.1: ภาพบรรยากาศการออกบูธจัดแสดงผลงานวิจัยในงาน NECTEC-ACE 2024 125
รูปที่ 5.3.2: แนะนำผลงานและแลกเปลี่ยนความคิดเห็นจากผู้ที่สนใจ 126
รูปที่ 5.3.3: ภาพบรรยากาศการออกบูธจัดแสดงผลงานวิจัยในงาน BTFP Showcase 2024 127
รูปที่ 5.3.4: การเยี่ยมชมผลงานวิจัยโดยผู้ทรงคุณวุฒิ 127
รูปที่ 5.3.5: บรรยากาศการแลกเปลี่ยนความคิดเห็นจากผู้ที่สนใจ 128
รูปที่ 5.3.6: กิจกรรมเสวนาบนเวที 128
รูปที่ 5.4.1: ทีมวิจัยลงพื้นที่เพื่อเข้าร่วมประชุมหารือถึงแนวทางการติดตั้งและทดสอบใช้งานนวัตกรรมร่วมกับเจ้าหน้าที่โรงพยาบาลสระบุรี ต. ปากเพรียว อ. สระบุรี จ. สระบุรี เมื่อวันที่ 11 กุมภาพันธ์ พ.ศ. 2568 129
รูปที่ 5.4.2: การเข้าร่วมประชุมออนไลน์ระหว่างทีมวิจัยกับสำนักงานสาธารณสุขจังหวัดลพบุรี เพื่อหารือถึงแนวทางการติดตั้งและทดสอบใช้งานนวัตกรรม เมื่อวันที่ 27 กุมภาพันธ์ พ.ศ. 2568 130

รูปที่ 5.5.1: สกมช.เชิญชวนส่งผลงานเพื่อเข้าประกวดรางวัลผลิตภัณฑ์และบริการด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ “Thailand Cybersecurity Product and Service 2024” 131
รูปที่ 5.5.2: ประกาศผลรางวัลผลิตภัณฑ์และบริการด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ “Thailand Cybersecurity Product and Service 2024” 132
รูปที่ 5.5.3: การกล่าวเปิดงานโดยพล.อ.ต.อมร ชมเชย เลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) 132
รูปที่ 5.5.4: บรรยากาศการประกาศผลรางวัล 133
รูปที่ 5.5.5: บรรยากาศการเยี่ยมชมบูธจัดแสดงผลงานโดยผู้ทรงคุณวุฒิ 133
รูปที่ 5.5.6: บรรยากาศเวทีเสวนาแลกเปลี่ยนแนวคิดผลงานที่ได้รับรางวัล 134
รูปที่ 5.5.7: นายประเสริฐ จันทรวงทอง รองนายกรัฐมนตรี ร่วมแสดงความยินดีกับผู้ที่ได้รับรางวัล ณ อาคารสำนักงานใหญ่ บริษัทโทรคมนาคม 134

# บทที่ 1
# สาระสำคัญของโครงการ (Project Highlight)

## 1.1 หลักการและเหตุผล

สถานการณ์โลกในปัจจุบันมีการเปลี่ยนแปลงอย่างรวดเร็วและสลับซับซ้อนมากกว่าในอดีต ผลกระทบที่กำลังเกิดขึ้นจากการเคลื่อนตัวสู่ยุคอุตสาหกรรม 4.0 ได้นำมาซึ่งการเปลี่ยนแปลงในทุกมิติ ทั้งด้านการพัฒนาโครงสร้างพื้นฐาน ICT ด้านธุรกิจและการบริการ ผู้บริโภคส่วนใหญ่ต่างคาดหวังว่าโครงสร้างพื้นฐาน ICT และระบบนิเวศดิจิทัลจะสามารถรองรับการขยายตัวที่เพียงพอต่อความต้องการขององค์กรและปริมาณการรับส่งข้อมูลที่ไหลเข้ามาอย่างต่อเนื่องและมหาศาลได้ อีกทั้งความคาดหวังว่าการพึ่งพาเทคโนโลยีนั้นจะสามารถยกระดับประสิทธิภาพการประมวลผลและการดำเนินงาน และมีศักยภาพมากพอในการจัดสรรหรือขยายทรัพยากรเน็ตเวิร์คได้อย่างยืดหยุ่นและอิสระ เพื่อให้สอดคล้องกับความต้องการที่หลากหลายของกลุ่มผู้ใช้งานในแต่ละภาคธุรกิจได้ กลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ได้แก่ (1) การจำลองฟังก์ชันเครือข่ายเสมือน (Network Functions Virtualization: NFV) (2) การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) และ (3) การจัดเก็บและการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) กลายเป็นกลุ่มเทคโนโลยีที่ได้เข้ามามีบทบาทสำคัญในการขับเคลื่อนโครงสร้างพื้นฐานระบบเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบดั้งเดิม (traditional networking model) ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดโดยซอฟต์แวร์ (Software-defined infrastructure) เพื่อรองรับสภาพแวดล้อม 5G ในอนาคตในทุกๆ มิติของภาคอุตสาหกรรม (5G services) รวมถึง Internet of Things (IoTs), smart city, smart factory เป็นต้น นำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการให้บริการเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่ายที่สามารถตอบโจทย์ทั้งในเรื่อง (1) การลดค่าใช้จ่ายในการจัดซื้ออุปกรณ์เครือข่ายและค่าใช้จ่ายที่เกี่ยวข้องกับการดำเนินงาน (Capital Expenditure (CapEx) และ Operational Expenditure (OpEx)) (2) การตอบสนองการให้บริการอย่างทันถ่วงที (rapid service deployment) (3) การให้ผลลัพธ์การประมวลผลข้อมูลได้อย่างมีประสิทธิภาพ (Operational efficiency) (4) การสร้างความยืดหยุ่นในการจัดสรรหรือขยายทรัพยากรเครือข่ายให้มีความเหมาะสมกับปริมาณงานในช่วงเวลานั้นๆ (dynamic resource scaling) ตลอดจน (5) การเรียกใช้งานเน็ตเวิร์คเซอร์วิสต่างๆ สามารถทำได้อย่างยืดหยุ่นเพื่อตอบสนองความต้องการที่หลากหลายของกลุ่มผู้ใช้งานในแต่ละภาคธุรกิจได้ (*network on-demand services*) โดยแนวคิดของเทคโนโลยี NFV คือ การแยกเน็ตเวิร์คเซอร์วิส (network services) ในส่วนของซอฟต์แวร์ซึ่งเดิมเคยผูกติดกับฮาร์ดแวร์เฉพาะ (proprietary hardware) ของแต่ละค่ายผู้ผลิตออกมาให้กลายเป็นซอฟต์แวร์ที่สามารถรันได้บนเครื่องเสมือนจริงบนระบบคลาวด์แพลตฟอร์มได้ หรือที่เรียกว่า การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (Virtualized Network Functions: VNFs) ซึ่งข้อดีของเทคโนโลยี NFV จะช่วยเพิ่มประสิทธิภาพการทำงานได้ดียิ่งขึ้นทั้งในแง่ของความรวดเร็วในการ deploy เน็ตเวิร์คเซอร์วิส ความยืดหยุ่นในการจัดสรรทรัพยากรเครือข่าย และประสบการณ์ที่ผู้ใช้จะได้รับ ขณะที่แนวคิดของเทคโนโลยี SDN ช่วยสนับสนุนเทคโนโลยี NFV ในเรื่องการบริหารจัดการ ควบคุม และกำหนดการเชื่อมต่อเครือข่ายด้วยซอฟต์แวร์ เช่น เข้าถึงการตั้งค่าและควบคุมการรับส่งข้อมูลบนเครือข่าย (full network control capabilities) กำหนดเส้นทางการส่งต่อแพ็กเก็ต (routing

decision) เพื่อให้สอดคล้องกับภาระงานของอุปกรณ์เครือข่ายนั้นๆ (routing decision) และควบคุมการไหลของแพ็กเก็ต (traffic steering and forward) ให้ตรงตามข้อกำหนดหรือนโยบายที่กำหนดไว้ เหล่านี้เป็นต้น ทำให้ผู้บริการหรือผู้ที่เกี่ยวข้อสามารถบูรณาการเส้นทางการเชื่อมต่อเครือข่ายและกำหนดเส้นทางการส่งต่อแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสต่างๆ ที่ต้องการได้อย่างยืดหยุ่นเพื่อให้เหมาะสมกับการขยายตัวและความต้องการที่หลากหลายของกลุ่มผู้บริโภคได้

อย่างไรก็ตาม แม้ว่ากลุ่มเทคโนโลยีการจำลองระบบเสมือนจริงนำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการบริหารเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่าย อีกทั้งคุณประโยชน์มากมายต่อการสร้างโอกาสทางธุรกิจใหม่ๆ และเป็นเสาหลักที่สำคัญต่อการวางรากฐานการขับเคลื่อนโครงสร้างพื้นฐาน ICT ของประเทศไปสู่ยุคดิจิทัลอย่างยั่งยืนในอนาคต ในอีกมุมหนึ่ง องค์ความรู้เกี่ยวกับเทคโนโลยีนี้ในประเทศไทยแทบจะมีน้อยมาก โดยเฉพาะอย่างยิ่งผู้ให้บริการเครือข่ายส่วนใหญ่ต่างหันมาพึ่งพาการนำเข้าเทคโนโลยีและนวัตกรรมจากต่างประเทศที่มีต้นทุนสูง ปัจจัยเหล่านี้ล้วนนำมาซึ่ง (1) อุปสรรคให้นวัตกร นักคิด ผู้พัฒนาหรือผู้ประกอบการ ในการต่อยอดงานวิจัยและพัฒนา (R&D) ไปสู่การสร้างกระบวนทัศน์ใหม่สำหรับผลิตภัณฑ์หรือเซอร์วิสที่สามารถตอบโจทย์กลุ่มเป้าหมายในบริบทประเทศไทยได้ (2) อุปสรรคต่อการพัฒนาประเทศที่เกิดจากความสามารถในการพึ่งพาตนเอง (3) อีกทั้งเป็นอุปสรรคหลักสำคัญที่ทำให้ประเทศไทยไม่สามารถเข้าสู่ Thailand 4.0 ได้อย่างเต็มรูปแบบต่อการพัฒนาเศรษฐกิจให้ก้าวข้ามพ้นจากการเป็นประเทศที่กำลังพัฒนาสู่การเติบโตอย่างยั่งยืนและระยะยาว

ในอีกมุมหนึ่ง หากวิเคราะห์ข้อมูลเจาะลึกลงไปในรายละเอียด ปัจจุบันพบว่าหลายๆ ประเทศทั้งในทวีปยุโรปและเอเชียต่างให้ความความสนใจที่จะพัฒนาและขับเคลื่อนงานวิจัยที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริงไปสู่การใช้งานจริงมากขึ้น โดยสาเหตุหลักสำคัญเนื่องจากประเทศส่วนใหญ่ต่างเห็นพ้องต้องกันว่าเทคโนโลยีการจำลองระบบเสมือนจริงจะนำมาซึ่งการเปลี่ยนแปลงโครงสร้างเครือข่ายครั้งสำคัญ อันสามารถลดข้อจำกัดต่างๆ ที่เคยประสบในระบบเครือข่ายแบบดั้งเดิมได้ (ดังเช่น ปัจจัยด้านค่าใช้จ่ายที่ต้องสูญเสียไปสำหรับการจัดซื้อและบำรุงรักษาอุปกรณ์เครือข่าย ความล่าช้าในการจำลองเน็ตเวิร์คเซอร์วิส ความซับซ้อนในการเข้าถึงการตั้งค่าต่างๆ การขาดสภาพคล่องในการทำงานต่อการจัดสรรทรัพยากรได้อย่างไดนามิก ฯลฯ) โดยปัจจัยเหล่านี้ล้วนเปรียบเสมือนปัจจัยหลักที่สำคัญต่อการวางรากฐานโครงสร้าง ICT และการเตรียมระบบนิเวศ 5G ในอนาคต แม้กระนั้น หากพิจารณาในบริบทประเทศไทย เส้นทางการพัฒนานวัตกรรมที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริงสู่การประยุกต์ใช้งานจริงพบว่ายังอยู่ในช่วงเริ่มต้นของการศึกษาและยังไม่ได้รับความนิยมใช้กันอย่างแพร่หลายมากนัก นอกเหนือจากปัญหาดังที่กล่าวมาข้างต้น ปัญหาที่สำคัญอีกประการหนึ่ง คือ ความไม่พร้อมของกำลังคนและความท้าทายในการนำเทคโนโลยีการจำลองระบบเสมือนจริงมาปรับใช้ภายในองค์กร เนื่องจากผู้ที่เกี่ยวข้องส่วนใหญ่มักมองเห็นภาพไม่ชัดเจนว่าเทคโนโลยีดังกล่าวฯ จะเข้ามามีบทบาทหรือสร้างผลกระทบต่อองค์กรอย่างไร หรือแม้กระทั่งการกำหนดกรณีศึกษา (use case) ที่เหมาะสมต่อองค์กรเพื่อประโยชน์สูงสุดต่อการปรับใช้เทคโนโลยีฯ ยังคงเป็นเรื่องใหม่และท้าทาย

ด้วยเหตุนี้ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) จึงมีความสำคัญอย่างมากอันเปรียบเสมือนโครงการนำร่องในการแสดงให้หน่วยงาน องค์กร

หรือบุคคลที่เกี่ยวข้องได้เห็นภาพชัดเจนยิ่งขึ้นว่าการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง จักนำมาซึ่งคุณประโยชน์และผลกระทบเชิงบวกให้กับองค์กรได้อย่างไรผ่านการประยุกต์ใช้งานจริง ซึ่งการออกแบบสถาปัตยกรรมระบบมุ่งเน้นการพัฒนาเพื่อให้เกิดผลสัมฤทธิ์ด้วยกัน 3 ด้าน ดังนี้

1. การจัดเตรียมระบบนิเวศเสมือนจริง (management and orchestration framework) สำหรับจัดสรรและบริหารทรัพยากรเครือข่าย (network infrastructure and resources) ให้พร้อมสำหรับใช้งาน
2. การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs as a Service: VNFaaS) ดังเช่น เน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ ไว้บนคลาวด์ ดังเช่น Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber traps, Traffic analyzer ฯลฯ เพื่อให้องค์กรหรือหน่วยงานที่สนใจสามารถใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ ที่จัดเตรียมไว้ให้ได้อย่างเหมาะสม เพื่อสนับสนุนการรักษาความปลอดภัยบนระบบเครือข่ายขององค์กร
3. การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนและการทำ service function chaining ตามบริบทและความต้องการของผู้ใช้งาน (VNFs on-demand services)

**1.2 วัตถุประสงค์ของโครงการ**

โครงการนี้จัดตั้งขึ้นมาโดยมีแนวคิดหลัก 3 ประการดังต่อไปนี้

1. **เชิงการวิจัยและพัฒนา (R&D):** เพื่อวิจัยและพัฒนาแพลตฟอร์มดังกล่าวฯ (1) สนับสนุนให้เกิดการสร้างองค์ความรู้และนวัตกรรมใหม่ๆ ด้วยการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ที่สอดคล้องกับการพัฒนาโครงสร้างพื้นฐาน ICT และบริบทของประเทศ (2) ส่งเสริมให้เกิดการสร้างระบบนิเวศเครือข่ายเสมือนจริงเพื่อช่วยให้ผู้บริการหรือผู้ดูแลเครือข่ายสามารถบริหารและจัดการวัฏจักรของแต่ละเน็ตเวิร์คเซอร์วิส และมีความคล่องตัวในการจัดสรรทรัพยากรเครือข่ายเพื่อให้มีความเหมาะสมกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ ได้อย่างมีประสิทธิภาพ และ (3) ส่งเสริมให้เกิดการให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนประเภทต่างๆ ดังเช่น เน็ตเวิร์คเซอร์วิสที่เกี่ยวข้องกับการรักษาความปลอดภัยในเครือข่าย ได้แก่ Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber traps, Traffic analyzer, ฯลฯ ตามความต้องการของผู้ใช้งานได้ (VNFs on-demand services)
2. **เชิงเศรษฐกิจ:** เพื่อสนับสนุนให้เกิดการสร้างกระบวนทัศน์หรือนวัตกรรมใหม่ๆ ด้วยการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริงที่เหมาะสมต่อการใช้งานในบริบทของประเทศไทย ตลอดจนเป็นการส่งเสริมให้เกิดการลดสัดส่วนการพึ่งพาเทคโนโลยีจากต่างประเทศ โดยการใช้องค์ความรู้ เครื่องมือ และบุคลากรที่มีในการพัฒนานวัตกรรมขึ้นเอง เพื่อสร้างโอกาสให้การเพิ่มขีดความสามารถด้านการแข่งขันและการพึ่งพาตัวเอง
3. **เชิงสังคม:** เพื่อพัฒนาแนวทางในการใช้ประโยชน์จากแพลตฟอร์มดังกล่าวฯ โดยการส่งเสริมให้เกิดการใช้งานจริง ตลอดจนเพื่อส่งเสริมให้เกิดการสร้าง use case ใหม่ๆ ในทุกๆ มิติของภาคอุตสาหกรรมอย่างยั่งยืน เช่น การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนสนับสนุนงานด้าน smart city, mission critical

application, condition monitoring, network-security monitoring ฯลฯ เพื่อส่งเสริมขีดความสามารถของเทคโนโลยีการจำลองระบบเสมือนจริง สนับสนุนการขับเคลื่อนเศรษฐกิจและสังคมดิจิทัลสู่ยุค 5G ในอนาคต โดย use case ที่สนใจในระยะเริ่มต้นของการพัฒนา (*ระยะที่ 1: proof of concept*) คือ การจัดเตรียมเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs as a Service: VNFaaS) ด้านความปลอดภัยต่างๆ ไว้บนระบบคลาวด์ ดังเช่น Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber traps, Traffic analyzer ฯลฯ เพื่อให้ผู้มาขอรับบริการเครือข่ายหรือผู้ที่เกี่ยวข้องสามารถเรียกใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ได้ถูกจำลองขึ้น สำหรับการรักษาความปลอดภัยภายในองค์กรของตนเอง

**1.3 ขอบเขตการดำเนินงาน**

โครงการนี้มีวัตถุประสงค์ในการวิจัย พัฒนานวัตกรรม และสร้างกระบวนทัศน์ใหม่ด้วยการจัดเตรียมระบบนิเวศเสมือนจริง (management and orchestration platform) สำหรับบริหารและจัดการทรัพยากรเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs as a Service: VNFaaS) สนับสนุนเทคโนโลยี NFV และ SDN เพื่อให้ผู้ที่เกี่ยวข้องสามารถเรียกใช้เน็ตเวิร์คเซอร์วิสต่างๆ ได้ตามความเหมาะสมสอดคล้องกับความต้องการใช้งานในแต่ละองค์กรได้ อย่างไรก็ตามเทคโนโลยีดังกล่าวฯ ถือว่าเป็นเทคโนโลยีใหม่ในบริบทของประเทศไทย ดังนั้นขอบเขตการดำเนินงานของโครงการจะแบ่งออกเป็น 3 เฟสย่อย ดังนี้

1. ดำเนินการศึกษาเพื่อสำรวจเทคโนโลยีการจำลองระบบเสมือนจริงเชิงลึก โดยคลอบคลุมถึงสถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้อง อันเป็นประโยชน์ต่อการสร้างองค์ความรู้ในเทคโนโลยีการจำลองระบบเสมือนจริง
2. ดำเนินการพิสูจน์แนวคิด (proof of concept) และศึกษาถึงความเป็นไปได้สู่การใช้งานจริง โดยการพัฒนาแพลตฟอร์มสำหรับบริหารจัดการ และให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนจริงเพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต โดย use case ที่สนใจในระยะเริ่มต้นของการพัฒนาโครงการนี้ (ระยะที่ 1: Proof of Concept) คือ การจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (Virtual Network Functions: VNFs) ด้านความปลอดภัยต่างๆ ไว้บนคลาวด์ ดังเช่น DPI (e.g., Suricata), IDS (e.g., Snort), Cyber traps (e.g., Honeypots), Traffic analyzer (e.g., Zeek) ฯลฯ เพื่อให้ผู้รับบริการเครือข่ายหรือผู้ที่เกี่ยวข้องสามารถเรียกใช้และใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสดังกล่าวฯ ได้ตรงกับตามความต้องการใช้งาน (security requirements) สำหรับการรักษาความมปลอดภัยภายในองค์กร โดยการพัฒนาระบบจะประกอบด้วย 2 ระบบย่อยโดยมีรายละเอียดดังนี้
    a. ระบบหลังบ้าน (Back-end framework)
        i. พัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง (management and orchestration platform) เพื่อทำหน้าที่หลักในการจัดเตรียมทรัพยากรต่างๆ ได้แก่ CPU, หน่วยความจำ พื้นที่จัดเก็บข้อมูล และระบบเครือข่าย ให้

พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิสต่างๆ ตลอดจนควบคุมลำดับการทำงานให้เป็นไปอย่างอัตโนมัติสอดคล้องกับนโยบายที่กำหนดไว้

ii. พัฒนาส่วนเชื่อมต่อเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ด้านความปลอดภัยต่างๆ เช่น Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber traps, Traffic analyzer ฯลฯ เพื่อให้องค์กรที่รับบริการสามารถดำเนินการตั้งค่าและเรียกใช้งานเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ต้องการได้อย่างเหมาะสมสำหรับรักษาความปลอดภัยในองค์กร

iii. พัฒนาช่องทาง REST-based APIs ไว้สำหรับดึงข้อมูลจากเน็ตเวิร์คเซอร์วิสมาต่างๆ มาแสดงผลผ่าน dashboard เช่น ข้อมูลแสดงผลประเภทภัยคุกคามที่ตรวจพบ จำนวนภัยคุกคามที่เกิดขึ้นในช่วงเวลาต่างๆ ฯลฯ

iv. พัฒนาโมดูลสำหรับตรวจสอบการใช้งานทรัพยากรภายในระบบ (resource monitoring) เพื่อเพิ่มขีดความสามารถในการปรับเพิ่ม/ลด หรือขยายทรัพยากรให้กับเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น (scaling in/out network resources) เพื่อให้มีความเหมาะสมและสอดคล้องกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ

b. ระบบหน้าบ้าน (Front-end framework)

i. พัฒนา web portal ซึ่งเป็นส่วนการจัดการข้อมูลหน่วยงานผู้ใช้บริการและจัดการสิทธิ์การเข้าถึงข้อมูลตามบทบาทและหน้าที่ที่ได้รับ รวมถึงการรับเข้าข้อมูลจากผู้ใช้งาน เช่น ข้อมูลการตั้งค่านโยบายความปลอดภัย

ii. พัฒนา dashboard เพื่อนำผลลัพธ์ที่ได้จากการประมวลผลของเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ที่ถูกเชื่อมต่อเข้าด้วยกันในระบบหลังบ้าน (back-end) มาแสดงผลในรูปแบบต่างๆ ที่ผู้ใช้งานสามารถเข้าใจได้ง่าย เช่น ในรูปแบบกราฟแสดงผลหรือรายงาน ฯลฯ

3. ดำเนินการทดสอบประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน ได้แก่ การทำงานเชิงหน้าที่ (functionality) ด้านการใช้งาน (feasibility) และด้านประสิทธิภาพ (performance)

**1.4 ผลที่คาดว่าจะได้รับ**

1. มีการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริงไปใช้เพื่อต่อยอด สร้าง และพัฒนานวัตกรรมใหม่ๆ เพื่อวางรากฐานโครงสร้างพื้นฐาน ICT และการสร้างระบบนิเวศดิจิทัลที่เข้มแข็งต่อการขับเคลื่อนประเทศไปสู่สภาพแวดล้อม 5G ในอนาคต
2. มีแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงและให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนประเภทต่างๆ (เช่น เน็ตเวิร์คเซอร์วิสด้านความปลอดภัย) ที่สอดรับกับการส่งเสริมให้เกิดการพัฒนานวัตกรรมใหม่ๆ ด้วยเทคโนโลยีการจำลองระบบเสมือนจริง โดยผ่านการพิสูจน์แนวคิด (proof of

concept) เพื่อพิสูจน์ว่าระเบียบวิธีที่สร้างขึ้นสามารถทำงานได้ถูกต้องทางทฤษฎี และทดสอบความเป็นได้ในการนำไปสู่การใช้งานจริง (feasibility validation)

3. เกิดการสร้าง use case ใหม่ๆ ในทุกๆ มิติของภาคอุตสาหกรรม เช่น การให้บริการและการเชื่อมต่อเน็ตเวิร์คเซอร์วิสแบบเสมือนสนับสนุนงานด้านความปลอดภัย (threat detection, monitoring and prevention) ผลที่ได้จักนำมาซึ่งการส่งเสริมขีดความสามารถของเทคโนโลยีการจำลองระบบเสมือนจริงและการสร้างความได้เปรียบทางการค้าให้สามารถแข่งขันได้ในตลาดโลก

**1.5 ตัวชี้วัดความสำเร็จ**

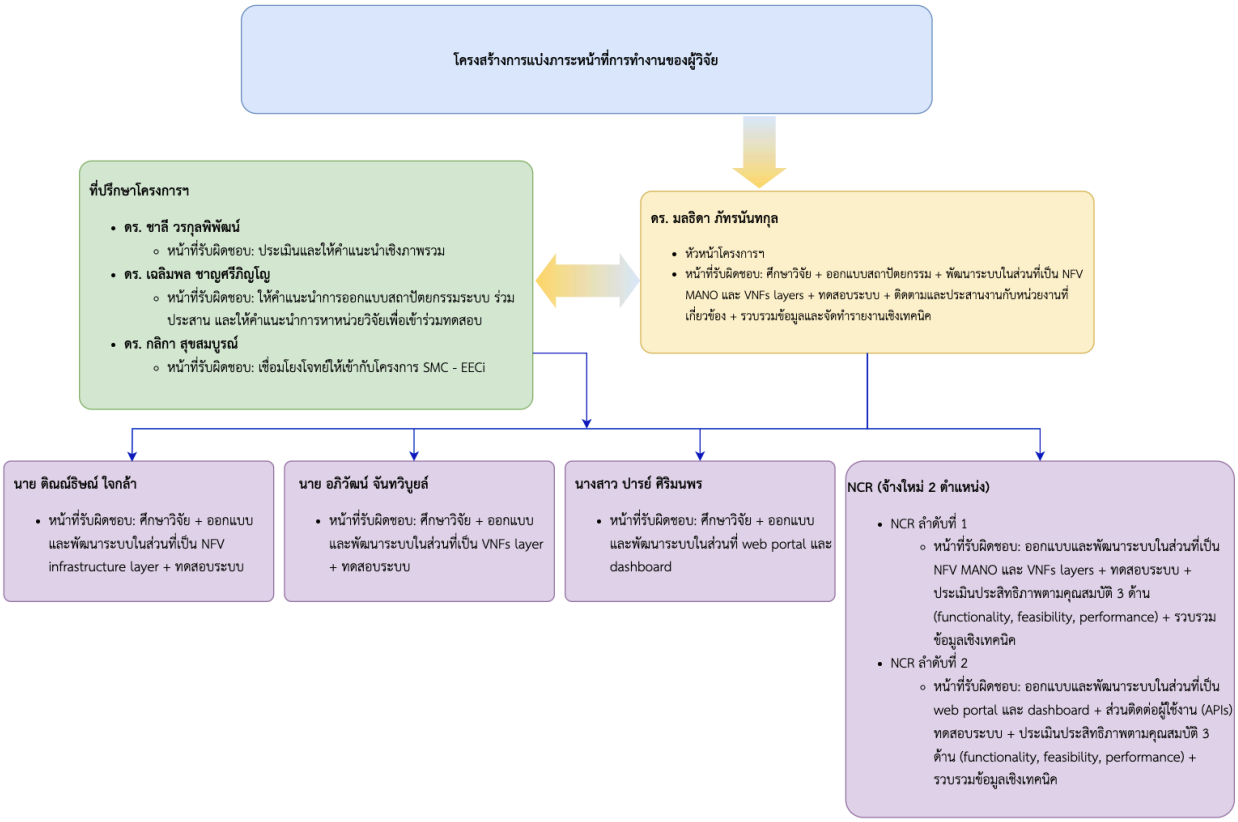
1. แพลตฟอร์มสำหรับบริหารทรัพยากรเครือข่ายและการจัดการเน็ตเวิร์คเซอร์วิสแบบเสมือน 1 ระบบพร้อมสถาปัตยกรรม
2. มีการสร้าง use case เพื่อสนับสนุนการใช้งานเน็ตเวิร์คเซอร์วิสแบบเสมือนอย่างน้อย 1 use case โดยในระยะเริ่มต้นของการพัฒนา (ระยะที่ 1: proof of concept) มุ่งเน้นการจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนด้านความปลอดภัยเป็นหลัก ได้แก่ Deep Packet Inspection (e.g., Suricata), Intrusion Detection System (e.g., Snort), Traffic analyzer (e.g., Zeek)
3. สามารถทำงานได้ในสภาพแวดล้อมที่จัดเตรียมไว้ให้ เช่น การนำระบบไปติดตั้งในพื้นที่จัดแสดงในเขตนวัตกรรมระเบียงเศรษฐกิจพิเศษตะวันออก (EECi)
4. มีการวัดผลในเชิงประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน ได้แก่ การทำงานเชิงหน้าที่ (functionality) ด้านการใช้งาน (feasibility) และด้านประสิทธิภาพ (performance)
5. มีหน่วยงานเข้าร่วมทดสอบและใช้งานอย่างน้อย 1 หน่วยงาน
6. มีการตีพิมพ์ผลงานหรือได้รับการตอบรับบทความในวารสารระดับนานาชาติที่มี Impact Factor (SCI Index) 1 บทความ

**1.6 คณะผู้วิจัยในการดำเนินโครงการ**

1. ดร. มลธิดา ภัทรนันทกุล (หัวหน้าโครงการวิจัย)
   ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
2. ดร. ชาลี วรกุลพิพัฒน์
   ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
3. ดร. กลิกา สุขสมบูรณ์
   ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
4. นาย ติณณ์ธิษณ์ ใจกล้า

ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)

5. นาย อภิวัฒน์ จันทวิบูลย์
   ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
6. นางสาว ปารย์ ศิริมนพร
   ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
7. ดร. เฉลิมพล ชาญศรีภิญโญ
   ด้านสนับสนุนการวิจัยและพัฒนา ศูนย์เทคโนโลยีอิเล็กทรอนิกสDและคอมพิวเตอร์แห่งชาติ (เนคเทค)




***รูปที่ 1.1:*** *แผนผังการแบ่งภาระหน้าที่การดำเนินงานของผู้วิจัย*

## 1.7 แผนการดำเนินโครงการ

แนวทางการดำเนินงานโครงการของแต่ละกระบวนการมีดังนี้

| แผนงาน/กิจกรรม | ปีที่ 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | เดือน | | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **WP1:** วางแผนโครงการ ศึกษา สำรวจ และรวบรวมข้อมูลที่เป็นประโยชน์ต่อการพัฒนาแพลตฟอร์ม | | | | | | | | | | | | |
| 1.1 จัดทำรายงานแผนการดำเนินงาน (Project plan) ประกอบด้วย รูปแบบ/วิธีการดำเนินงาน ขั้นตอน และระยะเวลาดำเนินงาน | | | | | | | | | | | | |
| 1.2 ศึกษา สำรวจ และรวบรวมข้อมูลที่เกี่ยวข้อง เช่น สถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้อง อันเป็นประโยชน์ต่อการสร้างองค์ความรู้ในเทคโนโลยีการจำลองระบบเสมือนจริงเชิงลึก | | | | | | | | | | | | |
| **WP2:** ดำเนินการวิจัย ออกแบบ และพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง | | | | | | | | | | | | |
| 2.1 ดำเนินการศึกษา สำรวจ และวิจัยความเป็นไปได้ในการเลือกใช้ open source software tools ต่างๆ ที่เกี่ยวข้อง | | | | | | | | | | | | |
| 2.2 วิจัย พัฒนา และออกแบบสถาปัตยกรรมแพลตฟอร์มเชิงภาพรวม | | | | | | | | | | | | |
| 2.3 ออกแบบและพัฒนาระบบในส่วนที่เป็น NFV infrastructure layer | | | | | | | | | | | | |
| 2.4 ออกแบบและพัฒนาระบบในส่วนที่เป็น VNFaaS layer | | | | | | | | | | | | |
| 2.5 ออกแบบและพัฒนาระบบในส่วนที่เป็น NFV MANO layer | | | | | | | | | | | | |
| 2.6 ออกแบบและพัฒนา web portal และส่วนติดต่อกับผู้ใช้งาน (API) เพื่อการเชื่อมโยงบริการทั้งหมดในระบบให้รวมศูนย์อยู่ในช่องทางเดียวกัน | | | | | | | | | | | | |
| | ปีที่ 2 | | | | | | | | | | | |
| **WP3:** ทดสอบประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน | | | | | | | | | | | | |
| 3.1 การทำงานเชิงหน้าที่ (functionality) | | | | | | | | | | | | |
| 3.2 ด้านการใช้งาน (feasibility) | | | | | | | | | | | | |
| 3.3 ด้านประสิทธิภาพ (performance) | | | | | | | | | | | | |
| 3.4 จัดทำรายงานสรุปผล | | | | | | | | | | | | |
| **WP4:** ถ่ายทอดความรู้เชิงวิจัยและนวัตกรรม | | | | | | | | | | | | |
| 4.1 จัดกิจกรรมสัมมนาแนะนำแพลตฟอร์ม | | | | | | | | | | | | |
| 4.2 ประสานงานกลุ่มเป้าหมายเพื่อนำแพลตฟอร์มไปติดตั้ง ทดสอบและใช้งานจริง | | | | | | | | | | | | |
| 4.3 สรุปผลการดำเนินงาน และจัดทำรายงานฉบับสมบูรณ์ (Final report) | | | | | | | | | | | | |

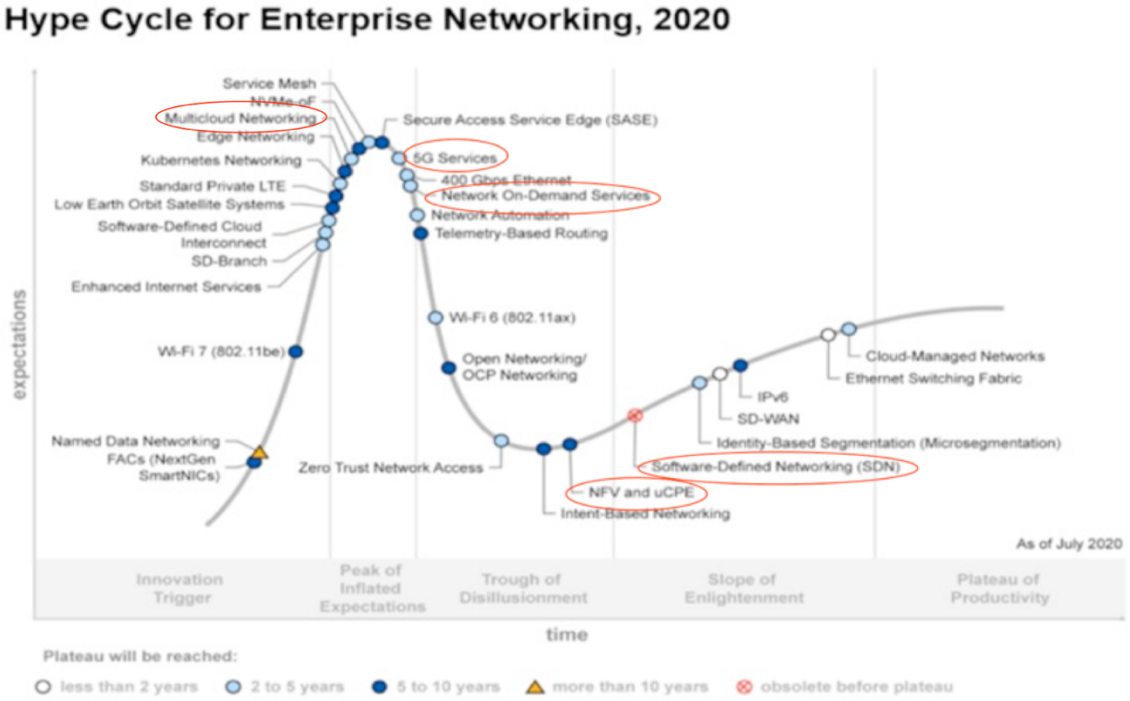
## บทที่ 2

## กิจกรรมที่ 1: ดำเนินการศึกษา สำรวจ และรวบรวมข้อมูลที่เกี่ยวข้อง เช่น สถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้อง อันเป็นประโยชน์ต่อการสร้างองค์ความรู้และต่อยอดในเทคโนโลยีการจำลองระบบเสมือนจริงเชิงลึก

### 2.1 บทนำ

ทุกวันนี้ เทคโนโลยีดิจิทัลได้เข้ามามีบทบาทสำคัญและสร้างผลกระทบอย่างมากต่อทุกแง่มุมของการดำรงชีวิต เห็นได้จากการดำรงชีวิตประจำวันของมนุษย์ส่วนใหญ่มักจะรายล้อมด้วยเทคโนโลยีดิจิทัล ดังเช่น (1) ความต้องการใช้งานอย่างแพร่หลายของเครื่องคอมพิวเตอร์ ซึ่งนับว่าเป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีความจำเป็นอย่างมากต่อการปฏิบัติงานในสาขาวิชาชีพต่างๆ รวมไปถึงเพื่อความบันเทิงและการติดต่อสื่อสาร (2) การแสวงหาประโยชน์จากเทคโนโลยีดิจิทัลเพื่อการคำนวน ประมวลผล รวบรวมข้อมูลขนาดใหญ่ และช่วยในการตัดสินใจ (data-driven decision making) (3) การปฏิสัมพันธ์ระหว่างอุปกรณ์กับคน นับว่าความก้าวหน้าของเทคโนโลยีไม่เพียงเชื่อมต่อผู้คนเท่านั้น แต่ยังรวมถึงการเพิ่มขีดความสามารถให้อุปกรณ์ (Internet of Things: IoTs) หรือเครื่องจักร (Machine to machines: M2M) สามารถเชื่อมต่อเข้ากับเครือข่าย เพื่อส่งเสริมให้เกิดการรวบรวม วิเคราะห์ ตัดสินใจ และประมวลผลข้อมูลได้อย่างมีประสิทธิภาพ (4) ส่งเสริมให้เกิด ‘เศรษฐกิจแบ่งปัน’ (sharing economy) ซึ่งเป็นรูปแบบเศรษฐกิจที่ไม่ได้เน้นการเติบโตแบบลุยเดี่ยว แต่อาศัยการแบ่งปันข้อมูลและความร่วมมือจากหลายๆ กลุ่ม เช่น กลุ่มผู้ให้บริการแพลตฟอร์ม กลุ่มผู้ใช้บริการ และกลุ่มผู้รับบริการหรือผู้ใช้งาน เพื่อทำให้เกิดรูปแบบบริการใหม่ที่ตอบรับความต้องการของผู้คนในวงกว้าง อย่างไรก็ตามการได้มาซึ่งระบบบริการ นวัตกรรม และสังคมดิจิทัลที่เข้มแข็งนั้นย่อมเกิดจาก 3 ปัจจัยหลักๆ คือ (1) การวางรากฐานการพัฒนาโครงสร้างพื้นฐานดิจิทัลที่ทันสมัยและมั่นคง (2) การใช้ประโยชน์จากเทคโนโลยีมาช่วยเพิ่มประสิทธิภาพการผลิตและเพิ่มมูลค่าให้แก่สินค้าและการบริการ ควบคู่ไปกับ (3) การสร้างกำลังคนด้าน S&T ให้พร้อมเพื่อสร้างความเข้มแข็งของเศรษฐกิจและเพิ่มขีดความสามารถในการยกระดับนวัตกรรมให้มีความเทียบเท่ากับตลาดโลกได้ ดังนั้นการอ้างอิงผลสำรวจ Gartner Hype Cycle [1] (รูปที่ 2.1.1) มีส่วนช่วยอยากมากในการล่วงรู้ถึงทิศทางเทคโนโลยีและนวัตกรรมที่กำลังจะเกิดขึ้นและมีอิทธิพลต่อการขับเคลื่อนเศรษฐกิจและสังคมในอนาคตอันใกล้ อันเป็นประโยชน์สำคัญต่อการกำหนดแผนพัฒนากำลังคนและการวางรากฐานโครงสร้างพื้นฐานดิจิทัลในประเทศต่อไป

รูปที่ 2.1.1 ชี้ให้เห็นว่ากลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ดังเช่น การจำลองฟังก์ชันเครือข่ายเสมือน (***Network Function Virtualization: NFV***) การจัดการเครือข่ายด้วยซอฟต์แวร์ (***Software-Defined Networking: SDN***) เครือข่ายมัลติคลาวด์ (***multi-cloud networking***) กลายเป็นเป็นเสาหลักสำคัญในการขับเคลื่อนโครงสร้างพื้นฐานระบบเครือข่ายและการให้บริการเซอร์วิสแบบดั้งเดิม (traditional networking model) ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดด้วยซอฟต์แวร์ (Software-defined infrastructure) เพื่อรองรับสภาพแวดล้อม 5G ในอนาคตในทุกๆ มิติของภาคอุตสาหกรรม (***5G services***) ดังเช่น smart home, smart city, e-health, industry automation, mission-critical services, เหล่านี้เป็นต้น นำมาซึ่งการสร้างกระบวนทัศน์ใหม่ในการให้บริการเน็ตเวิร์คเซอร์วิสและการจัดการเครือข่ายรูปแบบใหม่ ที่สามารถช่วย

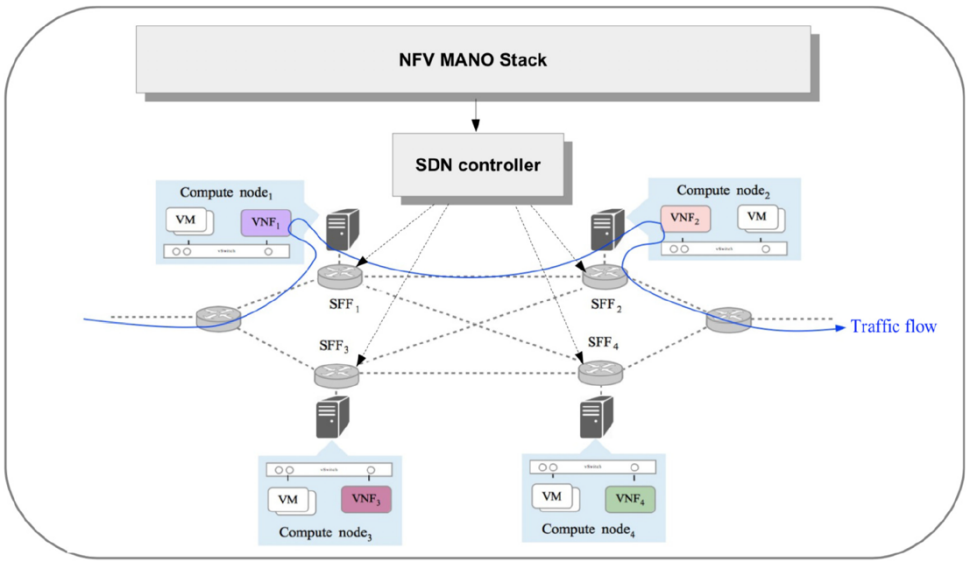



***รูปที่ 2.1.1:*** *ผลสำรวจ Gartner Hype Cycle ปี ๒๕๖๓ – ถึงการวิเคราะห์ทิศทางและแนวโน้มของเทคโนโลยีโครงข่าย (enterprise networking) ที่เกิดใหม่และกำลังสร้างความได้เปรียบทางการแข่งขันในตลาดระดับสูงในอีก 5-10 ปีข้างหน้า*

ลดข้อจำกัดการให้บริการเครือข่ายแบบดั้งเดิมได้ ไม่ว่าจะเป็น (1) การลดค่าใช้จ่ายด้านการลงทุนและค่าดำเนินการ (2) ความสามารถในการตอบสนองการให้บริการและการจำลองเน็ตเวิร์คเซอร์วิสอย่างทันถ่วงที (3) ประสิทธิภาพในการประมวลผล (4) สร้างความยืดหยุ่นในการจัดสรรหรือขยายทรัพยากรเครือข่ายให้มีความสอดคล้องกับปริมาณงานในช่วงเวลานั้นๆ ตลอดจน (5) การให้อิสระต่อการเลือกใช้งานเน็ตเวิร์คเซอร์วิสประเภทต่างๆ ตามบริบทขององค์กรหรือตามความต้องการใช้งานในแต่ละภาคธุรกิจได้ (***network on-demand services***)

เทคโนโลยี NFV [2-4] และ SDN [5-6] นับว่าเป็นเทคโนโลยีใหม่ที่เพิ่งเกิดขึ้นในช่วงสิบปีที่ผ่านมา ทั้งสองเทคโนโลยีนี้มีความเกี่ยวข้องกันอย่างมีนัยยะสำคัญในการขับเคลื่อนและเปลี่ยนแปลงโครงสร้างเครือข่ายแบบดั้งเดิมไปสู่โครงสร้างเครือข่ายยุคใหม่ แม้ว่าเทคโนโลยี NFV และเทคโนโลยี SDN มีหน้าที่ในการบริหารและจัดการเครือข่ายในระดับของซอฟต์แวร์ที่เหมือนกัน แต่ทั้งสองเทคโนโลยีนี้จะทำงานในระดับชั้นเครือข่ายที่ต่างกันเพื่อเกื้อหนุนการทำงานร่วมกัน ซึ่งวัตถุประสงค์หลักของเทคโนโลยี NFV คือ การแยกฟังก์ชั่นบริการด้านเครือข่าย (network function) ในส่วนของซอฟต์แวร์ซึ่งเดิมเคยผูกติดกับฮาร์ดแวร์เฉพาะ (proprietary hardware) ของแต่ละค่ายผู้ผลิตออกมาให้กลายเป็นซอฟต์แวร์ที่สามารถรันได้บนเครื่องเสมือนจริง (Virtual Machines: VMs) บนระบบคลาวด์แพลตฟอร์มได้ หรือที่เรียกว่า การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (Virtualized Network Functions: VNFs) ซึ่งข้อดีของเทคโนโลยี NFV จะช่วยเพิ่มประสิทธิภาพการทำงานให้ดียิ่งขึ้น ทั้งในแง่ของความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส ความยืดหยุ่นในการจัดสรรทรัพยากรเครือข่าย และประสบการณ์ที่ผู้ใช้จะได้รับ ขณะที่แนวคิดของเทคโนโลยี SDN จะช่วยสนับสนุนเทคโนโลยี NFV ในเรื่องการบริหารจัดการ ควบคุมและกำหนดการเชื่อมต่อเครือข่ายด้วยซอฟต์แวร์ เช่น เข้าถึงการตั้งค่าและควบคุมการรับส่งข้อมูลบนเครือข่าย กำหนด

เส้นทางการส่งต่อแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสต่างๆ ที่ได้ถูกติดตั้งขึ้น ควบคุมการไหลของแพ็กเก็ตให้ตรงตามข้อกำหนดหรือนโยบายการรักษาความปลอดภัย (security policy) ที่ได้กำหนดไว้ ฯลฯ ทำให้ผู้ให้บริการหรือผู้ที่เกี่ยวข้องสามารถบูรณาการเส้นทางการเชื่อมต่อเครือข่ายและกำหนดเส้นทางการส่งต่อแพ็กเก็ตไปยังเซอร์วิสต่างๆ ที่ต้องการได้อย่างยืดหยุ่นเพื่อให้เหมาะสมกับการขยายตัวและความต้องการใช้งานที่หลากหลายของแต่ละกลุ่มผู้บริโภคได้ รูปที่ 2.1.2 แสดงภาพรวมของการทำงานร่วมกันระหว่างเทคโนโลยี NFV และ SDN เพื่อยกระดับประสิทธิภาพการบริหารจัดการเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน ทั้งในเชิงความคุ้มทุน ความคล่องตัว ความรวดเร็วในการประมวลผล และประสิทธิภาพในการดำเนินการ




***รูปที่ 2.1.2:*** *: แสดงโครงสร้างภาพรวมของเทคโนโลยี NFV และ SDN กับการทำงานเกื้อหนุนกัน โดยมี (1) ส่วนจัดการและประสานงาน (NFV Management and Orchestration: NFV MANO Stack) ทำหน้าที่จัดสรรทรัพยากรเครือข่าย (network resources) และจำลอง เน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ประเภทต่างๆ เช่น virtual firewalls, virtual switches เป็นต้น และ (2) ส่วนควบคุม (SDN controller) ทำหน้าที่กำหนดเส้นทางการส่งต่อแพ็กเก็ตบนเครือข่ายไปยัง VNFs ที่ได้ถูกจำลองไว้ให้ตรงตามข้อกำหนดและนโยบายการรักษาความปลอดภัยที่ได้กำหนดไว้*

## 2.2 ข้อจำกัดของโครงสร้างเครือข่ายแบบดั้งเดิม

ปัจจุบันนี้ เทคโนโลยีเครือข่ายได้ถูกพัฒนาขึ้นอย่างต่อเนื่องเพื่อตอบสนองความต้องการทางตลาดที่มีแนวโน้มความต้องการใช้งานที่เพิ่มสูงขึ้นเรื่อยๆ ผู้บริโภคคาดหวังว่าเทคโนโลยีเครือข่ายสามารถรับรองการขยายตัวที่เพียงพอต่อความต้องการขององค์กรและปริมาณการรับส่งข้อมูลที่ไหลเข้ามาอย่างต่อเนื่องและมหาศาลได้ (เช่น ธุรกิจในยุค 5G และอุตสาหกรรม 4.0) มีความคาดหวังว่าเทคโนโลยีเครือข่ายนั้นสามารถตอบสนองความต้องการได้อย่างทันถ่วงที (Rapid service deployment) ให้ผลลัพธ์การประมวลผลข้อมูลได้อย่างมีประสิทธิภาพ (Operational efficiency) และสามารถเรียกใช้งานเน็ตเวิร์คเซอร์วิสประเภทต่างๆ ได้อย่างยืดหยุ่นเพื่อให้เหมาะสมกับความต้องการใช้งานที่เฉพาะเจาะจง (On demand service delivery) ตามราคาที่เหมาะสม (Reduction in CapEx and OptExt) ) ปัจจัยเหล่านี้ส่งผลให้โครงสร้างเครือข่ายแบบดั้งเดิม ซึ่งมีข้อจำกัดดัง

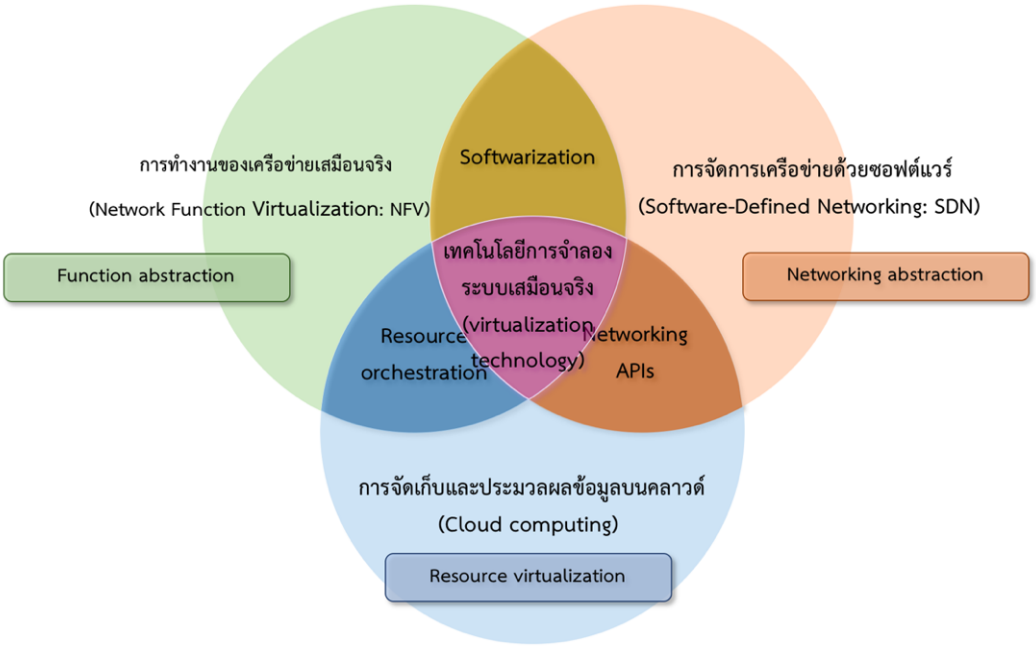
ด้านล่างนี้ [7] กลายเป็นสาเหตุหลักที่ทำให้ระบบเครือข่ายเข้าสู่สภาวะหยุดนิ่ง ไม่มีความคล่องตัว และขาดความยืดหยุ่นที่จะสามารถตอบสนองความต้องการของผู้บริโภคในปริมาณที่เพิ่มสูงขึ้นได้

- **ค่าใช้จ่าย (Expense):** โครงสร้างเครือข่ายแบบดั้งเดิมมีการเชื่อมต่ออุปกรณ์เครือข่ายหลายๆ ตัวเข้าด้วยกัน (เช่น routers, firewall, NAT, load balancer, IDS, IPS) ซึ่งแต่ละตัวมีวัตถุประสงค์หรือรูปแบบการทำงานที่แตกต่างกันออกไป ดังนั้น ในการจัดตั้งโครงสร้างเครือข่ายภายในองค์กรแต่ละครั้ง องค์กรหนึ่งๆ จำเป็นต้องสูญเสียรายจ่ายจำนวนมากสำหรับการลงทุนจัดซื้ออุปกรณ์เครือข่าย ติดตั้ง ดำเนินการตลอดจนค่าบำรุงรักษาระบบเครือข่ายในระยะยาว โดยส่วนใหญ่แล้วค่าใช้จ่ายมักจะแปรผันตามขนาดของระบบเครือข่าย กล่าวคือ ยิ่งจำนวนของอุปกรณ์เครือข่ายยิ่งมาก ค่าใช้จ่ายในการจัดซื้อ ซ่อมแซม และบำรุงรักษาก็ยิ่งสูงขึ้นตาม ในบทความ [8] ได้ยกตัวอย่างให้เห็นว่า สำหรับองค์กรขนาดกลางที่มีเครื่องคอมพิวเตอร์และอุปกรณ์เครือข่าย 1,000 – 10,000 เครื่อง จะต้องใช้เงินลงทุนประมาณ 50,000 – 500,000 ดอลล่าร์สหรัฐ ในขณะที่เครือข่ายขนาดใหญ่ที่มีเครื่องคอมพิวเตอร์และอุปกรณ์เครือข่ายมากกว่า 100,000 เครื่องจะต้องใช้เงินลงทุนมากกว่าหลักล้านดอลล่าร์สหรัฐ
- **ความยุ่งยากซับซ้อน (Complexity):** อีกหนึ่งปัญหาหลักสำคัญที่พบในโครงสร้างเครือข่ายแบบดั้งเดิม คือ ความยุ่งยากและความซับซ้อนในการบริหารและจัดการระบบเครือข่าย เนื่องจากปัญหาการผูกขาดทางเทคโนโลยีและการพัฒนาโพรโตคอลบนแต่ละอุปกรณ์เครือข่ายที่แตกต่างกัน โดยส่วนใหญ่แล้วอุปกรณ์เครือข่ายแต่ละมักจะขึ้นตรงกับผู้ผลิต ข้อจำกัดเหล่านี้ล้วนส่งผลให้การเข้าถึงและจัดการระบบทำได้ยากและมีข้อจำกัด
- **ความไม่ยืดหยุ่น (Elasticity):** ความไม่ยืดหยุ่นและความไม่สอดคล้องกันของนโยบายบนอุปกรณ์เครือข่ายแต่ละตัว และขาดมาตรฐานกลางในการพัฒนา ล้วนเป็นข้อจำกัดที่ทำให้ระบบเครือข่ายเข้าสู่สภาวะหยุดนิ่ง ไม่มีความคล่องตัว และขาดความยืดหยุ่นที่จะสามารถตอบสนองความต้องการของผู้บริโภคในปริมาณที่เพิ่มสูงขึ้นได้
- **การใช้ทรัพยากร (Resource utilization):** โดยส่วนใหญ่แล้วอุปกรณ์เครือข่ายที่ได้ถูกติดตั้งบนโครงสร้างเครือข่ายแบบดั้งเดิมนั้นมีรูปแบบการติดตั้งแบบ stand-alone และมีลักษณะการทำงานแบบแยกส่วนเพื่อตอบโจทย์งานด้านใดด้านหนึ่งที่เฉพาะเจาะจง (one-off solution) กล่าวคือ แม้ว่าเป็นอุปกรณ์เครือข่ายประเภทเดียวกัน แต่อุปกรณ์เครือข่ายทั้งสองเครื่องนี้โดยส่วนใหญ่แล้วมักจะทำงานอิสระต่อกัน ซึ่งข้อจำกัดนี้ส่งผลให้เกิดการใช้งานทรัพยากรฮาร์ดแวร์ที่ขาดประสิทธิภาพและขาดความยืดหยุ่น เนื่องจากโครงสร้างเครือข่ายแบบดั้งเดิมไม่เอื้อประโยชน์ให้ผู้ดูแลระบบสามารถปรับขนาดของทรัพยากรฮาร์ดแวร์ให้เหมาะสมตามปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ ได้

### 2.3 เทคโนโลยีที่เกี่ยวข้อง

โดยทั่วไปกลุ่มเทคโนโลยีที่ช่วยในการขับเคลื่อนการจำลองระบบเสมือนจริง ประกอบด้วย 3 เทคโนโลยีหลักๆ ดังต่อไปนี้ (แสดงดังรูปที่ 2.3.1)

### 3.3.1 การจัดเก็บและประมวลผลข้อมูลบนคลาวด์ (Cloud computing)



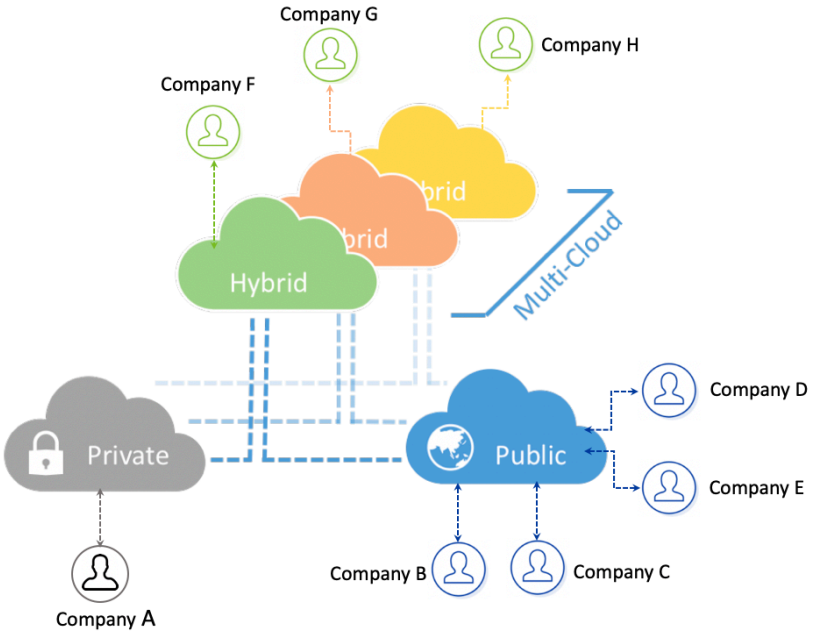

*รูปที่ 2.3.1: กลุ่มเทคโนโลยีที่ช่วยในการขับเคลื่อนการจำลองระบบเสมือนจริง (Virtualization technology)*

เทคโนโลยีคลาวด์ [9-11] คือ ระบบเก็บข้อมูลที่นำเอาข้อมูลทั้งหมดขึ้นไปเก็บไว้บนระบบอินเทอร์เน็ต โดยผู้ใช้สามารถเรียกข้อมูลออกมาใช้งานได้ตลอดเวลาที่ต้องการ ดังนั้นการให้บริการคลาวด์จะคลอบคลุมถึงการให้ใช้/เช่าทรัพยากรด้านคอมพิวเตอร์จากผู้ให้บริการทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการประมวลผล การจัดเก็บข้อมูลและระบบออนไลน์ต่างๆ ซึ่งผู้ใช้สามารถเลือกกำลังการประมวลผล และจำนวนทรัพยากรได้ตามความเหมาะสมต่อการใช้งาน ซึ่งแนวคิดของเทคโนโลยีคลาวด์ คือ การนำเอาเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงหลายๆ เครื่องมาเชื่อมต่อกันผ่านระบบเครือข่ายอินเทอร์เน็ต เพื่อให้เกิดการทำงานร่วมกัน ในกรณีที่เครื่องเซิร์ฟเวอร์เครื่องใดเครื่องหนึ่งเกิดความเสียหายหรือไม่สามารถทำงานได้ ก็จะไม่ส่งผลกระทบต่อเครื่องเซิร์ฟเวอร์อื่นๆ ที่อยู่ภายในระบบคลาวด์ ด้วยเหตุนี้ เทคโนโลยีคลาวด์จัดได้ว่าเป็นเทคโนโลยีหนึ่งที่กำลังมาแรงและได้รับความนิยมเพิ่มขึ้นอย่างต่อเนื่องทั้งในธุรกิจ Startup SMEs หรือองค์กรขนาดเล็ก กลาง และใหญ่ ด้วยคุณสมบัติเด่นที่ช่วยลดความยุ่งยากในการติดตั้ง ดูแลระบบ ช่วยประหยัดเวลาและต้นทุนในการสร้างระบบคอมพิวเตอร์และเครือข่ายเอง โดยผู้ใช้บริการสามารถเข้าถึงระบบคลาวด์และข้อมูลต่างๆ จากที่ไหนก็ได้ผ่านอินเทอร์เน็ต ทำให้ง่ายต่อการเข้าถึงและบริหารจัดการ โดยทั่วไปแล้วรูปแบบการให้บริการของเทคโนโลยีคลาวด์แบ่งออกเป็น 3 ประเภทหลักๆ (ดังแสดงในรูปที่ 2.3.2) คือ

- **Private cloud:** คือการตั้งคลาวด์ส่วนตัว โดยแต่ละบริษัทหรือองค์กรลงทุนจัดซื้อฮาร์ดแวร์และซอฟต์แวร์ที่ใช้เป็นพื้นฐานในการทำ cloud datacenter ขึ้นมาเป็นของตัวเองเพื่อให้พนักงานในองค์กรใช้งานเท่านั้น ทำให้การใช้งานและการเข้าถึงข้อมูลบนคลาวด์มีความเป็นส่วนตัวสูง โดยจุดเด่นของ private cloud คือ (1) ผู้ใช้สามารถควบคุมระบบการเข้าถึงความปลอดภัยและกำหนดการตั้งค่าต่างๆ ได้อย่างอิสระ (2) มั่นใจได้ว่าข้อมูลมีความปลอดภัย เนื่องจากระบบฮาร์ดแวร์และซอฟต์แวร์มีความเป็นส่วนตัวสูง (3) ลดปัญหาการเชื่อมต่อและลดระยะเวลาการตอบสนอง (latency) เนื่องจากแบนด์วิธไม่ได้ถูกใช้หรือแบ่งปันให้ใคร

ทำให้การใช้งานมีความเป็นส่วนตัวสูง (4) องค์กรสามารถบริหารจัดการทรัพยากรทั้งหมดได้ด้วยตนเองผ่านการควบคุมบนแพลตฟอร์ม และสามารถเข้าถึงบริการได้จากทุกที่ ด้วยการเชื่อมต่อข้อมูลผ่านเครือข่ายอินเทอร์เน็ต ผ่านผู้ให้บริการจึงไม่มีข้อจำกัดเรื่องสถานที่และอุปกรณ์ (5) ประหยัดการลงทุนทรัพยากรคอมพิวเตอร์ เพราะใช้รูปแบบบริการแบบเช่าผ่านผู้ให้บริการ สามารถใช้บริการได้เท่าที่ต้องการโดยไม่จำเป็นต้องติดตั้ง พัฒนาและขึ้นระบบเองทั้งหมด (6) สามารถสร้างระบบใหม่ขึ้นมาทันที ผู้ให้บริการจะจัดเตรียมทรัพยากรเพื่อรองรับผู้ใช้บริการ การสร้างและลบจึงสามารถทำได้ทันที โดยไม่มีข้อจำกัดเรื่องเวลา (7) เพิ่มขนาดทรัพยากรได้ง่ายดายและรวดเร็ว ตามปริมาณการใช้งานหรือลดขนาดทรัพยากรตามที่ผู้ใช้งานต้องการ และ (8) ลดปัญหาการดูแลระบบ เนื่องจากบริการเป็นรูปแบบของการเช่าใช้ จึงเป็นหน้าที่ของผู้ให้บริการที่จะต้องดูแลระบบแทน ซึ่งจะช่วยลดทั้งความยุ่งยากของการดูแลและลดจำนวนบุคลากรที่ต้องจ้างมาเพื่อมาดูแลระบบ

- **Public cloud:** คือคลาวด์ที่มีระบบการประมวลผลหรือการจัดเก็บข้อมูลของผู้ใช้บนเครื่องเซิร์ฟเวอร์สาธารณะ ซึ่งผู้ให้บริการจะมีการเก็บข้อมูลของผู้ใช้งานแบบสาธารณะไว้ในเครื่องเซิร์ฟเวอร์เดียวกัน แต่ผู้ใช้งานจะไม่สามารถเข้าถึงข้อมูลของผู้อื่นได้หากไม่ได้รับอนุญาต โดยจะมีความปลอดภัยในการใช้งานระดับหนึ่ง ในอีกมุมหนึ่งผู้ให้บริการระบบคลาวด์จะเป็นคนจัดตั้งระบบฮาร์ดแวร์และซอฟต์แวร์ขึ้นมา แล้วให้แต่ละบริษัทหรือองค์กรสามารถเช่าใช้บริการ โดยจ่ายค่าเช่าระบบฯเป็นรายเดือนหรือรายปีแทน ซึ่งจุดเด่นของ public cloud คือ (1) ลดต้นทุนค่าใช้จ่ายสำหรับการติดตั้ง ดูแล และบำรุงรักษา (2) สามารถเพิ่มหรือลดทรัพยากรการใช้งานคลาวด์ได้ตามที่ต้องการผ่านการสั่งงานบนแพลตฟอร์มใช้งาน (3) ลดระยะเวลา downtime หรือปัญหาระบบล่ม เนื่องจากผู้ให้บริการเช่าคลาวด์ส่วนใหญ่มีมาตรฐานการบริการรองรับ (4) มีเจ้าหน้าที่คอยช่วยเหลือและอำนวยความสะดวกให้ตลอดเวลา 24 ชม.ต่อวัน (5) สามารถบริหารและจัดสรรทรัพยากรทั้งหมดขององค์กรได้ด้วยตนเองผ่านการควบคุมบนแพลตฟอร์ม

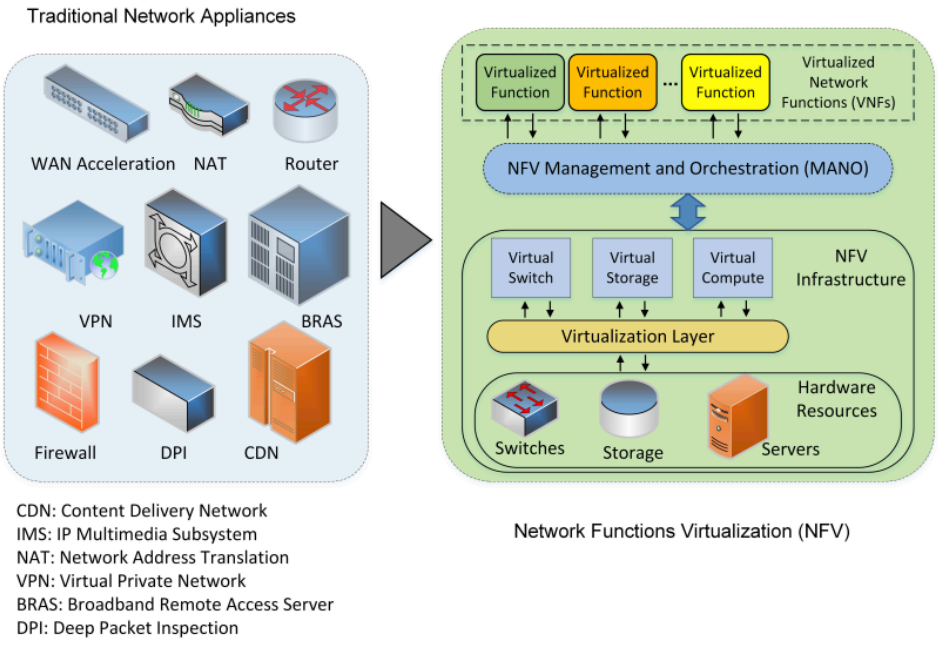



*รูปที่ 2.3.2: รูปแบบการให้บริการเทคโนโลยีคลาวด์*

- **Hybrid cloud:** คือระบบคลาวด์ที่ใช้งานร่วมกันระหว่าง private cloud และ public cloud ทำให้องค์กรสามารถบริหารและจัดการแอปพลิเคชั่นหรือข้อมูลต่างๆ ภายในองค์กรได้ทั้งส่วนตัวและสาธารณะ เพื่อให้มีการทำงานที่คล่องตัวมีความปลอดภัยมากขึ้น เช่น การนำ private cloud มาใช้สำหรับเก็บข้อมูลในองค์กร และใช้ public cloud มาใช้เพื่อการ scale out ในการประมวลผลในช่วงเวลาที่คาดว่าจะเกิดการใช้งานสูงสุดของวัน (peak load) โดยจุดเด่นของ hybrid cloud คือ (1) ลดต้นทุนบางส่วนในการจัดการให้น้อยลง เนื่องจากไม่ต้องลงทุนกับ private cloud ทั้งหมด (2) สร้างความยืดหยุ่นและทำให้เกิดการใช้งานที่มีประสิทธิภาพมากขึ้น โดยการลดข้อจำกัดของการเลือกใช้งานคลาวด์ประเภทใดประเภทหนึ่ง

### 2.3.2 การทำงานของเครือข่ายเสมือนจริง (Network Function Virtualization: NFV)

เทคโนโลยี NFV [2-4] และเทคโนโลยี SDN [5-6] นับว่าเป็นเทคโนโลยีใหม่ที่เพิ่งเกิดขึ้นในช่วงสิบปีที่ผ่านมา ทั้งสองเทคโนโลยีนี้มีความเกี่ยวข้องกันอย่างมีนัยยะสำคัญในการขับเคลื่อนและเปลี่ยนแปลงโครงสร้างเครือข่ายแบบดั้งเดิมไปสู่โครงสร้างเครือข่ายยุคใหม่ แม้ว่าเอ็นเอฟวี (NFV) และเอสดีเอ็น (SDN) มีหน้าที่ในการบริหารและจัดการเครือข่ายในระดับของซอฟต์แวร์ที่เหมือนกัน แต่ทั้งสองเทคโนโลยีนี้จะทำงานในระดับชั้นเครือข่ายที่ต่างกันเพื่อเกื้อหนุนการทำงานร่วมกัน โดยวัตถุประสงค์หลักของเทคโนโลยี NFV คือ การแยกเน็ตเวิร์คเซอร์วิส (network services) ในส่วนของซอฟต์แวร์ซึ่งเดิมเคยผูกติดกับฮาร์ดแวร์เฉพาะ (proprietary hardware) ของแต่ละค่ายผู้ผลิตออกมาให้กลายเป็นซอฟต์แวร์ที่สามารถรันได้บนเครื่องเสมือนจริงบนระบบคลาวด์แพลตฟอร์มได้ หรือที่เรียกว่า การให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (Virtualized Network Functions: VNFs) ดังแสดงในรูปที่ 2.3.3 ซึ่งข้อดีของเทคโนโลยี NFV คือ มีรูปแบบการจัดสรรทรัพยากรเครือข่ายแบบไดนามิก รองรับความยืดหยุ่นในการจัดสรรทรัพยากรเครือข่ายเพื่อให้มีความสอดคล้องตามปริมาณและภาระงานที่เข้ามาในระบบ ความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิสและการรันงาน ทั้งหมดนี้ล้วนส่งผลต่อการช่วยเพิ่มประสิทธิภาพการทำงานในระบบ




*รูปที่ 2.3.3: การจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริงด้วยเทคโนโลยี NFV*

เครือข่ายได้ดียิ่งขึ้น ทั้งในแง่ของความเร็วในการประมวลผล ความยืดหยุ่นในการจัดสรรทรัพยากรเครือข่าย และประสบการณ์ที่ผู้ใช้จะได้รับ นอกเหนือจากนี้ หากอ้างอิงสถาปัตยกรรมการออกแบบ NFV ตามมาตรฐานที่ออกโดยสถาบันมาตรฐานโทรคมนาคมยุโรป (European Telecommunications Standards Institute: ETSI) [12] (ดังรูปที่ 2.3.3 ขวามือ) เทคโนโลยี NFV ประกอบด้วย 3 ระดับชั้น ได้แก่

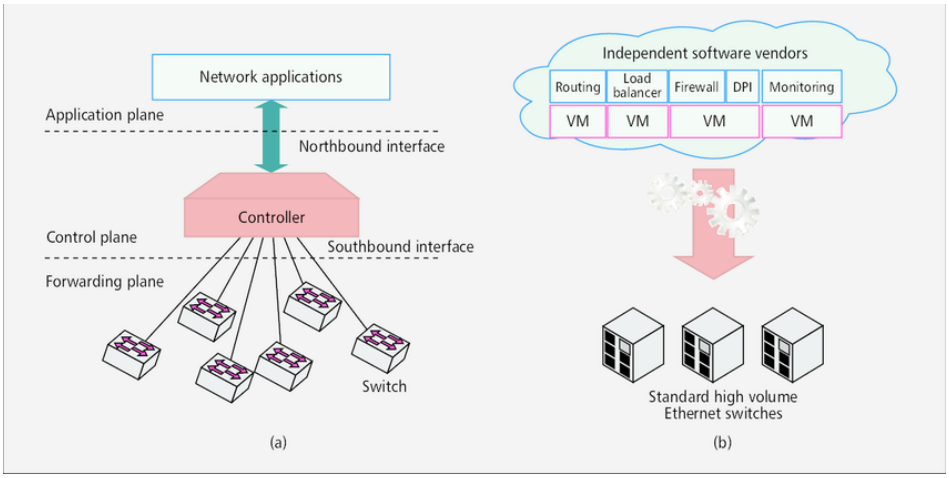
- **ชั้นจัดเตรียมทรัพยากรโครงสร้างพื้นฐาน (NFV Infrastructure: NFVI):** ทำหน้าที่ในการจัดเตรียมโครงสร้างพื้นฐานและทรัพยากรเครือข่ายต่างๆ ที่จำเป็นต้องใช้ในการจำลองเน็ตเวิร์คเซอร์วิส ซึ่งคลอบคลุมถึง ระบบประมวลผล ระบบเครือข่าย และแหล่งจัดเก็บข้อมูลต่างๆ โดยหน้าที่หลักๆ ของเลเยอร์นี้ คือ (1) จัดสรรทรัพยากรเครือข่ายตามการร้องขอ (2) จัดเตรียมเครื่อง VMs ให้พร้อมสำหรับจำลอง เน็ตเวิร์คเซอร์วิส และ (3) ดำเนินการตรวจสอบและวิเคราะห์ประสิทธิภาพการใช้งานทรัพยากรเครือข่าย
- **ชั้นจำลองเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNF):** ทำหน้าสร้างและจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส (VNF) ให้มีขีดความสามารถในการทำงานบนทรัพยากรโครงสร้างพื้นฐานที่ได้กำหนดขึ้น ซึ่งหน้าที่หลักๆ ของเลเยอร์ คือ (1) สร้างและจำลองเน็ตเวิร์คเซอร์วิสประเภทต่างๆ (2) ดำเนินการตั้งค่าแต่ละเน็ตเวิร์คเซอร์วิสที่ถูกจำลองขึ้นตามความต้องการใช้งานหรือตามนโยบาย (policy) ที่ผู้ดูแลระบบกำหนดไว้ (3) ทำการปรับลดหรือขยายเน็ตเวิร์คเซอร์วิส (scaling in/out VNFs) เมื่อประสิทธิภาพของทรัพยากรที่รันงานเกินกว่าค่า threshold ที่กำหนดไว้ (4) อัพเกรดซอฟต์แวร์ที่เกี่ยวข้อง และ (5) ถอดถอนการติดตั้งเน็ตเวิร์คเซอร์วิสนั้นๆ เมื่อไม่มีการใช้งาน
- **ชั้นบริหารและจัดการประสาน (NFV Management and Orchestration: NFV MANO)** ทำหน้าที่ในการควบคุมลำดับการทำงานของระบบ และประสานให้การทำงานเป็นไปอย่างอัตโนมัติสอดคล้องกับบริบทความต้องการใช้งานหรือนโยบายที่กำหนดไว้ โดยหน้าที่หลักๆ ของ NFV MANO คือ (1) บริหารและจัดการเน็ตเวิร์คโทโพโลยีในภาพรวม (2) ตรวจสอบสิทธิการเข้าถึงทรัพยากรเครือข่ายเมื่อมีการร้องขอ (3) ประสานงานของหน่วย NFVI เพื่อจัดสรรทรัพยากรเครือข่ายและหน่วย VNF เพื่อจำลองเน็ตเวิร์คเซอร์วิสตามการร้องขอ และ (4) บริหารและจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น

### 2.3.3 การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN)

เทคโนโลยี SDN [5-6] เป็นเทคโนโลยีที่คอยสนับสนุนเทคโนโลยี SDN เพื่อช่วยให้ผู้ดูแลระบบสามารถบริหาร จัดการ ควบคุม และตั้งค่าทรัพยากรเครือข่ายได้อย่างมีประสิทธิภาพ โดยหน้าที่หลัก คือ (1) เข้าถึงการตั้งค่าและควบคุมการรับส่งข้อมูลบนเครือข่าย (full network control capabilities) (2) กำหนดเส้นทางการส่งต่อแพ็กเก็ต เพื่อให้สอดคล้องกับภาระงานของอุปกรณ์เครือข่ายนั้นๆ (routing decision) และ (3) ควบคุมการไหลของแพ็กเก็ต (traffic steering and forwarding) ให้ตรงตามข้อกำหนดหรือนโยบายที่กำหนดไว้ ด้วยคุณสมบัติเด่นของเทคโนโลยี SDN ที่กล่าวมาข้างต้นนี้ ทำให้ผู้ดูแลระบบสามารถบูรณาการเส้นทางการเชื่อมต่อเครือข่ายและกำหนดเส้นทางการส่งต่อแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสที่ต้องการได้อย่างไดนามิกและยืดหยุ่น รูปที่ 2.2.4 แสดงสถาปัตยกรรมการออกแบบเทคโนโลยี SDN ซึ่งแยก ส่วนควบคุม (control plane) ออกจาก ส่วนของการรับส่ง

ข้อมูล (forwarding plane) แต่เดิมทีการให้บริการเครือข่ายจะผูกติดกับระบบฮาร์ดแวร์เฉพาะ โดยมีส่วนของการจัดการในการเชื่อมต่อทั้งสองส่วนนี้เข้าด้วยกัน โดยเทคโนโลยี SDN ประกอบด้วย 3 เลเยอร์หลักๆ ดังนี้

- **ส่วนแอปพลิเคชั่น (Application plane):** เป็นชั้นที่เน็ตเวิร์คเซอร์วิสต่างๆ ได้ถูกติดตั้งอยู่ แต่ละเน็ตเวิร์คเซอร์วิสจะติดต่อสื่อสารกับหน่วยควบคุม (controller) ที่อยู่ในเลเยอร์ด้านล่าง เพื่อดำเนินการตั้งค่า กำหนดนโยบาย (policy) ตรวจสอบสถานะเครือข่าย หรือเรียกดูข้อมูลต่างๆ ผ่าน APIs
- **ส่วนควบคุม (Control plane):** จะทำหน้าที่ในการตัดสินใจว่าแพ็กเก็ตที่วิ่งอยู่ในเครือข่ายหรือเมื่อเข้าถึงระบบแล้ว จะต้องจัดการส่งต่อหรือทำอย่างไรต่อไป ผู้ดูแลระบบสามารถใช้ซอฟต์แวร์เพื่อกำหนดค่าและจัดการการไหลของแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสต่างๆ บนเครือข่ายได้ตามที่ต้องการผ่านจุดควบคุมจากส่วนกลางนี้ ส่งผลให้การจัดการและบริหารเครือข่ายสามารถทำได้อย่างไดนามิก ยืดหยุ่น และมีความคล่องตัวสูง ซึ่งมีความสอดรับกับการตอบสนองความต้องการทางธุรกิจที่ปัจจุบันนี้นับว่ามีความหลายหลาย (diversity in business requirements) และมีการเปลี่ยนแปลงอย่างรวดเร็ว
- **ส่วนรับส่งข้อมูล (Forwarding plane):** จะทำหน้าที่ในการส่งต่อข้อมูลไปตามการตัดสินใจของส่วนควบคุมนั่นเอง ข้อดีที่สำคัญอีกประการหนึ่งของการออกแบบ SDN โดยรวมส่วนควบคุมเอาไว้ที่จุดเดียวจะช่วยให้ผู้ดูแลระบบสามารถจัดการเครือข่าย และควบคุม (program) เส้นทางของแต่ละแพ็กเก็ตได้อย่างมีประสิทธิภาพ

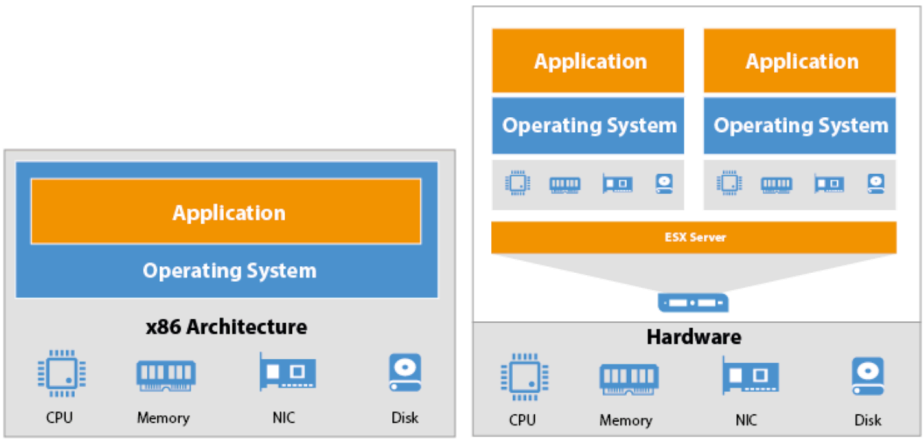



*__รูปที่ 2.3.4:__ การทำงานของเทคโนโลยี SDN สำหรับควบคุมการรับส่งข้อมูลบนเครือข่ายและกำหนดเส้นทางการส่งต่อแพ็กเก็ตไปยังเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ประเภทต่างๆ เช่น Firewall, DPI, Load balancer*

### 2.3.4 เครื่องเสมือนจริง (Virtual Machine: VMs)

Virtual Machine (VM) คือ ซอฟต์แวร์ที่ใช้ทรัพยากรของเครื่องคอมพิวเตอร์ไม่ว่าจะเป็น RAM, hard disk, CPU ในการจำลองเครื่องคอมพิวเตอร์เสมือน ซึ่งจะเรียกเครื่องที่ถูกใช้ทรัพยากรว่า host และเรียกเครื่องเสมือนว่า guest โดยในเครื่อง guest สามารถติดตั้งและใช้งานระบบปฏิบัติการได้เสมือนกับเป็นซอฟต์แวร์หนึ่งของเครื่องคอมพิวเตอร์ ดังนั้นจึงสามารถติดตั้งระบบปฏิบัติการอื่นที่แตกต่างจากระบบปฏิบัติการหลักของเครื่อง host

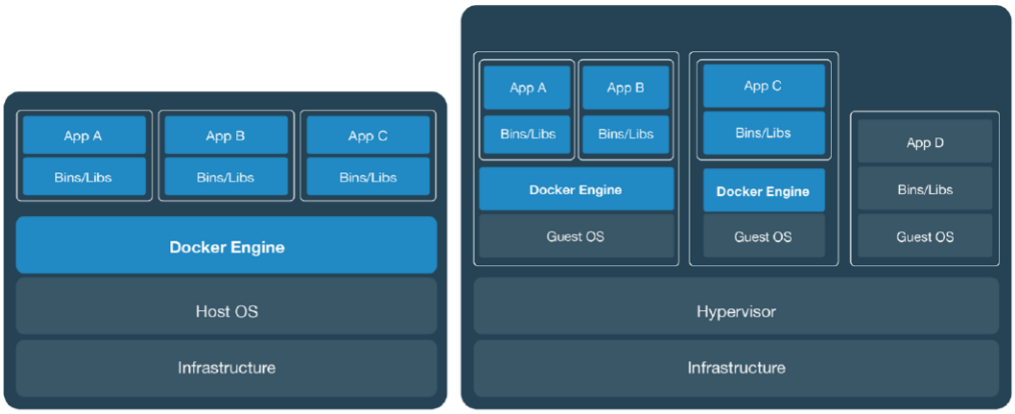
ได้ เช่น เครื่อง host ใช้งาน Microsoft windows แต่เครื่อง Guest ติดตั้ง Linux เป็นต้น นอกจากนี้ยังสามารถสร้าง guest ได้มากกว่า 1 guest ในเครื่อง Host เดียวกัน รูปที่ 2.3.5 แสดงสถาปัตยกรรมของเครื่องเสมือนจริง (VMs) โดย VM มีเกี่ยวข้องกับ cloud computing เนื่องจาก VM นั้นเป็นส่วนหนึ่งของการทำ cloud computing โดย cloud computing จะใช้ server หลาย ๆ ตัว มาวางระบบ virtualization technology ให้ผู้ใช้งานสามารถสร้างคอมพิวเตอร์เสมือน (Virtual Machines) หรือ VM เองได้ ซึ่งผู้ใช้งานสามารถเลือกหรือกำหนดทรัพยากรที่ต้องการจะใช้ได้ รวมไปถึงสามารถเข้าถึงได้ทุกที่ทุกเวลาเพราะเชื่อมต่อถึงกันด้วยระบบอินเทอร์เน็ต




*รูปที่ 2.3.5: สถาปัตยกรรมการออกแบบเครื่องเสมือนจริง (VMs) เมื่อเทียบกับการทำงานของเครื่องเซิร์ฟเวอร์ทั่วไป [26]*

### 2.3.5 Container

เทคโนโลยี container ตัวอย่าง เช่น Docker container คือ เครื่องมือแบบ open-source หรือแพลตฟอร์มซอฟต์แวร์ที่ช่วยให้ผู้พัฒนาระบบสร้าง ทดสอบ และติดตั้งแอปพลิเคชันใช้จริงได้อย่างรวดเร็ว โดย Docker จะบรรจุซอฟต์แวร์ลงไปในหน่วยที่เป็นมาตรฐานเรียกว่า คอนเทนเนอร์ (Container) ซึ่งจะมีทุกสิ่งที่ซอฟต์แวร์ต้องใช้ในการเรียกใช้งาน รวมทั้งไลบรารี เครื่องมือสำหรับระบบ โค้ด และรันไทม์ เมื่อใช้ Docker จะสามารถติดตั้งใช้จริงและปรับขนาดแอปพลิเคชันให้เหมาะกับทุกสภาพแวดล้อม (Deploy) และทราบว่าโค้ดจะ

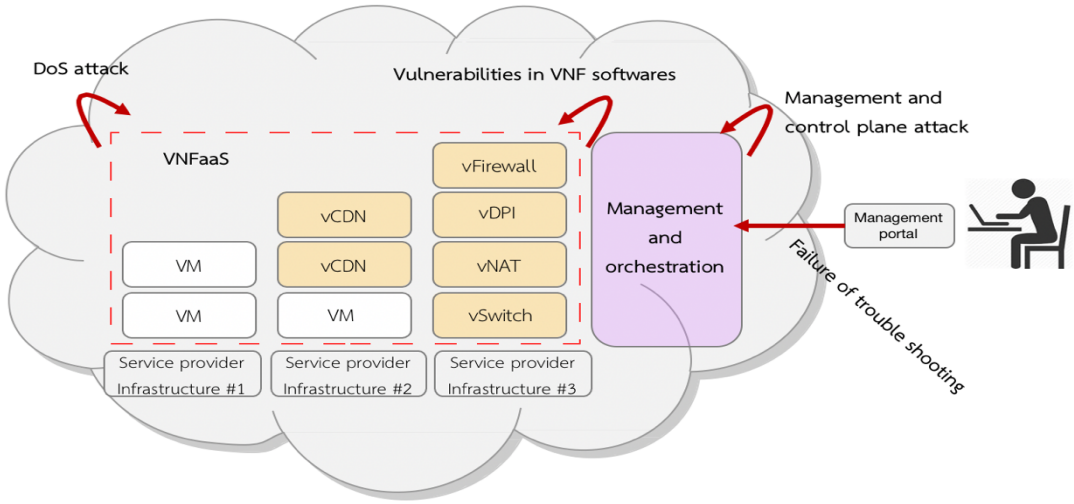



*รูปที่ 2.3.6: สถาปัตยกรรมการออกแบบ Container (ด้านซ้าย) เมื่อเทียบกับเครื่อง VMs (ด้านขวา) [13]*

เรียกใช้ได้อย่างอย่างรวดเร็ว Docker จึงเป็นที่รู้จักในวงกว้างและเริ่มเข้ามามีบทบาทอย่างมากในการพัฒนา Software สามารถรองรับการติดตั้งใช้งานบนระบบปฏิบัติการที่หลากหลายไม่ว่าจะเป็น Linux, Windows, MAC รูปที่ 2.3.6 แสดงสถาปัตยกรรมการออกแบบ Container

## 2.4 ช่องโหว่และภัยคุกคาม

แม้ว่ากลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง ดังเช่น เทคโนโลยี NFV และเทคโนโลยี SDN นำมาซึ่งการเปลี่ยนแปลงโครงสร้างเครือข่ายครั้งสำคัญ อันมีผลอย่างมากต่อการขับเคลื่อนโครงสร้างเครือข่ายแบบดั้งเดิมไปสู่โครงสร้างยุคใหม่ที่กำหนดด้วยซอฟต์แวร์ และมีคุณประโยชน์มากมายดังที่กล่าวมาข้างต้น อย่างไรก็ตาม การเกิดขึ้นของทั้งสองเทคโนโลยีนี้ก็เปรียบเสมือนเทคโนโลยีอื่นๆ ที่เคยเกิดขึ้นก่อนหน้า ซึ่งมีทั้งคุณประโยชน์ และมีทั้งจุดอ่อน ช่องโหว่ หรือภัยคุกคามที่ต้องคำนึงถึง [7, 14] ดังนั้นในหัวข้อนี้จะพูดถึง ความเสี่ยง ภัยคุกคาม และช่องโหว่ที่อาจจะเกิดขึ้นเมื่อมีการประยุกต์ใช้กลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง ดังแสดงในรูปที่ 2.4.1




*รูปที่ 2.4.1: แสดงตัวอย่างรูปแบบการโจมตีและภัยคุกคามที่เกี่ยวข้อง*

- **ปัญหาด้านความปลอดภัยจากการประยุกต์ใช้เครื่องเสมือนจริง (Virtual Machines: VMs)**: เป็นที่ทราบกันดีว่าเทคโนโลยี NFV ได้ถูกออกมาเพื่อแยกเน็ตเวิร์คเซอร์วิส เช่น Firewall, DPI, Switches, NAT, ฯลฯ ในส่วนที่เป็นซอฟต์แวร์ออกมาจากอุปกรณ์ฮาร์ดแวร์เฉพาะของแต่ละค่ายผู้ผลิต เพื่อให้เน็ตเวิร์คเซอร์วิสดังกล่าวฯ สามารถรันได้บนเครื่องเสมือนจริง (VMs) ภายในเครื่องเซิร์ฟเวอร์ทั่วไปบนระบบคลาวด์แพลตฟอร์มได้ อย่างไรก็ตามการออกแบบทางสถาปัตยกรรมของเครื่อง VMs นั้นจะมีความแตกต่างจากการออกแบบคอมพิวเตอร์ทั่วไป โดยมีระบบปฏิบัติการที่สามารถจำลองเครื่องคอมพิวเตอร์ขึ้นมาให้สามารถใช้ซอฟต์แวร์เพื่อจำลองการทำงานของคอมพิวเตอร์เครื่องอื่น (เสมือนจริง) ไว้ในเครื่องเดียวกันได้หรือหลายๆ เครื่อง VMs บนเครื่องคอมพิวเตอร์เครื่องจริงเครื่องเดียวได้ (Host computer) การออกแบบในลักษณะนี้เครื่อง VM จะใช้ทรัพยากรร่วมกันกับเครื่อง VMs อื่นๆ แต่จะถูกจำลองด้วยเทคโนโลยีเสมือน

(Virtualization technology) เปรียบเสมือนว่าแต่ละเครื่องVM มีทรัพยากรแยกออกจากกัน (virtual resource isolation) อย่างไรก็ตามภัยคุกคามและช่องโหว่ที่มาพร้อมกับการใช้งานบน VMs เช่น

- โอกาสที่เครื่อง VMs จะติดไวรัสผ่านไฟล์ VM images[1] ได้ เนื่องจากโดยทั่วไปแล้ว VM images นั้นสามารถหาดาวน์โหลดได้จากอินเทอร์เน็ต ผู้ไม่พึงประสงค์ดีอาจจะอัพโหลด VM images ซึ่งแฝงด้วยมัลแวร์ เมื่อหนึ่งเมื่อใดที่ผู้ใช้งาน (เหยื่อ) นำ VM images ที่แฝงด้วยมัลแวร์ไปใช้ในการติดตั้งเครื่องเสมือนจริง เครื่องที่ทำการติดตั้งนั้นอาจจะติดไวรัสได้
- ดักฟังหรือดักจับข้อมูลขณะดำเนินการย้ายเครื่องเสมือนจริง (VM migration) ด้วยคุณสมบัติของเทคโนโลยีเสมือนจริงที่มีขีดความสามารถให้เครื่อง VMs สามารถโยกย้ายจากเครื่องคอมพิวเตอร์ฮาร์ดแวร์เครื่องหนึ่ง (Host computer) ไปยังเครื่องคอมพิวเตอร์ฮาร์ดแวร์อีกครึ่งหนึ่งได้ ซึ่งเป็นข้อดีอย่างหนึ่งที่ช่วยให้การจัดการและการประมวลผลมีประสิทธิภาพมากยิ่งขึ้น อย่างไรก็ตาม หากปราศจากการป้องกันด้านความปลอดภัยแล้ว (เช่น ปราศจากการเข้ารหัสข้อมูล) ข้อมูลต่างๆ ที่เกี่ยวข้องกับการตั้งค่าเครื่อง VMs นี้ อาจถูกดับจับในระหว่างที่ข้อมูลนั้นส่งผ่านเครือข่ายสาธารณะระหว่างที่ทำการย้ายเครื่องเสมือนจริงได้
- การโจมตีเครื่อง VMs โดยการขัดขวางและพยายามใช้ทรัพยากรบนเครื่องคอมพิวเตอร์ฮาร์ดแวร์ (Host computer) ที่มีอยู่อย่างจำกัดให้หมดไป เพื่อไม่ให้เครื่อง VMs เครื่องอื่นๆ ที่ติดตั้งบนเครื่องคอมพิวเตอร์ฮาร์ดแวร์ (Host computer) เดียวกันสามารถทำงานได้ หรือเรียกวิธีการโจมตีรูปแบบนี้ว่า VM DoS (Danial of Service) เนื่องจากเครื่อง VMs ถูกออกแบบมาให้ใช้ทรัพยากรร่วมกัน (เช่น CPU, memory, bandwidth) ร่วมกับเครื่อง VMs เครื่องอื่นๆ ที่ติดตั้งอยู่บนเครื่องคอมพิวเตอร์ฮาร์ดแวร์ตัวเดียวกัน ซึ่งข้อดีของการออกแบบด้วยวิธีการนี้ คือ ทรัพยากรคอมพิวเตอร์สามารถใช้ประโยชน์ได้อย่างคุ้มค่าและเกิดประโยชน์อย่างสูงสุด อย่างไรก็ตาม การแชร์ทรัพยากรร่วมกันยังพบว่ามีความเสี่ยงและยังมีช่องโหว่อยู่ กล่าวคือ ผู้ไม่พึงประสงค์ดีสามารถโจมตีเครื่อง VMs (เหยื่อ) ได้ โดยทำให้เครื่องเหยื่อนี้เกิดการใช้ทรัพยากรระบบอย่างสิ้นเปลือง เพื่อหวังไม่ให้เครื่อง VMs เครื่องอื่นๆ ที่รันอยู่บนเครื่องคอมพิวเตอร์เดียวกัน (Host computer) ตกอยู่ในสภาวะหยุดชะงัก ไม่สามารถทำงาน หรือตอบสนองคำร้องขอได้ เพราะขาดทรัพยากรเพื่อดำเนินงาน

- **ภัยคุกคามหรือช่องโหว่ที่แอบแฝงตัวมากับซอฟต์แวร์ที่ใช้จำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNF softwares):** เมื่อเทคโนโลยี NFV ได้เข้ามามีบทบาทอย่างมากในการขับเคลื่อนโครงสร้างเครือข่ายจากรูปแบบดั้งเดิมไปสู่ยุคใหม่ เพื่อตอบสนองความรวดเร็วในการประมวลผล ความยืดหยุ่นในการตั้งค่าและเรียกใช้งานเน็ตเวิร์คเซอร์วิสประเภทต่างๆ สอดคล้องตามความต้องการใช้งานเฉพาะทางของแต่ละองค์กร ด้วยเหตุนี้หลากหลายค่ายผู้ผลิต (Vendors) ต่างเริ่มให้ความสนใจและหันมาพัฒนาซอฟต์แวร์ที่ใช้สำหรับจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNFs) ที่สามารถรองรับเทคโนโลยี NFV ได้ ผู้ใช้หรือผู้ดูแลเครือข่าย

[1] VM images คือ แพ็กเก็ตซอฟต์แวร์ที่ถูกแพ็กมาอย่างสมบูรณ์สำหรับพร้อมใช้งานเพื่อใช้ในการตั้งค่าและติดตั้งเครื่องเสมือนจริง ยกตัวอย่างเช่น Ubuntu image ใช้สำหรับการติดตั้งระบบปฏิบัติการอูบุนตู

สามารถเรียกใช้งานและติดตั้งเน็ตเวิร์คเซอร์วิสประเภททต่างๆ ตามที่ต้องการได้อย่างอย่างรวดเร็วและง่ายดาย เช่น ไฟล์วอลล์เสมือนจริง (virtual firewall) อย่างไรก็ตาม ซอฟต์แวร์ที่ถูกพัฒนาขึ้นสำหรับใช้สำหรับจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริงส่วนใหญ่แล้วมีแนวโน้มสูงมากที่จะตรวจพบข้อผิดพลาด (software flaws) เช่น ช่องโหว่ในซอฟต์แวร์ไฟล์วอล์ ซึ่งอาจทำให้ผู้ไม่พึงประสงค์ดีสามารถบายพาสข้อจำกัดของไฟล์วอลล์ด้วยแพ็กเก็ตที่สร้างขึ้นได้ (CVE-2012-2663) หรือการใช้ประโยชน์จากช่องโหว่ในบัฟเฟอร์ (Stack-based buffer overflow) ในการรันคำสั่งโดยพลการเพื่อวัตถุประสงค์บางอย่างที่มิชอบด้วยกฎหมาย (CVE-2006-5276) [15]

- **การโจมตีหน่วยควบคุม:** เมื่อส่วนที่เป็นซอฟต์แวร์ของเน็ตเวิร์คเซอร์วิส (VNFs) ได้ถูกแยกออกจากส่วนที่เป็นอุปกรณ์ฮาร์ดแวร์ ส่งผลให้การเข้าถึง ตั้งค่า บริหารและจัดการต่างๆ สามารถทำได้จากรวดเร็วและง่ายขึ้นจากศูนย์กลาง โดยระบบออกแบบให้มีหน่วยควบคุม (manage and control plane) เพื่อทำหน้าที่เสมือนสมองแกนหลักที่คอยดูแลและควบคุมการทำงานต่างๆ ภายในระบบให้เป็นไปอย่างราบรื่น เช่น การจัดเตรียมทรัพยากรระบบให้พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิส การตั้งค่าเน็ตเวิร์คเซอร์วิสตามนโยบาย (policy) ที่กำหนดไว้ การกำหนดเส้นทางการเชื่อมต่อในเครือข่าย และการตรวจสอบสถานะการทำงาน เหล่านี้เป็นต้น แม้ว่าการทำงานในลักษณะบริหารและจัดการผ่านศูนย์กลาง (control plane) นำมาซึ่งข้อดีหลายประการ อย่างไรก็ตามการออกแบบในลักษณะนี้มีความเสี่ยงอย่างมากที่จะเกิด (1) ระบบล่ม (single point of failures) เนื่องจากปัญหาคอขวดที่เกิดจากการเป็นศูนย์รวมของการทำงานทุกอย่างและการควบคุมเอาไว้ที่จุดเดียว ดังนั้นเมื่อหนึ่งเมื่อใดที่หน่วยควบคุมเกิดความล้มเหลวไม่สามารถทำงานได้ ก็จะส่งผลให้ทุกๆ การทำงานที่เกี่ยวข้องต้องหยุดชะงักไปด้วย และ (2) โอกาสที่หน่วยควบคุมจะถูกโจมตีและเกิดภัยคุกคาม เช่น ผู้ไม่พึงประสงค์ดีอาจจะจู่โจมหน่วยควบคุมซึ่งเปรียบเสมือนสมองแกนหลักของระบบเพื่อเข้าถึงการตั้งค่าและควบคุมการทำงานทั้งหมด เช่น เปลี่ยนการตั้งค่าด้านความปลอดภัย เปลี่ยนเส้นทางการส่งต่อแพ็กเก็ตในเครือข่าย ดักจับหรือโจรกรรมข้อมูลที่สำคัญออกไป
- **การวิเคราะห์ปัญหาล้มเหลว**: โดยทั่วไปแล้วการทำงานของหน่วยควบคุม (management and control plane) มักจะมีความเกี่ยวข้องกับหลายภาคส่วน กล่าวคือ หน่วยควบคุมนี้เปรียบเสมือนสมองแกนหลักของระบบที่คอยจัดการ ประสาน (orchestrate) และติดต่อกับคอมโพเนนท์หรือโมดูลอื่นๆ ที่เกี่ยวข้องภายในระบบเพื่อให้การทำงานต่างๆ เป็นไปอย่างราบรื่น ด้วยความซับซ้อนของระบบที่พยายามแยกการทำงานของแต่ละคอมโพเนนท์ให้มีความเป็นอิสระต่อกันแต่ยังคงประสานรับส่งข้อมูลและทำงานร่วมกันได้ ดังนั้น จึงมีความเป็นไปได้ว่าข้อมูล outputs ที่ได้นั้นอาจผ่านการถูกประมวลผลจากหลายๆ คอมโพเนนท์ ด้วยเหตุนี้ บนความซับซ้อนดังกล่าวฯ ส่งผลให้การวิเคราะห์ปัญหาหรือข้อผิดพลาด (troubleshooting) ทำได้ยากมาก เนื่องจากเราไม่สามารถรู้ได้เลยว่าปัญหาหรือข้อผิดพลาดนั้นแท้จริงแล้วเกิดขึ้น ณ จุดใดหรือคอมโพเนนท์ใดในระบบ ยกตัวอย่างเช่น หากจะวิเคราะห์หาสาเหตุความล่าช้าในการตอบสนองเซอร์วิสจากต้นทางไปยังปลายทาง (end-to-end latency delay) ปัญหาอาจจะเกิดจากประสิทธิภาพและการประมวลผลของตัวเน็ตเวิร์คเซอร์วิส (VNFs) ที่ให้บริการ แพลตฟอร์มที่ให้บริการ หรือปัญหาที่เครื่อง VMs ที่รัน VNFs นั้นๆ เป็นต้น ซึ่งปัจจัยเหล่านี้ต่างเป็นตัวแปรที่ก่อให้เกิดความล่าช้า (latency delay) ดังนั้นจึง

เป็นสิ่งที่ยากและท้าทายมากที่จะวิเคราะห์ถึงต้นตอและสาเหตุของปัญหาที่แท้จริงได้เมื่อเกิดข้อผิดพลาดหรือความล้มเหลวขึ้นในระบบ

**2.5 แนวทางป้องกัน**

หัวข้อนี้จะกล่าวถึงแนวทางการปฏิบัติและวิธีการรักษาความปลอดภัยเพื่อป้องกันและบรรเทาภัยความเสี่ยง ช่องโหว่ หรือภัยคุกคามที่อาจจะเกิดขึ้นเมื่อมีการประยุกต์ใช้กลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง

- **การอัพเดทแพตช์ VM อย่างสม่ำเสมอ:** เนื่องจากว่าไวรัสและมัลแวร์ถูกสร้างขึ้นตลอดเวลา ดังนั้น หนึ่งในแนวทางปฏิบัติที่ดีที่สุดด้านความปลอดภัยเพื่อลดช่องโหว่และความเสี่ยงที่เกิดจากการโจมตีเครื่องเสมือนจริง (VMs) คือ ผู้ดูแลระบบควรตรวจสอบให้แน่ใจว่าระบบปฏิบัติการที่ใช้จำลองเครื่อง VMs ที่กำลังรันอยู่บนเครื่องคอมพิวเตอร์ฮาร์ดแวร์ (Host computer) ตลอดจนซอฟต์แวร์ต่างๆ ที่เกี่ยวข้องนั้นมีการอัพเดทแพตช์ด้านความปลอดภัยอย่างสม่ำเสมอ ซึ่งแพตช์เหล่านี้มีส่วนสำคัญอย่างมากที่จะช่วยอุดช่องโหว่และป้องกันการจู่โจมจากผู้ไม่พึงประสงค์ดี
- **การตรวจสอบความปลอดภัยบน Hypervisor และ VMs:** กลุ่มเทคโนโลยีการจำลองระบบเสมือนจริงถือได้ว่าเป็นเทคโนโลยีที่สามารถทำให้เครื่องเซิร์ฟเวอร์เครื่องหนึ่งมีระบบปฏิบัติการ (Operating system: OS) หรือระบบเสมือนจริง (Virtual system) อยู่ภายในเครื่องเซิร์ฟเวอร์เครื่องเดียวกันได้มากกว่าหนึ่งระบบปฏิบัติการ โดยการใช้วิธีการสร้างชั้น (layer) ของการเชื่อมต่อระหว่างอุปกรณ์ทางกายภาพ (hardware) และระบบปฏิบัติการ (OS) ซึ่งเรียกเทคนิคนี้ว่า Hypervisor เพื่อทำหน้าที่บริหารและจัดการให้ระบบปฏิบัติการของเครื่อง VMs หลายๆ เครื่องให้สามารถรันบนเครื่องเซิร์ฟเวอร์เครื่องเดียวกันและใช้ทรัพยากรต่างๆ ร่วมกันได้อย่างมีประสิทธิภาพ ในอีกมุมหนึ่ง Hypervisor อาจกลายเป็นเป้าหมายสำคัญที่ผู้ไม่พึงประสงค์ดีใช้ในการโจมตีเครื่อง VMs ที่กำลังรันอยู่ได้ ดังนั้นแนวทางการรักษาความปลอดภัยที่ดีที่สุด คือ มั่นคอยตรวจสอบความปลอดภัยบน Hypervisor โดยการติดตั้งระบบตรวจจับภัยคุกคาม (Intrusion Detection System: IDS) หรือที่เรียกว่า hypervisor inspection หรือ Virtual Machine Introspection (VMI) [16] สำหรับทำหน้าที่ตรวจสอบการรับส่งข้อมูลบนเครือข่าย เข้าถึงข้อมูล และสถานะต่างๆ บนเครื่อง VMs เพื่อคอยตรวจจับว่ามีเหตุการณ์ที่ผิดปกติหรือเหตุการณ์ที่น่าสงสัยอันก่อให้เกิดภัยคุกคามบนเครื่อง VMs หรือไม่
- **การตรวจสอบความน่าเชื่อถือของซอฟต์แวร์ที่ใช้จำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNF software integrity):** ผู้ดูแลระบบจะใช้ VNF softwares ในการจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNFs) ประเภทต่างๆ ขึ้นมาบนเครื่อง VMs ที่ถูกสร้างขึ้น ซึ่งการจำลอง VNFs นั้นสามารถจำลองได้บนเครื่อง VM เครื่องใดเครื่องหนึ่ง หรือหลายๆ VMs พร้อมก็ได้กันเพื่อทำ load balancing ซึ่งโดยทั่วไปแล้ว VNF softwares ดังกล่าวฯ สามารถดาวน์โหลดได้โดยตรงจากเว็บไซต์ผู้ให้บริการหรือจากแหล่งพื้นที่จัดเก็บข้อมูลสาธารณะ (public repository) แม้กระนั้น ในมุมด้านความปลอดภัย ผู้ไม่พึงประสงค์ดีอาจตรวจหาช่องโหว่ใน VNF softwares และทำการลักลอบแฝงมัลแวร์เข้าไป ดังนั้น ก่อนที่จะดาวน์โหลด VNF softwares สำหรับใช้งานนั้น ผู้ดูแลระบบควรตรวจสอบความน่าเชื่อถือของ VNF softwares นั้นก่อนเสมอ ซึ่งหนึ่งในเทคนิคที่

ใช้ได้รับความนิยม คือ การตรวจสอบความน่าเชื่อถือว่า VNF softwares ดังกล่าวฯ มีใบรับรองอิเล็กทรอนิกส์ (digital certificate) ที่ออกโดยองค์กรออกใบรับรอง (Certificate Authority: CA) ที่น่าเชื่อถือได้ แนบมาด้วยหรือไม่

- **การพิสูจน์ตัวตนและการตรวจสอบสิทธิ์:** เพื่อป้องกันความเสี่ยงและภัยคุกคามที่อาจจะเกิดขึ้นจากการเข้าถึงข้อมูลและหน่วยควบคุมในระบบ ดังนั้น วิธีการพิสูจน์ตัวตน (Authentication) และการตรวจสอบสิทธิ์การเข้าถึงข้อมูล (Authorization) ถือได้ว่าเป็นวิธีการที่ได้รับความนิยมและได้รับการยอมรับมากที่สุด
- **การเข้ารหัสข้อมูล:** การใช้เทคโนโลยีการเข้ารหัสลับ หรือการกำหนดนโยบายการเข้าถึง Access Control Policy ทั้งในระดับข้อมูลและในระดับเครือข่ายสามารภป้องกันการรั่วไหลข้อมูลได้
- **การสำรองข้อมูล (Replication):** การสร้างระบบสำรองทั้งในส่วนฐานข้อมูล หน่วยการทำงานหลัก และการทำแผนการดำเนินการเพื่อเร่งการกู้คืนระบบ
- **การตรวจสอบความผิดปกติและด้านความปลอดภัย (Analytic and security monitoring):** เนื่องจากกลุ่มเทคโนโลยีการจำลองระบบเสมือนจริงถูกออกแบบสถาปัตยกรรมให้เป็นแบบระดับชั้น (Layering model) ซึ่งแต่ละคอมโพเนนท์ที่อยู่ในแต่ละระดับชั้นจะทำงานแยกส่วนกัน ทั้งนี้จะมีหน่วยควบคุม (management and control plane) เพื่อทำหน้าที่คอยจัดการและประสาน (orchestrate) ให้การทำงานต่างๆ ภายในระบบเป็นไปอย่างราบรื่น อย่างไรก็ตามการออกแบบสถาปัตยกรรมแบบระดับชั้นนี้นำมาซึ่งความท้าทายในการวิเคราะห์หาสาเหตุทีแท้จริงที่เกิดปัญหาล้มเหลวและความท้าทายในการตรวจสอบด้านความปลอดภัย อย่างไรก็ตาม หนึ่งในวิธีการที่ถูกแนะนำโดยสถาบันมาตรฐานโทรคมนาคม (European Telecommunications Standards Institute: ETSI) คือ การจัดให้มีระบบ Analytics และ Security monitoring [17] เพื่อคอยตรวจสอบสถานะการทำงานของแต่ละคอมโพเนนท์ในระดับชั้นต่างๆ และการรับส่งข้อมูลในเครือข่ายว่ามีความผิดปกติหรือไม่

## 2.6 บทสรุป

การเกิดขึ้นของกลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization technology) ได้แก่ เทคโนโลยีการจัดเก็บและการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) เทคโนโลยีการทำงานของเครือข่ายเสมือนจริง (NFV) และเทคโนโลยีการจัดการเครือข่ายด้วยซอฟต์แวร์ (SDN) นำมาซึ่งการเปลี่ยนแปลงมากมายต่อการจัดการโครงสร้างพื้นฐาน ICT และการจัดการโครงสร้างเครือข่าย อีกทั้งยังได้เข้ามามีบทบาทสำคัญในการขับเคลื่อนโครงสร้างพื้นฐานระบบเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบดั้งเดิม ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดโดยซอฟต์แวร์ โดยจุดเด่นหลักๆ คือ รายจ่ายด้านการลงทุนและค่าดำเนินการลดลง ความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส ความยืดหยุ่นในการจัดสรรหรือขยายทรัพยากรเครือข่าย ประสิทธิภาพในการประมวลผลและการให้บริการ และการให้อิสระต่อการเลือกใช้งานเน็ตเวิร์คเซอร์วิสประเภทต่างๆ ตามบริบทขององค์กรหรือตามความต้องการใช้งานในแต่ละภาคธุรกิจได้

ในอีกมุมหนึ่งการประยุกต์ใช้งานกลุ่มเทคโนโลยีการจำลองระบบเสมือนจริงนี้ก็อาจนำมาซึ่งความเสี่ยงและภัยคุกคาม ซึ่งตัวอย่างช่องโหว่และภัยคุกคามที่ได้กล่าวมาข้างต้น ทำให้เราได้ทราบว่าช่องทางหรือพื้นที่เสี่ยงต่อการ

โจมตีโดยผู้ไม่พึงประสงค์ดีนั้นมีพื้นที่ค่อนข้างกว้างและคลอบคลุมหลายระดับชั้นของสถาปัตยกรรม ซึ่งหมายรวมถึง (1) ชั้นโครงสร้างพื้นฐาน (infrastructure layer) ที่มีเครื่องเซิร์ฟเวอร์ อุปกรณ์ด้านเครือข่าย และเครื่องเสมือนจริง (VMs) ติดตั้งอยู่ ดังนั้นตัวอย่างภัยคุกคามที่เกี่ยวข้องหมายรวมถึงภัยคุกคามที่มาจากเครื่อง VMs (ดังเช่นตัวอย่างที่ได้กล่าวไปข้างต้น) (2) ชั้นเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNF layer) ที่เป็นแหล่งรวมเน็ตเวิร์คเซอร์วิสประเภทต่างไว้ด้วยกัน ดังเช่น firewall, DPI, NAT, CDN, switches เป็นต้น ดังนั้นตัวอย่างช่องโหว่และภัยคุกคามที่เกี่ยวข้องหมาย เช่น ภัยคุกคามที่แฝงตัวมาพร้อมกับซอฟต์แวร์ที่ใช้จำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (3) ชั้นจัดการและประสาน (management and orchestration layer) ที่ทำหน้าที่เสมือนสมองแกนหลักของระบบในการดูแลประสานงาน และควบคุมการทำงานให้เป็นไปตามนโยบายหรือข้อกำหนดที่ได้วางไว้ โดยช่องโหว่งและภัยคุกคามที่เกี่ยวข้องกับชั้นนี้ คือ การโจมตีหน่วยควบคุม และการวิเคราะห์ปัญหาล้มเหลว ด้วยเหตุนี้ การรักษาความปลอดภัยในทุกระดับชั้นของสถาปัตยกรรมเทคโนโลยีการจำลองระบบเสมือนจริง จึงเป็นสิ่งที่สำคัญอย่างยิ่งที่ผู้ดูแลและจัดการเครือข่ายต้องให้ความสนใจและควรตระหนักถึง ดังนั้นการเตรียมความพร้อม มีมาตราการรักษาความปลอดภัยและป้องกันภัยคุกคามอย่างเข้มแข็งและตรงจุดจะเป็นการช่วยลดโอกาสได้ดีเพื่อป้องกันระบบจากการถูกโจมตี

ในอีกมุมหนึ่งโครงการนี้มีเป้าหมายหลักเพื่อส่งเสริมให้เกิดการนำกลุ่มเทคโนโลยีการจำลองระบบเสมือนจริงไปใช้ในการพัฒนานวัตกรรมใหม่ ๆ ให้กับผู้ให้บริการโครงข่าย สนับสนุนการวางรากฐานโครงสร้างพื้นฐาน ICT และการสร้างระบบนิเวศดิจิทัลที่เข้มแข็ง รวมทั้งการสร้างความเชื่อมั่นและคุ้มครองความปลอดภัยให้กับประชาชนในการใช้บริการและส่งเสริมให้เกิดการให้บริการเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNFs as a service) ประเภทต่างๆ เช่น Firewall, IDS/IPS, antivirus, ฯลฯ ซึ่งการสร้างแพลตฟอร์มสำหรับบริหารและจัดการสภาพแวดล้อมเสมือนจริง (Management and Orchestration (MANO) platform) จะเป็นการช่วยให้ผู้ให้บริการและผู้ดูแลเครือข่ายมีความคล่องตัวในการจัดสรรทรัพยากร ควบคุมลำดับการทำงานของฟังก์ชั่นต่างๆ (VNFs) บริหารและจัดการวัฏจักรของแต่ละฟังก์ชั่น และจัดเตรียมสภาพแวดล้อมเสมือนจริงให้เหมาะสมกับการทำงานต่าง ๆ อย่างมีประสิทธิภาพตลอดจนประสานให้การทำงานเป็นไปอย่างอัตโนมัติสอดคล้องกับบริบทและความต้องการที่หลากหลายของผู้ใช้งาน ส่งเสริมขีดความสามารถของเทคโนโลยีการจำลองระบบเสมือนจริง สนับสนุนการขับเคลื่อนเศรษฐกิจและสังคมดิจิทัลสู่ยุค 5G ในอนาคตและยังมีส่วนช่วยให้เกิดการลดสัดส่วนการพึ่งพาเทคโนโลยีจากต่างประเทศ โดยการใช้องค์ความรู้ เครื่องมือ และบุคลากรที่มีในการพัฒนานวัตกรรมขึ้นเอง ช่วยสร้างโอกาสในการเพิ่มขีดความสามารถด้านการแข่งขันและการพึ่งพาตัวเองภายในประเทศไทย

# บทที่ 3
# กิจกรรมที่ 2: ดำเนินการวิจัย ออกแบบ และพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง

## 3.1 บทนำ

ปัจจุบันนี้ พบว่ามีหลายๆ ประเทศทั้งจากทวีปยุโรปและทวีปเอเชียต่างเล็งเห็นถึงความสำคัญในการพัฒนาและขับเคลื่อนงานวิจัยที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริงไปสู่การใช้งานจริงมากขึ้น (ดังเช่นโครงการที่ปรากฏด้านล่างนี้) เนื่องจากประเทศส่วนใหญ่ต่างให้การยอมรับว่าเทคโนโลยีการจำลองระบบเสมือนจริงจะนำมาซึ่งการเปลี่ยนแปลงโครงสร้างเครือข่ายครั้งสำคัญ ซึ่งเปรียบเสมือนเสาหลักที่สำคัญต่อการวางรากฐานโครงสร้าง ICT และการเตรียมระบบนิเวศ 5G ในอนาคต ที่สามารถลดข้อจำกัดต่างๆ ที่ประสบในเครือข่ายแบบดั้งเดิม (traditional networking model) ได้ ไม่ว่าจะเป็น การช่วยลดค่าใช้จ่ายด้านการรลงทุนสำหรับการจัดซื้อและติดตั้งอุปกรณ์เครือข่าย (Capital Expenditure: CapEx) และค่าใช้จ่ายที่เกี่ยวข้องกับการบำรุงรักษาซ่อมแซมและค่าดำเนินการ (Operational Expenditure: OpEx) ที่เกิดขึ้นภายในองค์กร ในอีกมุมหนึ่งการประยุกต์ใช้งานเทคโนโลยีการจำลองระบบเสมือนจริงจะช่วยเพิ่มศักยภาพในการตอบสนองการให้บริการและความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส ตลอดจนความสามารถในการจัดสรรทรัพยากรได้อย่างยืดหยุ่นเพื่อให้เกิดการใช้ทรัพยากรเครือข่ายได้อย่างมีประสิทธิภาพ โดยการปรับลดหรือเพิ่มจำนวนขนาดของทรัพยากรเครือข่ายให้มีความเหมาะสมและสอดคล้องกับจำนวนเวิร์คโหลดที่เข้ามาในช่วงเวลานั้นๆ ได้ ในอีกแง่มุมหนึ่ง หากพิจารณาในบริบทประเทศไทยเส้นทางการพัฒนานวัตกรรมที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริงสู่การประยุกต์ใช้งานจริงพบว่ายังอยู่ในช่วงเริ่มต้นของการศึกษาและยังไม่ได้รับความนิยมใช้กันอย่างแพร่หลายมากนัก โดยสาเหตุหลักสำคัญพบว่าองค์ความรู้ที่เกี่ยวข้องกับเทคโนโลยีนี้ในประเทศไทยแทบจะมีน้อย บวกกับความไม่พร้อมของกำลังคนและความท้าทายในการนำเทคโนโลยีการจำลองระบบเสมือนจริงมาปรับใช้ภายในองค์กร ด้วยเหตุนี้ ในหัวข้อนี้ จะพูดถึงโครงการที่เกี่ยวข้องกับการพัฒนาแพลตฟอร์ม NFV Management and Orchestration (MANO) ในปัจจุบัน โดยมีวัตถุประสงค์หลัก เพื่อศึกษา วิเคราะห์ และทำความเข้าใจว่าแต่ละโครงการมีแนวทางในการออกแบบสถาปัตยกรรมระบบสำหรับการบริหารและจัดการทรัพยากรเครือข่าย และเน็ตเวิร์คเซอร์วิส (management and orchestration platform) อย่างไร

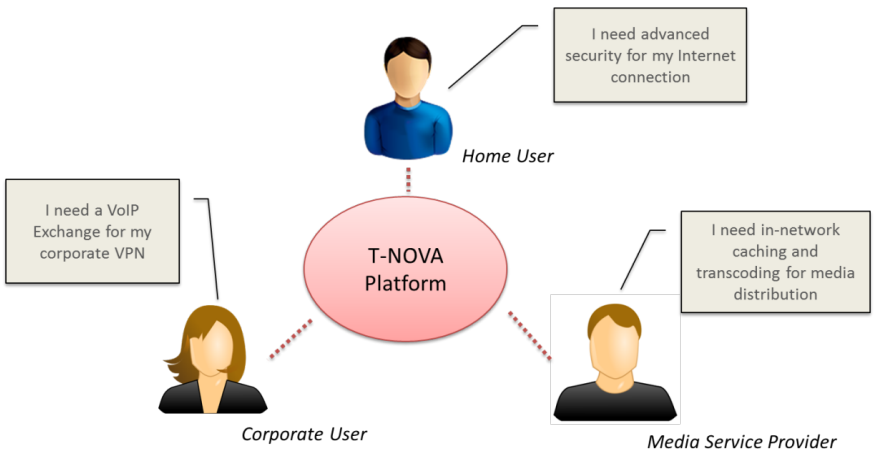
## 3.2 โครงการที่เกี่ยวข้องกับการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเสมือนจริงในปัจจุบัน

### 3.2.1 โครงการ T-NOVA

โครงการ T-NOVA [18-21] เป็นโครงการที่ได้รับทุนวิจัยสนับสนุนจากคณะกรรมการยุโรปในปี ค.ศ. 2014 – 2016 ซึ่งถูกก่อตั้งขึ้นมาจากหลายภาคส่วน ทั้งจากภาคภาคอุตสาหกรรม สถาบันการศึกษา และวิสาหกิจขนาดกลางและขนาดย่อม (SMEs) เพื่อดำเนินการวิจัย พัฒนา และทดสอบเทคโนโลยี Network Functions Virtualization (NFV) และ Software Defined Networking (SDN) ซึ่งเป็นเทคโนโลยีที่มีบทบาทสำคัญต่อการยกระดับการบริหารจัดการเน็ตเวิร์คเซอร์วิสและควบคุมเครือข่ายแบบสมัยใหม่ที่กำหนดโดยซอฟต์แวร์ โดย

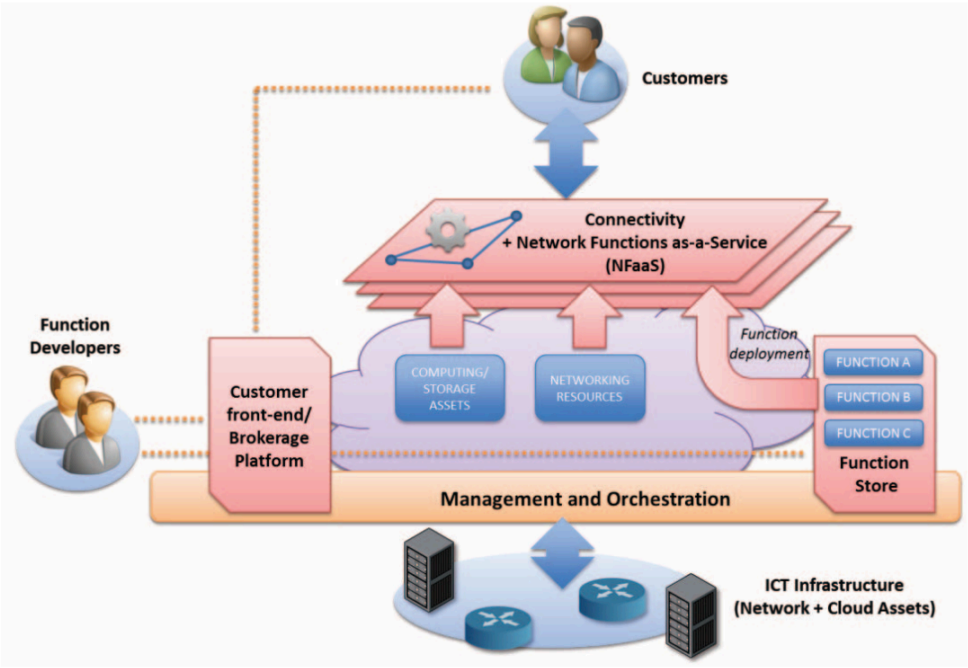
จุดประสงค์หลักของโครงการนี้ คือ การนำเสนอแนวความคิดในการออกแบบและสร้างสถาปัตยกรรมการบริหารและจัดการเน็ตเวิร์คเซอร์วิสแบบบูรณาการ หรือที่เรียกว่า NFV Management and Orchestration (NFV MANO) สำหรับทำหน้าที่ในการจัดเตรียม บริหารจัดการ ตรวจสอบ และเพิ่มประสิทธิภาพในการจำลองเน็ตเวิร์คเซอร์วิสแบบเสมือนให้เป็นไปโดยอัตโนมัติ นำมาซึ่งการลดความซับซ้อน สร้างความยืดหยุ่นในการจัดสรรทรัพยากรเครือข่าย และช่วยลดต้นทุนในการบริหารจัดการเครือข่ายและการจำลองเน็ตเวิร์คเซอร์วิส นอกเหนือจากนี้ โครงการ T-NOVA ยังให้ความสำคัญกับเทคโนโลยี Openflow เพื่อจัดการทรัพยากรเครือข่ายอย่างมีประสิทธิภาพ เช่น การแบ่งส่วนเครือข่าย การเปลี่ยนเส้นทางการรับส่งข้อมูล และการจัดหาคุณภาพบริการ (Quality of Service: QoS) อีกทั้ง โครงการ T-NOVA ยังแนะนำแนวคิดใหม่ที่เรียกว่า Network Functions as-a-Service (NFaaS) เพื่อช่วยให้ผู้ให้บริการเครือข่ายสามารถบริหารและจัดการเน็ตเวิร์คเซอร์วิสของตนเองได้อย่างมีประสิทธิภาพตามความต้องการใช้งาน ในขณะเดียวกัน โครงการ T-NOVA เปิดโอกาสให้บุคคลที่ 3 หรือผู้ให้บริการรายอื่นได้สามารถปล๊ักอินเน็ตเวิร์คเซอร์วิสของตนเอง เช่น gateway, proxy, firewall, IDS/IPS, media transcoders, traffic analyzer ฯล, เพื่อให้บริการในรูปแบบ Network Functions as-a-Service ทำให้ผู้ขอรับบริการไม่จำเป็นต้องจัดซื้อ ติดตั้งและรับผิดชอบค่าบำรุงรักษาฮาร์ดแวร์ในสถานที่ของตนเอง รูปที่ 3.2.1 แสดงแนวคิดของการให้บริการเน็ตเวิร์คเซอร์วิสในรูปแบบ Network Functions as-a-Service (NFaaS) โครงการ T-NOVA มีการออกแบบและพัฒนา 5 โมดูลหลัก (แสดงดังรูปที่ 3.2.2)

- Network Functions as-a-Service (NFaaS): ซึ่งทำหน้าที่ในจัดสรรทรัพยากรคอมพิวเตอร์ ได้แก่ CPU, RAM, storage ให้พร้อมสำหรับการสร้าง VNFs ตามคำร้องขอของผู้ใช้ รวมถึงการจัดการโฟลว์ กลไกการควบคุม และการประมวลผลเพย์โหลดแพ็กเก็ตในเครือข่ายตามความต้องการเฉพาะของแต่ละกลุ่มผู้ใช้งาน
- Network Function Store: เป็นแนวคิดที่ได้รับแรงบันดาลใจจากร้านค้าแอปพลิเคชั่นบนระบบปฏิบัติการที่ได้รับความนิยมสูง ซึ่งเปรียบเสมือนคลังสินค้าที่ใช้จัดเก็บ VNF package ประเภทต่างๆ ที่จำเป็นต้องใช้สำหรับการสร้าง VNFs โดยแต่ละ VNF package จะประกอบด้วย VNF software image, metadata file, และ script ซึ่งบรรจุข้อมูลและชุดคำสั่งที่สำคัญสำหรับการสร้าง VNFs สามารถเรียกดูและเลือกใช้ VNFs ประเภทต่างๆ ได้ตามความต้องการ สามารถกำหนด หรือปรับแต่งการตั้งค่าของแต่ละ VNF เพื่อให้มีเหมาะสมกับบริบทของแต่ละกลุ่มผู้ใช้




*รูปที่ 3.2.1: แนวคิดการออกแบบโครงการ T-NOVA ในรูปแบบ Network Functions as-a-Service (NFaaS)*

- Connectivity service: เพื่อคอยจัดสรรทรัพยากรเครือข่าย ได้แก่ แบนด์วิธ, QoS และจัดการการเชื่อมต่อ VNFs เข้ากับระบบเครือข่ายจริง โดยใช้เทคโนโลยี SDN เป็นพื้นฐานในการพัฒนาสำหรับทำหน้าที่จัดการเชื่อมต่อ และสร้างเครือข่ายเสมือน (vNets) ตามนโยบายที่ได้กำหนดไว้
- Management and orchestration: โมดูลนี้มีหน้าที่รับผิดชอบหลักในควบคุมลำดับการทำงาน ประสานกับโมดูลอื่นๆ ภายในระบบ บริหารและจัดการให้การทำงานเป็นไปอย่างอัตโนมัติ รวมถึง การบริหารและจัดการวงจรชีวิตของแต่ละ VNF ที่ได้ถูกสร้างขึ้น การตรวจสอบสถานะการใช้งานทรัพยากรเครือข่าย และปรับเพิ่ม/ลดทรัพยากรเครือข่ายเพื่อให้มีความเหมาะสมกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ
- Customer front-end: ซึ่งเป็นส่วนที่ผู้ใช้ใช้ในการติดต่อกับระบบ




*รูปที่ 3.2.2: สถาปัตยกรรม T-NOVA*

โดยคุณสมบัติหลักของโครงการ T-NOVA มุ่งเน้น 6 ด้านหลักๆ ดังนี้

- มีขีดความสามารถในการจำลองเน็ตเวิร์คเซอร์วิส จัดการทรัพยากรเครือข่ายและปรับขนาดของทรัพยากรเครือข่ายที่ใช้งานได้อย่างเหมาะสม
- มีขีดความสามารถในการตรวจสอบและติดตามการใช้งานทรัพยากรเครือข่ายเพื่อให้เหมาะสมกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ
- ระบบมีการออกแบบโครงสร้างและอินเทอร์เฟสที่เปิดเป็นสาธารณะ มีการใช้โพรโตคอลการสื่อสารที่ได้รับความนิยมม เช่น HTTP-based REST APIs และรูปแบบข้อมูลที่ชใช้กันแพร่หลาย เช่น XML, JSON ทำให้ระบบสามารถทำงานร่วมกันได้และสามารถให้บริการกับหลายระบบ/โปรแกรมได้อย่างยืดหยุ่น

- แนวคิดโครงการมุ่งเน้นการออกแบบและพัฒนาระบบออกเป็นโมดูลย่อยๆ โดยแต่ละโมดูลจะมีความอิสระและยืดหยุ่นต่อกัน ช่วยให้แต่ละโมดูลสามารถกำหนดค่าได้หลายแบบ ทำให้ระบบมีความยืดหยุ่นเพื่อตอบสนองความต้องการของผู้ใช้ได้หลากหลาย
- ระบบใช้ซอฟต์แวร์โอเพนซอร์สที่สามารถติดตั้งได้บนเซิร์ฟเวอร์ ทำให้เกิดความเข้ากันได้สูงกับโครงสร้างพื้นฐานที่มีอยู่ เช่น การรองรับทั้ง SDN และ non-SDN เพื่อทำให้ระบบสามารถทำงานได้โดยมีความคุ้มค่า
- การนำเทคนิค VM migration และประยุกต์ใช้การจัดการ SDN สำหรับการเปลี่ยนเส้นทางการส่งข้อมูลมาช่วยแก้ปัญหาหากพบว่าโหนดใดโหนดหนึ่งเกิดความล้มเหลว ทำให้ระบบสามารถตรวจสอบข้อผิดพลาดและสามารถแก้ปัญหาได้อย่างรวดเร็วเมื่อระบบขาดการทำงาน

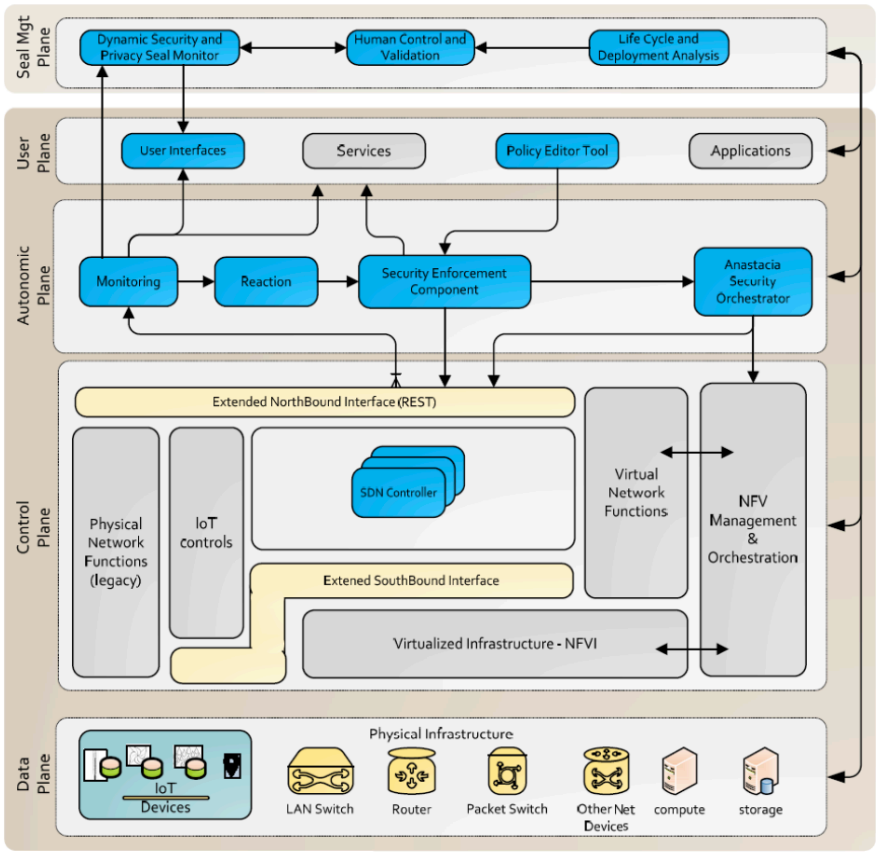
### 3.2.2 โครงการ ANASTACIA

โครงการ ANASTACIA [22-24] เป็นโครงการวิจัยที่ได้รับทุนสนับสนุนจากสหภาพยุโรปภายใต้โครงการความร่วมมือด้านการวิจัยและนวัตกรรมที่ชื่อว่า Horizon 2020 ซึ่งเป็นโครงการที่ได้ถูกก่อตั้งขึ้นมากจากหลายภาคส่วนทั้งที่มาจากภาคเอกชนและภาคการศึกษา โดยมีเป้าหมายหลักสำคัญ คือ การออกแบบและพัฒนาแพลตฟอร์มเพื่อบริหารจัดการเน็ตเวิร์คเซอร์วิส และควบคุมความมั่นคงปลอดภัยให้เป็นไปอย่างอัตโนมัติ โดยสามารถจัดการและแก้ไขปัญหาด้านความปลอดภัยได้ด้วยตนเองปราศจากการได้รับความช่วยเหลือหรือการแทรกแซงจากมนุษย์ (zero human intervention) เพื่อสร้างแพลตฟอร์มที่มีความยืดหยุ่นและสามารถทำงานร่วมกันได้สนับสนุนสภาพแวดล้อม IoTs และ Cyber Physical Systems (CPS) ในอีกมุมหนึ่งโครงการมีจุดมุ่งหมายเพื่อรวมการรักษาความปลอดภัยเข้ากับระบบ ICT ตั้งแต่ระยะเริ่มต้น ปรับให้เข้ากับเงื่อนไขการรักษาความปลอดภัยที่พัฒนา ลดความจำเป็นในการแก้ไขหลังการปรับใช้ และรับประกันความปลอดภัยและความน่าเชื่อถือของระบบอย่างต่อเนื่อง โดยการพัฒนากรอบการรักษาความปลอดภัยที่สร้างขึ้นบนหลักการของการออกแบบที่น่าเชื่อถือ เฟรมเวิร์กนี้จะครอบคลุมทุกขั้นตอนของวงจรการพัฒนาระบบ ICT (SDL) และจะรวมความสามารถในการตัดสินใจด้วยตนเองโดยอาศัยเทคโนโลยี NFV และ SDN เพื่อช่วยขับเคลื่อนการดำเนินงานให้เป็นไปอย่างไดนามิกและอัตโนมัติสำหรับการรักษาความปลอดภัย

โดยแนวคิดของการออกแบบและพัฒนาระบบ มุ่งเน้นการพัฒนาแพลตฟอร์มให้มีขีดความสามารถในการป้องกันตนเอง (self-protection) รักษาตนเอง (self-healing) และซ่อมแซมตนเอง (self-repair) ซึ่งมีการทำงานที่ครอบคลุมตั้งแต่การออกแบบไปจนถึงการปรับใช้และการบำรุงรักษา เพื่อเพิ่มประสิทธิภาพของมาตรการรักษาความปลอดภัย สำหรับการออกแบบเฟรมเวิร์กนี้จะเป็นไปตามมาตรฐานที่กำหนดโดย ETSI NFV (European Telecommunications Standards Institute Network Function Virtualization) และ OFN SDN (OpenFlow Network) เพื่อให้มั่นใจว่าเป็นไปตามข้อกำหนด SDN/NFV อย่างสมบูรณ์ รูปที่ 3.2.3 แสดงสถาปัตยกรรมการออกแบบโครงการ ANASTACIA ซึ่งประกอบด้วย 4 ชั้นหลัก ดังนี้

- Seal management plane: เป็นส่วนที่รวบรวมมาตรฐานและระเบียบข้อบังคับต่างๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัยเอาไว้ เพื่อให้ผู้ใช้ได้ใช้อ้างอิงแนวทางการตรวจสอบและป้องกันว่าเป็นไปตามมาตราฐานหรือไม่ และมั่นใจถึงการป้องกันที่ครอบคลุม

- User plane: จะมีส่วนประกอบต่างๆ เช่น อินเทอร์เฟซ แอปพลิเคชัน และเครื่องมือต่างๆ ที่ช่วยให้ผู้ดูแลระบบสามารถจัดการ IoT platform ผ่าน ANASTACIA framework เพื่อให้ผู้ใช้สามารถเข้าถึง กำหนด แก้ไข และดำเนินการตั้งค่านโยบายการรักษาความปลอดภัย (security policy) ต่างๆ ได้
- Autonomic plane: มีหน้าที่บังคับใช้กลไกการรักษาความปลอดภัย ตลอดจนกำหนดการตั้งค่าด้านความปลอดภัยต่างๆ ให้กับเน็ตเวิร์คเซอร์วิสที่ถูกสร้างขึ้น ซึ่งแบ่งออกเป็น 3 โมดูลย่อยดังนี้




*รูปที่ 3.2.3: สถาปัตยกรรม ANASTACIA*

  - Security Orchestrator: เป็นโมดูลหลักสำคัญของระบบ ทำหน้าที่ในการรับผิดชอบการบังคับใช้กฎระเบียบการรักษาความปลอดภัยที่กำหนดไว้ ดังเช่น (1) ทำการแปลง security properties เป็น configuration rules เพื่อกำหนดค่าการรักษาความปลอดภัยให้กับเน็ตเวิร์คเซอร์วิสต่างๆ ที่ถูกจำลองขึ้น (2) ดำเนินการตั้งค่า security policies ให้มีความสอดคล้องกับเน็ตเวิร์คเซอร์วิสที่เกี่ยวข้อง เพื่อให้มั่นใจว่านโยบายรักษาความปลอดภัยทั้งหมดได้รับการบังคับใช้อย่างถูกต้อง และ (3) ดำเนินการจัดสรรทรัพยากรการประมวลผลและหน่วยความจำให้กับเน็ตเวิร์คเซอร์วิสด้านความมปลอดภัยต่างๆ รวมถึงจัดการการอัปเดตซอฟต์แวร์และฐานข้อมูลลายเซ็นไวรัส เป็นต้น และทำการตรวจสอบและวิเคราะห์เหตุการณ์รักษาความปลอดภัยต่างๆ ที่เกิดขึ้นในเครือข่าย เพื่อให้สามารถระบุและตอบสนองต่อภัยคุกคามได้อย่างทันท่วงที
  - Security Enforcement: เป็นโมดูลที่ทำหน้าบังคับใช้นโยบายความปลอดภัยตามที่กำหนดไว้

- Monitoring and Reaction: เป็นโมดูลที่คอยรวบรวมข้อมูลความปลอดภันที่เกี่ยวข้องกับพฤติกรรมของระบบ และตอบสนองต่อความผิดปกติหรือภัยคุกคามที่เกิดขึ้น โดยจะเชื่อมต่อกับ IoT Platform ผ่าน Security Enforcement Plane เพื่อรวบรวมข้อมูลความปลอดภัยจากอุปกรณ์ IoT, อุปกรณ์แบบฝังตัว และระบบเดิม โดยทำการการตรวจจับความผิดปกติ และสร้างการตอบสนองเพื่อลดผลกระทบจากการโจมตี เช่นการกำหนดค่าใหม่และแจ้งเตือนให้ผู้ดูแลระบบ

- Control plane: รับผิดชอบในการบริหารและจัดการเน็ตเวิร์คเซอร์วิส และทรัพยากรเครือข่ายให้เป็นไปอย่างมีประสิทธิภาพสอดคล้องกับนโยบายการรักษาความปลอดภัยที่กำหนดขึ้น
- Data plane: เป็นส่วนที่อุปกรณ์เครือข่ายและอุปกรณ์ IoTs ต่างๆ ติดตั้งอยู่เพื่อทำหน้าที่รับส่งข้อมูลและติดต่อสื่อสาร

ในอีกมุมหนึ่ง แนวทางการออกแบบและพัฒนาแพลตฟอร์ม ANASTACIA มุ่งเน้นที่จะพัฒนาระบบเพื่อตอบโจทย์คุณสมบัติ 4 ด้าน [25-26] ดังนี้

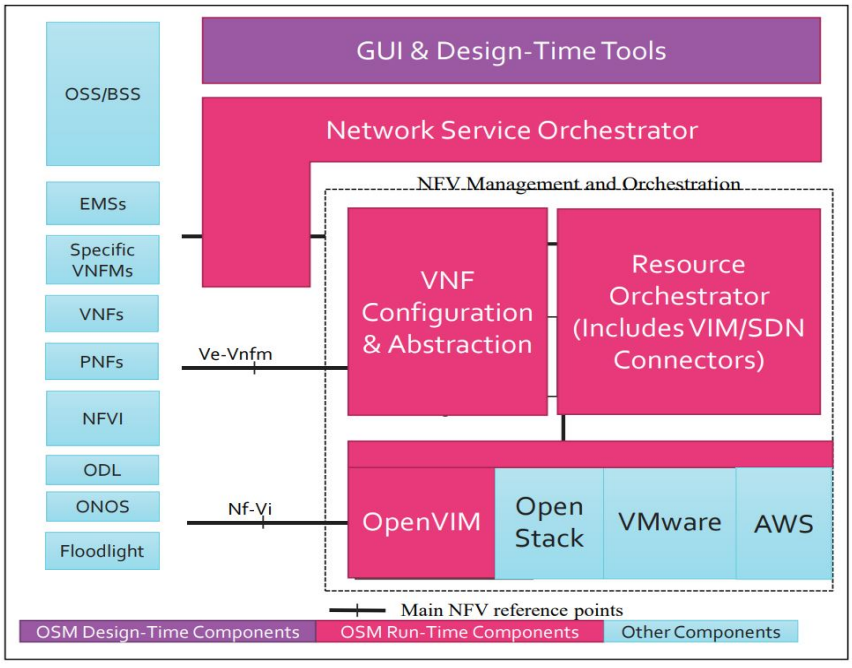
- Interoperability: มีขีดความสามารถในการทำงานร่วมกันได้กับเอเจนต์ภายนอก โดยผู้ใช้สามารถเลือกใช้เอเจนต์รักษาความปลอดภัยอื่นๆ นอกเหนือจากเอเจนต์รักษาความปลอดภัยที่ทางแพลตฟอร์มได้จัดเตรียมไว้ให้ ซึ่งจะช่วยให้ผู้ใช้สามารถสร้างระบบรักษาความปลอดภัยที่ปรับแต่งให้เหมาะสมกับความต้องการของตนเอง โดยสามารถทำได้ 2 วิธีการ คือ การแชร์ข้อมูลร่วมกันระหว่างเอเจนต์ภายนอกผ่านทาง Kafka broker หรือ Adapter ที่ทางแพลตฟอร์มได้จัดเตรียมไว้ให้
- Scalability: ขีดความสามารถในการขยายทรัพยากรเพื่อรองรับการใช้งานในระบบที่มีขนาดใหญ่และซับซ้อน โดยระบบสามารถใช้เอเจนต์หลายตัวทำงานร่วมกันเพื่อประมวลผลข้อมูลและตรวจจับภัยคุกคามได้ ซึ่งจะช่วยให้สามารถทำงานได้อย่างมีประสิทธิภาพแม้ว่าจะมีปริมาณข้อมูลที่มากขึ้นหรือระบบได้รับการโจมตี
- Machine Learning and AI Integration: การนำเข้าเทคโนโลยี Machine Learning และ AI เพื่อปรับปรุงประสิทธิภาพในการตรวจจับและป้องกันการละเมิดความมั่นคงปลอดภัย
- Real-time Monitoring and Response: การตรวจสอบและตอบสนองเป็นเวลาจริงต่อภัยคุกคามหรือการละเมิดความมั่นคงปลอดภัยที่เกิดขึ้น

### 3.2.3 โครงการ Open Source MANO (OSM)

โครงการ Open Source MANO (OSM) [27] เป็นโครงการริเริ่มโอเพ่นซอร์ส ที่กลุ่มของนักพัฒนาและผู้ให้บริการร่วมกันสร้างขึ้นมาเพื่อสนับสนุนการบริหารจัดการเน็ตเวิร์คเซอร์วิสต่างๆ โดย OSM นี้มีเป้าหมายหลักในการพัฒนาแพลตฟอร์ม NFV Management and Orchestration (NFV MANO) ที่รองรับมาตรฐาน ETSI NFV เพื่อเป็นตัวกลางในการบริหารจัดการทรัพยกรเครือข่าย ควบคุมการติดตั้งและจำลองเน็ตเวิร์คเซอร์วิสต่างๆ (VNFs) อีกทั้งดูแลและรับผิดชอบในการจัดสรรทรัพยากรเครือข่ายที่จำเป็นต้องใช้สำหรับจำลองเน็ตเวิร์คเซอร์วิส โดยรองรับการใช้งานร่วมกันกับแพลตฟอร์ม OpenStack เพื่อจำลองเน็ตเวิร์คเซอร์วิสบนคลาวด์ ดังนั้นผู้ใช้งานสามารถ

เข้าถึงและนำ source code ของ OSM มาใช้งานได้โดยไม่มีค่าใช้จ่าย และสามารถเข้าร่วมร่วมพัฒนาและปรับปรุงต่อยอดในโครงการ OSM ได้เช่นกัน อีกนัยหนึ่งวัตถุประสงค์หลักของโครงการ OSM สามารถสรุปได้เป็นข้อๆ ดังนี้

1. เพื่อพัฒนา MANO framework สำหรับทำหน้าที่บริหารและจัดการทรัพยากรเครือข่าย จำลองและจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิสให้เป็นไปอย่างราบรื่น อีกทั้งการทำหน้าที่ตรวจสอบและปรับขยายทรัพยากรเครือข่ายให้มีความเหมาะสมกับประสิทธิภาพการประมวลผลของแต่ละเน็ตเวิร์คเซอร์วิส
2. เพื่อพัฒนา MANO framework ที่สามารถรองรับมาตราฐาน ETSI NFV
3. เพื่อส่งเสริมให้เกิดชุมชนโอเพ่นซอร์ส สำหรับเป็นพื้นที่ให้นักพัฒนา ผู้ประกอบการ ผู้ขาย และผู้ที่เกี่ยวข้องสามารถร่วมพัฒนา ทดสอบ ปรับปรุง แบ่งปัน และใช้งานเฟรมเวิร์คดังกล่าวฯ สนับสนุนความก้าวหน้าโดยรวมของเทคโนโลยี NFV
4. เพื่อส่งเสริมให้เกิดการบูรณาการและการทำงานร่วมกันระหว่างกลุ่มคนที่เกี่ยวข้อง OSM มุ่งเน้นให้เกิดการทำงานร่วมกันอย่างราบรื่นเพื่อขับเคลื่อนโซลูชั่น MANO ซึ่งหมายรวมถึง โครงสร้างพื้นฐาน NFV Infrastructure (NFVI) ตัวจัดการโครงสร้างพื้นฐานเสมือน Virtual Infrastructure Managers (VIM) ตลอดจนมี adapters และ APIs เพื่อเป็นช่องทางให้โมดูล แพลตฟอร์มฮาร์ดแวร์ หรือซอฟต์แวร์ต่างๆ สามารถปลั๊กอินเข้ารกับระบบได้
5. เพื่อพัฒนา MANO framework ที่สามารถรองรับกรณีการใช้งานและสถานการณ์การปรับใช้ที่หลากหลาย ซึ่งรวมถึงการแบ่งส่วนเครือข่าย การปรับใช้หลายไซต์ สภาพแวดล้อมแบบหลายผู้เช่า และการรวมระบบคลาวด์แบบไฮบริด เพื่อสร้างความยืดหยุ่นต่อการรองรับความต้องการที่หลากหลาย ให้มีความเหมาะสมและสอดคล้องกับความต้องการเฉพาะของแต่ละกลุ่มผู้ใช้งาน
6. เพื่อสร้างระบบอัตโนมัติและการลดความซับซ้อนของกระบวนการ MANO ช่วยเพิ่มความคล่องตัวในการปรับใช้ การกำหนดค่า และการจัดการ VNF ลดความซับซ้อนในการดำเนินงานและเพิ่มประสิทธิภาพ

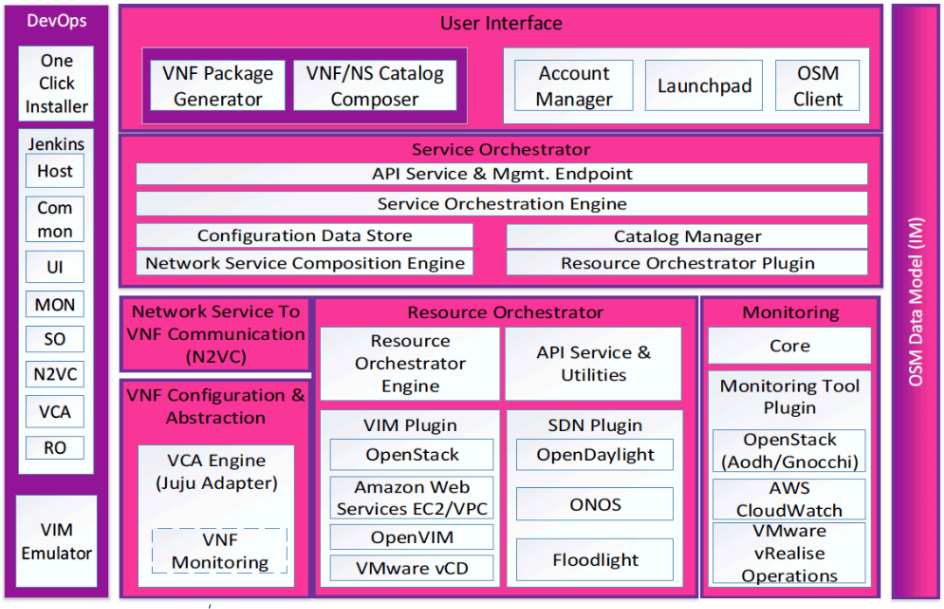



*รูปที่ 3.2.4: สถาปัตยกรรม OSM*

รูปที่ 3.2.4 แสดงสถาปัตยกรรมการออกแบบ OSM สนับสนุนมาตรฐาน ETSI โดยแบ่งออกเป็น 2 ระบบย่อย ดังนี้

- Design-time components: ถูกออกแบบมาเพื่อสนับสนุน model-driven environment รองรับมาตรฐาน ETSI NFV โดยมีหน้าที่รับผิดชอบหลัก คือ การจัดเตรียม VNF package ซึ่งหมายรวมถึงกระบวนการต่างๆ ที่เกี่ยวข้องกับการสร้าง อ่าน อัพเดท และลบข้อมูลหรือชุดคำสั่ง (CRUD) ที่เกี่ยวข้องกับการจำลองเน็ตเวิร์คเซอร์วิสนั้นๆ ตลอดจนการจัดเตรียม GUI เพื่อเป็นช่องทางให้ผู้แลระบบสามารถเรียกดูข้อมูลและเข้าถึงการตั้งค่าต่างๆ ได้
- Run-time components: ทำหน้าที่ในจัดเตรียมทรัพยากรเครือข่าย บริหารและจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น ตรวจสอบประสิทธิภาพการประมวลผลและสถานะการทำงาน ประสานและเชื่อมต่อกับโมดูลอื่นๆ ภายในระบบ ดังเช่น SDN controller เพื่อกำหนดเส้นทางการส่งต่อแพ็กเก็ตหรือ public cloud infrastructure ของผู้ให้บริการรายอื่นสำหรับการจำลองเน็ตเวิร์คเซอร์วิสต่างๆ ที่ต้องการ จากรูปที่ 3.2.5 หากมองจากโครงสร้างโดยรวมจะพบว่าในระบบ Run-time components นั้นมีการออกแบบโครงสร้างแบบชั้นๆ ซึ่งในแต่ละชั้นประกอบด้วยโมดูลต่าง ๆ โดยมีรายละเอียดดังนี้
    - User interface module: ได้ถูกออกแบบมาเพื่อเป็น Launchpad (GUI) ของ OSM โดยสามารถใช้ในการจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น เช่น สร้าง แก้ไข หรือลบเน็ตเวิร์คเซอร์วิส โดยผู้ใช้สามารถเรียกดูข้อมูลลักษณะทางกายภาพของเครือข่ายและสามารถแสดงสถิติแบบเรียลไทม์ ซึ่ง Launchpad นี้ใช้ Northbound REST API เพื่อติดต่อสื่อสารกับโมดูลต่างๆ ในระบบ OSM และมี CLI client (REST API) สำหรับอำนวยความสะดวกให้ผู้ที่เกี่ยวข้องสามารถเข้าถึงระบบจากระยะไกล
    - Service orchestration module**:** มีหน้าที่รับผิดชอบการจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสทั้งหมดใน OSM สามารถรองรับ multi-tenancy และ role-based access control โดยทำงานร่วมกับโมดูลอื่นๆ เช่น Configuration Data Store, Network Service Composition Engine, Catalogue Manager และปลั๊กอินสำหรับ resource orchestrators โดยมี monitoring module คอยตรวจสอบการทำงานอยู่เบื้องหลัง
    - Network service to VNF communication module: มีหน้าที่หลักในการจัดการการสื่อสารระหว่าง Service orchestrator (SO) และ VNF communication module ช่วยให้ SO สามารถสื่อสารกับ VNF ผ่านปลั๊กอิน
    - VNF configuration and abstraction (VCA) module: กำหนดการตั้งค่าของแต่ละเน็ตเวิร์คเซอร์วิส (VNF) ที่อัปโหลดโดย SO
    - Resource orchestrator (RO) module: ทำหน้าที่ในการจัดสรรทรัพยากรเครือข่ายต่างๆ ที่เกี่ยวข้องให้พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิส โดยขีดความสามารถของ RO สามารถสนับสนุนและรองรับโครงสร้างพื้นฐานที่หลากหลาย เช่น Openstack, AWS, Google Cloud Platform, ฯลฯ

- Monitoring module: ทำหน้าที่ในการรวบรวมข้อมูลเกี่ยวกับสถานะและประสิทธิภาพการทำงานของแต่ละเน็ตเวิร์คเซอร์วิส ข้อมูลนี้จะถูกใช้เพื่อตรวจสอบว่าแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้นทำงานได้อย่างถูกต้องหรือไม่ หากมีสัญญาณบ่งชี้ว่ามีปัญหา โมดูลนี้จะทำหน้าที่แจ้งเตือนผู้ดูแลระบบ เพื่อให้สามารถดำเนินการแก้ไขได้อย่างทันถ่วงที




*รูปที่ 3.2.5: โมดูลต่างๆ ภายในระบบ Run-time components*

สำหรับขีดความสามารถในการพัฒนาแพลตฟอร์ม ผู้พัฒนาระบบได้มีการระบุถึงคุณสมบัติหลักๆ โดยมุ่งเน้นการเพิ่มขีดความสามารถในการควบคุมการเข้าถึงทรัพยการและการบริหารจัดการเน็ตเวิร์คเซอร์วิส [28-30] ซึ่งมีคุณสมบัติหลักๆ ที่สำคัญ ดังต่อไปนี้

- ด้านความปลอดภัย: ระบบได้มีการควบคุมการเข้าถึงข้อมูลภายในระบบ โดยการจำกัดสิทธิ์และบทบาทของผู้ใช้งานด้วยการประยุกต์ใช้ (1) Role-based access control เช่น การให้สิทธิ์ในการสร้าง ปรับปรุงการตั้งค่า และถอดถอนเน็ตเวิร์คเซอร์วิส หรือการให้สิทธิ์ในการเข้าถึงข้อมูลบนโครงสร้างพื้นฐานคลาวด์ ฯลฯ และใช้ (2) Multi-tenancy/project (หรือ multi-project access control) เป็นตัวจัดการการเข้าถึงทรัพยากรต่างๆ ในแต่ละโปรเจ็คด้วย Role-based access control ซึ่งจะช่วยให้ผู้ใช้สามารถกำหนดสิทธิ์การเข้าถึงทรัพยากรที่อยู่ในโปรเจ็คได้อย่างละเอียดมากขึ้น เช่น ผู้ใช้สามารถได้รับสิทธิ์ในการสร้างและจัดการเน็ตเวิร์คเซอร์วิสในโปรเจ็ค เป็นต้น
- การรับประกันการให้บริการ: นอกเหนือจากขีดความสามารถด้านความปลอดภัย การพัฒนาแพลตฟอร์ม OSM นี้ยังมุ่งเน้นด้านการรับประกันการให้บริการด้านอื่นๆ อีกด้วย ซึ่งคลอบคลุมถึง (1) ขีดความสามารถในการปรับขยายทรัพยากรเครือข่าย เช่น การเพิ่มหรือลดจำนวนของเน็ตเวิร์คเซอร์วิสได้ตามความต้องการเพื่อให้มีความเหมาะสมกับจำนวนเวิร์คโหลดในช่วงเวลานั้นๆ (2) ความยืดหยุ่นในการกำหนดหรือระบุ cloud infrastructure ที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิส โดยผู้ใช้สามารเลือกได้ว่าจะจำลองเน็ตเวิร์ค

เซอร์วิสทั้งหมดบน cloud infrastructure เดียวกันหรือคนละที่ก็ได้ (3) ความสามารถในการเข้าถึง cloud infrastructure จากหลายๆ ที่พร้อมกันได้ (4) ความสามารถในการตรวจสอบสถานะของบริการเครือข่าย รองรับการแจ้งเตือนต่างๆ ผ่านทาง Kafka consumer และ (5) ความสามารถในการอัพเกรดซอฟต์แวร์ในแต่ละเน็ตเวิร์คเซอร์วิสที่กำลังทำงานได้โดยอัตโนมัติ โดยไม่จำเป็นต้องหยุดการให้บริการ

- ความยืดหยุ่น: การพัฒนาแพลตฟอร์ม OSM ยังคำนึงถึงขีดความสามารถในการทนทางต่อความล้มเหลว เพื่อที่จะให้ระบบสามารถทำงานต่อไปได้หากพบว่าบางส่วนของระบบเกิดความล้มเหลว
- การทำงานร่วมกัน: แพลตฟอร์ม OSM สามารถรองรับและสนับสนุน cloud infrastructure หลากหลาย เช่น Openstack, AWS, Google Cloud Platform, Microsoft Azure ฯลฯ ช่วยให้ผู้ใช้สามารถเลือกใช้งานได้อย่างเหมาะสมกับความต้องการใช้งาน

### 3.2.4 โครงการ ONAP

โครงการ ONAP (Open Network Automation Platform) [31-33] เป็นโครงการที่เกิดจากการรวมตัวกันของ 2 โครงการ ได้แก่ โครงการ Open-O (Linux Foundations Open Orchestration) และ โครงการ AT&T ECOMP (Enhanced Control, Orchestration, Management, and Policy) เข้าด้วยกัน ซึ่งเป็นโครงการ open source ที่ให้ความสำคัญในเรื่องของการออกแบบ การขับเคลื่อนนโยบาย การบริหารจัดการเครือข่าย การวิเคราะห์การจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส และการตั้งค่าอัตโนมัติ โดยมุ่งเน้นการรวมฟังก์ชันที่เกี่ยวข้องกับการจัดการระบบอัตโนมัติทั้งหมดไว้ในระบบเดียวกัน ได้แก่ การจัดการการเชื่อมต่อเครือข่าย (network connection management) การประสานและจัดสรรทรัพยากรเครือข่าย (resource orchestration) และการรับรองการให้บริการ (service assurance) โดยโครงการ ONAP นี้ได้นำเสนอ NFV MANO ที่เรียกว่า OOM (ONAP Operations Manager) ซึ่งสนับสนุนการจัดการเน็ตเวิร์คเซอร์วิส ดังเช่น การตั้งค่าการใช้งาน หรือการควบคุมสถานะการให้บริการในแต่ละเน็ตเวิร์คเซอร์วิส ตลอดจน ความดูแลและรับผิดชอบในการจัดการโครงสร้างพื้นฐาน (NFV Infrastructure) เพื่อจัดเตรียมทรัพยากรต่างๆ (เช่น ทรัพยากรด้านการประมวลผล หน่วยความจำ พื้นที่จัดเก็บข้อมูล และระบบเครือข่าย) ให้พร้อมสำหรับใช้งาน รูปที่ 3.2.6 แสดงแนวคิดโดยรวมในการออกแบบสถาปัตยกรรม ONAP ซึ่งแบ่งออกเป็น 2 เฟรมเวิร์คหลัก ได้แก่

- Design-time framework: ทำหน้าที่ในการจัดเตรียมสภาพแวดล้อมต่างๆ โดยครอบคลุมถึงเครื่องมือ เทคนิค ข้อกำหนด และแหล่งเก็บข้อมูล (repositories) ต่างๆ ที่จำเป็นสำหรับการสร้างและจำลองเน็ตเวิร์คเซอร์วิส ทั้งนี้ผู้ใช้สามารถเข้าถึงการตั้งค่า ออกแบบ กำหนดนโยบาย และติดตามสถานะต่างๆ ผ่าน web portal โดยภายใน Design-time framework ประกอบด้วยโมดูลต่างๆ ดังนี้
    - Service design and creation: ทำหน้าที่ในสร้างและการออกแบบเน็ตเวิร์คเซอร์วิส โดยใช้ Service Design and Creation (SDC) เป็นเครื่องมือในการออกแบบและสร้างแบบจำลอง
    - Policy design: ส่วนที่ใช้ในการออกแบบและกำหนดนโยบายหรือกฎเกณฑ์ต่างๆ สำหรับควบคุมการทำงานของระบบ ONAP

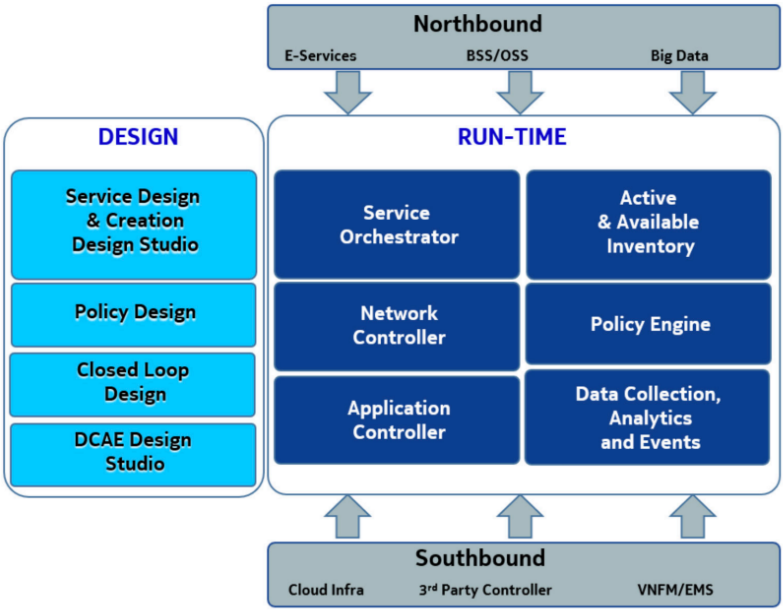
- Closed-lood design: ทำหน้าที่ในการออกแบบและจัดการการทำงานภายในระบบให้สามารถทำงานได้โดยอัตโนมัติ
- DCAE design studio : ทำหน้าที่ในการออกแบบเครื่องมือ DCAE ซึ่งประกอบด้วย การรวบรวมข้อมูล (data collection) การวิเคราะห์ข้อมูล (analytics) และการจัดการกับเหตุการณ์ที่เกิดขึ้น (events)

- Run-time framework: ส่วนนี้จะรับผิดชอบในการรันคำสั่งและจำลองเน็ตเวิร์คเซอร์วิสตามนโยบายที่ได้กำหนดไว้ ตลอดจนทำหน้าที่เชื่อมต่อกับโมดูลอื่นๆ ดังนี้
  - Service orchestrator: ทำหน้าที่บริหารและจัดการทรัพยากรเครือข่ายในระบบ รวมถึงเน็ตเวิร์คเซอร์วิสต่างๆ ที่เกี่ยวข้อง โดย Service orchestrator จะทำงานร่วมกันโมดูลอื่นๆ ได้แก่ SDC, network controller, application controller, AAI เพื่อค้นหาและคัดเลือกโมดูลที่เกี่ยวข้องที่จำเป็นต้องใช้สำหรับสร้างเน็ตเวิร์คเซอร์วิสต่างๆ ที่ต้องการ และใช้ OpenStack API เพื่อสร้างและจัดการ virtual resources ที่จำเป็นสำหรับการจำลองเน็ตเวิร์คเซอร์วิสดังกล่าวฯ
  - Active & Available Inventory (AAI): เพื่อติดตามและตรวจสอบสถานะการใช้ทรัพยากรของระบบ โดยมี Microservices Bus เพื่ออำนวยความสะดวกในการลงทะเบียน/ค้นหาบริการ และรองรับ API ภายในและภายนอก รวมถึงชุดพัฒนาซอฟต์แวร์หลัก (SDK) OOM สำหรับใช้ประโยชน์จาก Kubernetes
  - Network controller (หรือ SDN controller): สำหรับทำหน้าที่ในการจัดการและกำหนดเส้นทางการส่งต่อแพ็กเก็ตข้อมูลบนเครือข่าย
  - Application controller: เพื่อจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น
  - DCAE (Data collection, Analytics, and Events): เพื่อรวบรวมข้อมูล วิเคราะห์ผล และจัดการกับเหตุการณ์ที่เกิดขึ้น รองรับการควบคุมการทำงานแบบ Closed-loop

นอกเหนือจากนี้ โครงการ ONAP ได้ระบุคุณสมบัติของระบบดังสามารถสรุปได้เป็นข้อๆ ดังต่อไปนี้

- มีเครื่องมือที่ช่วยอำนวยความสะดวกในการจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น ตั้งแต่การออกแบบ พัฒนา ติดตั้ง จัดการ และถอดถอนเน็ตเวิร์คเซอร์วิสดังกล่าวแบบครบวงจร
- มีการออกแบบสถาปัตยกรรมระบบสนับสนุนการขับเคลื่อนด้วยข้อมูลและนโยบาย (Metadata-driven and policy-driven architecture) ช่วยให้ผู้ใช้งานสามารถกำหนดค่าและจัดการบริการต่างๆ ได้อย่างยืดหยุ่นและง่ายดาย
- มีขีดความสามารถในการปรับลดจำนวนหรือขยายขนาดของทรัพยากรที่ใช้งานได้อย่างไดนามิก เพื่อให้มีความสอดคล้องและเหมาะสมต่อการรองรับปริมาณการใช้งานที่สูงและจำนวนผู้ใช้จำนวนมาก
- รองรับระบบอัตโนมัติแบบ Closed-loop automation ช่วยให้ผู้ใช้งานสามารถลดข้อผิดพลาดและเพิ่มประสิทธิภาพในการบริหารและจัดการเครือข่าย

- สนับสนุนการดูแลระบบผ่านแดชบอร์ดเพื่อวิเคราะห์และตรวจสอบเหตุการณ์ที่สนใจในระหว่างการออกแบบและการทำงาน ทำให้ง่ายต่อการแก้ไขปัญหาหากพบข้อผิดพลาด
- สนับสนุนการทำงานแบบบูรณาการ ช่วยให้ผู้ให้บริการเครือข่ายสามารถจำลองเน็ตเวิร์คเซอร์วิสต่างๆ ที่มาจากต่างผู้จำหน่าย (vendors) ให้สามารถรันบนแพลตฟอร์มเดียวกันได้ โดย ONAP ได้ออกแบบและพัฒนา mapper components เพื่อแปลงข้อกำหนดต่างๆ จากแต่ละเน็ตเวิร์คเซอร์วิสจากผู้จำหน่ายต่างๆ เป็นข้อกำหนดมาตรฐานที่สามารถเข้าใจได้โดยระบบ ONAP




*รูปที่ 3.2.6: สถาปัตยกรรม ONAP*

### 3.2.5 โครงการ PALANTIR

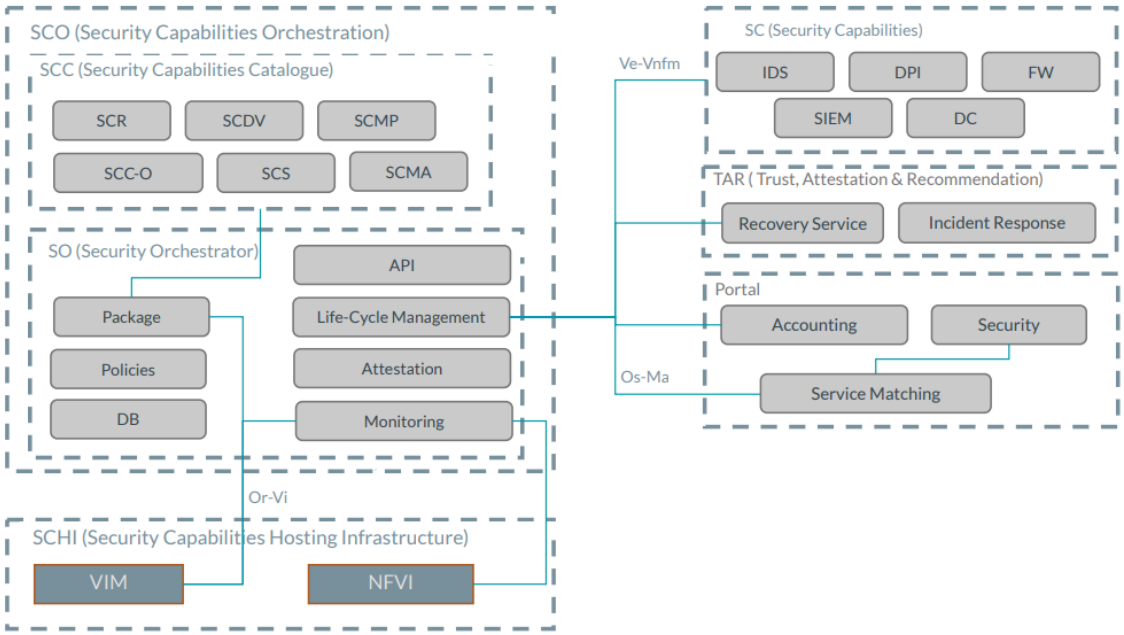
โครงการ PALANTIR [34] เป็นโครงการได้รับทุนจากสหภาพยุโรปซึ่งก่อตั้งขึ้นมาจากหน่วยงานภาครัฐและภาคเอกชน โดยมีวัตถุประสงค์เพื่อสร้างนวัตกรรมที่สามารถช่วยป้องกันการโจมตีและการกระทำที่ไม่เหมาะสมต่อระบบเครือข่ายในระดับวิสาหกิจขนาดกลาง (SMEs) หรือ Small and Medium Enterprises ด้วยการนำเสนอกลยุทธ์การมุ่งแก้ปัญหาด้านความปลอดภัย ดังนี้ (1) มีการประยุกต์ใช้เทคโนโลยี NFV และ SDN เป็นแนวทางในการพัฒนาระบบเครือข่ายในมิติที่แตกต่างกันไป (2) มีการประยุกต์ใช้เทคโนโลยีปัญญาประดิษฐ์ (artificial intelligence) เพื่อสร้างกระบวนทัศน์ใหม่ต่อการตรวจสอบความเสี่ยงและวิเคราะห์ประเภทของภัยคุกคามที่อาจจะเกิดขึ้น และ (3) การกำกับให้ความรับรองว่าข้อมูลขององค์กรจะได้รับความเป็นส่วนตัวและไม่ถูกละเมิด โดยในโครงการนี้มีการพัฒนาแพลตฟอร์ม NFV MANO ซึ่งเป็นส่วนสำคัญของการจัดการและควบคุมการบริการในระบบเครือข่ายโดยอาศัยการใช้เทคโนโลยีเสมือนจริง (virtualization) ซึ่งทำให้ง่ายต่อการตั้งค่าและการจัดการระบบเครือข่าย อีกทั้งมุ่งเน้นการพัฒนาและการใช้งานโปรแกรมโอเพ่นซอร์ส (Open Source) ในการพัฒนาระบบเพื่อให้การใช้งานเป็นไปอย่างต่อเนื่อง รวดเร็ว และมีความยืดหยุ่นในการใช้งาน โดยการสนับสนุนการพัฒนาระบบ

เครือข่ายโดยใช้เทคโนโลยี NFV MANO ในการควบคุมและจัดการบริการ โดยเฉพาะในระดับ SMEs ซึ่งเป็นส่วนสำคัญในการสร้างเศรษฐกิจและการพัฒนาอย่างยั่งยืนในระดับชุมชนและประเทศต่าง ๆ

ในอีกมุมหนึ่ง โครงการ PALANTIR ได้ให้ความสำคัญต่อการการแก้ไขเหตุการณ์ด้านความปลอดภัยที่มุ่งปกป้องค์กรขนาดกลาง รวมถึงวิสาหกิจขนาดกลาง และขนาดย่อม จากภัยคุกคามทางไซเบอร์ในสภาพแวดล้อม โดยนำเสนอ 3 เทคนิคมาช่วยในการตรวจสอบและวิเคราะห์ภัยคุกคามที่เกิดขึ้น

1. มีการนำเสนอ ontology ด้านความปลอดภัยเพื่ออธิบายความสัมพันธ์และคุณลักษณะเฉพาะด้านความปลอดภัย และใช้ฟังก์ชันการจับคู่บริการเพื่อระบุกลไกความปลอดภัยที่เกี่ยวข้องตามภัยคุกคามทางไซเบอร์และผลกระทบทางการเงินที่อาจเกิดขึ้นกับธุรกิจ และใช้รูปแบบความสัมพันธ์ฟังก์ชันการจับคู่บริการเพื่อระบุกลไกความปลอดภัยที่เกี่ยวข้องตามภัยคุกคามทางไซเบอร์และผลกระทบทางการเงินที่อาจเกิดขึ้นกับธุรกิจ
2. มีการอ้างอิงนโยบายการรักษาความปลอดภัยร่วมกับการประยุกต์ใช้ Finite State Machine เพื่อติดตามความเสี่ยงและแก้ปัญหาภัยคุกคามที่อาจจะเกิดขึ้นในแต่ละกระบวนการ นำมาซึ่งขีดความสามารถที่สามารถรับประกันได้ว่าเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้นสามารถตรวจสอบและทำงานตรงตามนโยบายการรักษาความปลอดภัยที่กำหนดไว้ได้
3. มีการนำเสนอแนวทางการป้องกันและตอบโต้ภัยคุกคามได้อย่างตรงจุด โดยครอบคลุมถึงการจัดเตรียมทรัพยากรเครือข่ายและเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ สำหรับให้องค์กรเรียกใช้งาน

นอกเหนือจากนี้ โครงการนี้ยังครอบคลุมถึงพิมพ์เขียวของการจัดการด้านความปลอดภัย โดยมีความคล้ายคลึงกันกับเฟรมเวิร์ก เช่น ETSI NFV MANO และ SBI มีการนำต้นแบบการพิสูจน์แนวคิดมาใช้สำหรับการประเมินเชิงปริมาณของประสิทธิภาพของขั้นตอนการแก้ไขในรูปแบบการปรับใช้ต่าง ๆ เช่น การปรับแต่ง




*รูปที่ 3.2.7: แนวคิดโดยรวมของสถาปัตยกรรม PALANTIR*

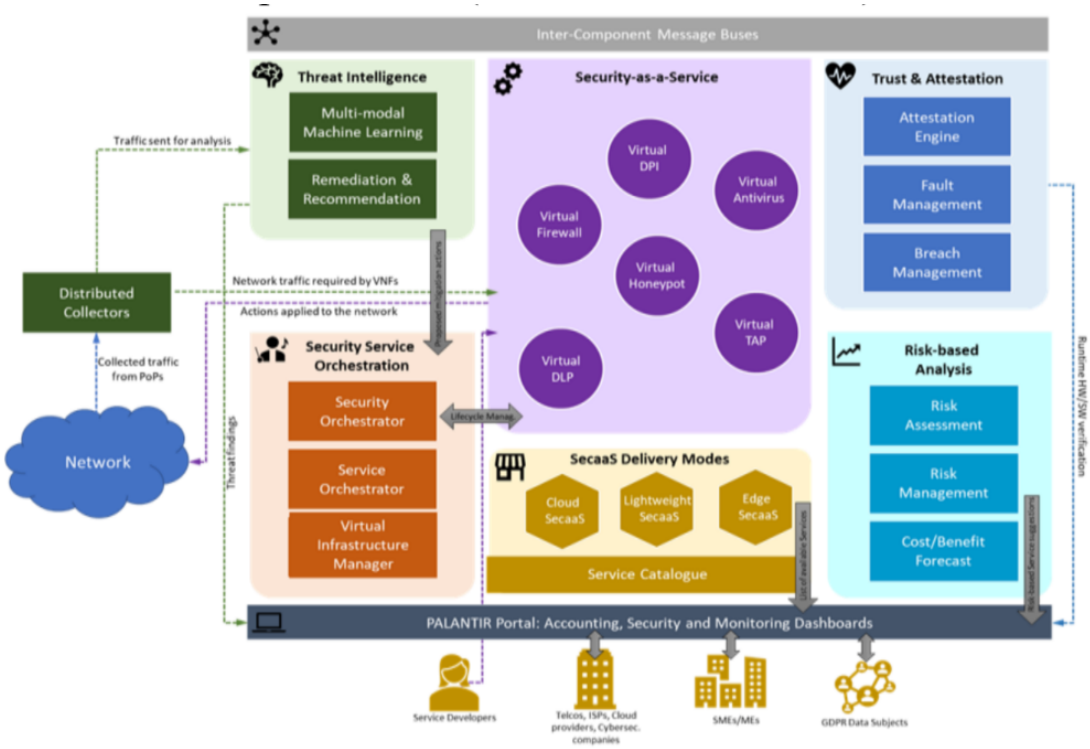
ontology เพื่อรวมคุณสมบัติทรัพยากรเฉพาะ ปรับปรุงการจับคู่บริการเพื่อพิจารณาข้อมูลเฉพาะทรัพยากร เพิ่มประสิทธิภาพความสามารถด้านความปลอดภัยตามคุณลักษณะของทรัพยากร ประโยชน์ของวิธีการเหล่านี้ ได้แก่ การอำนวยความสะดวกในการประสานงานตอบสนองต่อภัยคุกคามทางไซเบอร์ในสถานการณ์หลายไซต์ ทำให้ผู้ใช้บริการสามารถเปลี่ยนเครื่องมือป้องกันได้โดยไม่มีความยุ่งยาก และเปิดใช้งานการจัดการภัยคุกคามเชิงรุกโดยผสมผสานการประเมินความเสี่ยงและการแบ่งปันภัยคุกคามได้ รูปที่ 3.2.7 แสดงแนวคิดโดยรวมของสถาปัตยกรรม PALANTIR ซึ่งประกอบด้วย 4 ส่วนหลัก ดังนี้

- Security Capabilities Catalogue (SCC): เป็นเสมือนแหล่งจัดเก็บข้อมูลต่างๆ เช่น metadata, VNF packages, software images, ชุดคำสั่ง, ข้อกำหนดด้านความปลอดภัย (security policy), ฯลฯ ที่เกี่ยวข้องสำหรับใช้ในการจำลองเน็ตเวิร์คเซอร์วิส
- Security Capabilities (SC): ทำหน้าที่ในการจัดเตรียมเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยพื้นฐานประเภทต่างๆ เช่น IDS, DPI, firewall ที่จำเป็นต้องใช้สำหรับให้องค์กรต่างๆ สามารถเรียกใช้งานหรือใช้ประโยชน์ต่อการช่วยตรวจสอบและป้องกันภัยคุกคาม
- Security Capabilities Orchestration (SCO): ทำหน้าที่ประสาน บริหารและจัดการระบบเครือข่ายในองค์กรให้เป็นไปอย่างอัตโนมัติ
- Security Capabilities Hosting Infrastructure: โครงสร้างเครือข่ายพื้นฐานที่จัดเตรียมเอาไว้สำหรับการติดตั้งระบบและการจำลองเน็ตเวิร์คเซอร์วิส

อีกนัยหนึ่ง แนวทางการออกแบบสถาปัตยกรรม PALANTIR [35-36] อ้างอิงรูปแบบวิธีที่เรียกว่า "scenario-based approach" ซึ่งเป็นการออกแบบที่ยึดเหตุการณ์ที่สนใจหรือกรณีศึกษาที่สนใจเป็นหลัก ประกอบด้วยกระบวนการย่อยๆ ดังนี้ (1) การระบุและอธิบายสถานการณ์ที่สนใจ (Scenario identification and description) (2) การเชื่อมโยงระหว่างข้อกำหนดหรือนโยบายที่ต้องการกำหนดสู่การเลือกใช้เน็ตเวิร์คเซอร์วิสที่เหมาะสม (Mapping requirements to desired functionalities) (3) การระบุส่วนประกอบและอินเทอร์เฟซต่างๆ (Defining components and interfaces) (4) การเชื่อมโยงโครงสร้างทั้งหมดเข้าด้วยกัน (Connecting the entire structure) และสุดท้าย (5) การปรับปรุงและการเพิ่มขีดความสามารถ (Adaptability and enhancements) โดยขั้นตอนทั้งหมดนี้จะถูกนำมาเป็นโครงสร้าง PALANTIR ที่ประกอบไปด้วยส่วนประกอบต่างๆ ที่ครอบคลุมการทำงานของระบบทั้งหมด และมีระบบ E2E security enabled ในการรักษาความปลอดภัยของข้อมูลตลอดวงจรชีวิต ไม่ให้ข้อมูลถูกเข้าถึง เปลี่ยนแปลง หรือทำลายโดยบุคคลที่ไม่ได้รับอนุญาต โดยโครงสร้างระดับสูงของระบบ PALANTIR (แสดงดังรูปที่ 3.2.8) ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

- Threat intelligence: เป็นส่วนหลักในการตรวจจับความเสี่ยง การวิเคราะห์ และมาตรการป้องกันซึ่งประกอบด้วยแบบจำลองที่ได้รับการปรับปรุงสำหรับการตรวจจับและโจมตี และนำเสนอข้อเสนอแนะเพื่อแก้ไขและแนะนำในกรณีของความเสี่ยงที่เป็นไปได้
- Security service orchestration: เป็นระบบหลังบ้านที่รับผิดชอบในการจัดการโครงสร้างพื้นฐานและบริหารจัดการเน็ตเวิร์คเซอร์วิสต่างๆ โดยมุ่งเน้นด้านความปลอดภัยเป็นหลัก

- Security-as-a-Service (SecaaS): ส่วนนี้จะครอบคลุมเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ ที่ถูกจำลองขึ้นและให้บริการโดยแพลตฟอร์ม PALANTIR โดยในแต่ละเน็ตเวิร์คเซอร์วิสที่ถูกจำลองขึ้นมานั้นจะมีการตั้งค่าและกำหนดนโยบายด้านความปลอดภัยที่แตกต่างกัน โดยขึ้นอยู่กับบริบทและความต้องการใช้งานที่ได้ระบุไว้
- SecaaS delivery modes: เป็นส่วนที่ทำหน้าที่ออกแบบ กำหนด และคัดเลือกประเภทเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่เหมาะสม สอดคล้องกับสภาพแวดล้อมและความต้องการเฉพาะของผู้ใช้งานหรือบริบทขององค์กร
- Trust & attestation: เป็นส่วนที่ทำหน้าที่รักษาความปลอดภัยในระดับโครงสร้างพื้นฐานของแพลตฟอร์มเพื่อรับรองความถูกต้องของข้อมูล
- Risk-based Analysis: ส่วนนี้ที่ทำหน้าที่ในการประเมินความเสี่ยงและการดำเนินการจัดการกับความเสี่ยง




***รูปที่ 3.2.8:*** *สถาปัตยกรรมโดยรวมของระบบ PALANTIR*

กล่าวโดยสรุป โครงการ PALANTIR มุ่งเน้นที่จะพัฒนาระบบเพื่อให้มีขีดความสามารถและมีคุณสมบัติหลัก 5 ประการ ดังนี้

- Risk-based analysis framework: มีความสามารถในช่วยตรวจสอบและวิเคราะห์ความเสี่ยงหรือภัยคุกคามที่อาจจะเกิดขึ้น
- SecaaS capabilities: มีความสามารถในการจำลองและให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ ที่เกี่ยวข้องได้อย่างเหมาะสม สอดคล้องกับบริบทและความต้องการใช้งานของแต่ละองค์กร

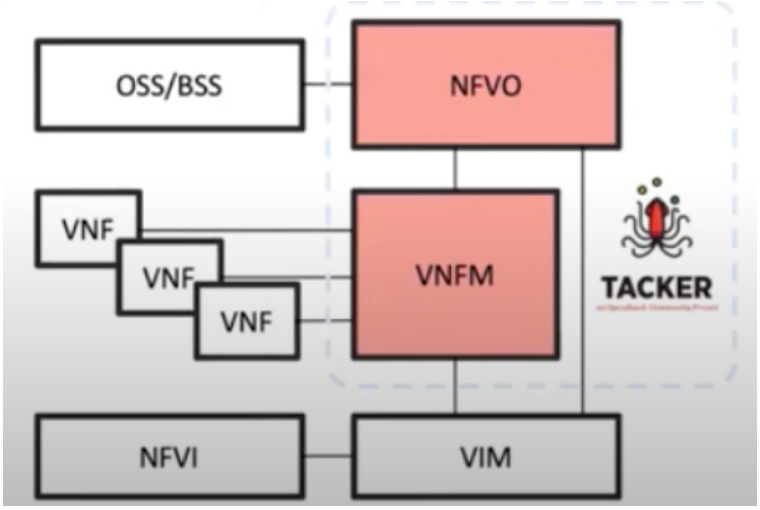
- Threat intelligence: มีความสามารถในการตรวจสอบและวิเคราะห์ความเสี่ยง ช่องโหว่ หรือภัยคุกคามได้อย่างชาญฉลาด ด้วยการประยุกต์ใช้เทคนิค machine learning ช่วยให้องค์กรสามารถปรับปรุงประสิทธิภาพด้านความปลอดภัยและลดความเสี่ยงจากภัยคุกคามได้อย่างมีประสิทธิภาพ ถูกต้อง และตรงจุด
- Trust, attestation & recovery: มุ่งเน้น (1) การสร้างความน่าเชื่อถือในสถานะและความน่าเชื่อถือของส่วนประกอบต่าง ๆ ว่าระบบทำงานอย่างถูกต้อง และไม่ได้รับผลกระทบจากการละเมิดความปลอดภัยหรือการโจมตี (Trust), (2) การตรวจสอบสถานะในระบบเพื่อตรวจสอบว่าส่วนนี้มีสถานะที่ถูกต้องและปกติ (Attestation), และ (3) การกู้คืนระบบ หากพบว่ามีความผิดปกติ การโจมตี หรือความเสี่ยงที่ส่งผลกระทบต่อความปลอดภัยและประสิทธิภาพของระบบ (Recovery)
- Accounting dashboard: การจัดการบัญชีตามบทบาทผู้ใช้งาน (user role) เช่น service developers, PALANTIR providers และผู้ใช้ทั่วไป โดยให้บริการการติดตามการแจ้งเตือนที่เกี่ยวข้องกับสถานะการประมวลผลและการให้บริการของแต่ละเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่เกี่ยวข้องผ่านแดชบอร์ด

### 3.2.6 โครงการ KDDI & NEC

โครงการนี้ [37] เป็นโครงการที่เกิดจากความร่วมมือระหว่างบริษัท KDDI และ NEC ในการพัฒนาและปรับปรุงโซลูชันการให้บริการสื่อสารและเครือข่าย โดยเน้นที่การใช้แพลตฟอร์ม NFV MANO ซึ่งเป็นส่วนหนึ่งของการพัฒนาระบบเครือข่ายเสมือนซึ่งมุ่งเน้นการทำงานแบบอัตโนมัติ โดยการนำเทคโนโลยี NFV MANO มาใช้งานจะช่วยให้สามารถจัดการกับการใช้งานทรัพยากรของระบบเครือข่ายได้อย่างเหมาะสมและมีประสิทธิภาพมากยิ่งขึ้น โครงการนี้มีการใช้งานและเปรียบเทียบผลการทดสอบระหว่างเทคโนโลยี NFV MANO ต่าง ๆ เพื่อหาวิธีการทำงานที่มีประสิทธิภาพและเหมาะสมในการจัดการทรัพยากรของระบบเครือข่าย โดยเฉพาะการใช้งานของ OpenStack Tacker (แสดงดังรูปที่ 3.2.9) ซึ่งเป็นโมดูลของ OpenStack ในการจัดการและปรับปรุงระบบเครือข่ายแบบอัตโนมัติ เพื่อให้สามารถทำงานร่วมกับเทคโนโลยีอื่นๆ เช่น OpenDaylight ซึ่งเป็น SDN Controller ในการควบคุมการส่งข้อมูลในระบบเครือข่าย และการใช้งานเน็ตเวิร์คเซอร์วิสต่างๆ ซึ่งเป็นส่วนที่สำคัญในการจัดการระบบเครือข่ายแบบอัตโนมัติ โดยเฉพาะการประยุกต์ใช้แพลตฟอร์ม Tacker ที่พัฒนาขึ้นบน OpenStack ร่วมด้วยเพื่อทำหน้าที่บริหารจัดการและประสานการทำงานของโมดูลต่างๆ ภายในระบบให้เป็นไปอย่างราบรื่น โดยมีส่วนประกอบ VNF Manager (VNFM) และ NFV Orchestration (NFVO) ตามมาตรฐาน ETSI NFV ได้รับการสนับสนุนจากผู้จำหน่ายและผู้ให้บริการเครือข่ายต่าง ๆ รวมถึง NEC, Brocade, 99cloud และ Intel ซึ่งมีส่วนร่วมในการพัฒนาและต่อยอดโมดูลต่างๆ ที่เกี่ยวข้องในแพลตฟอร์ม Tacker นี้ ทำให้ Tacker จึงกลายเป็นโอเพ่นซอร์สที่ได้รับความนิยมและใช้งานกันในวงกว้าง โดยเน้นเป็นพิเศษไปที่กรณีการใช้งาน 5G Core (5GC) และรวมเอา Kubernetes เข้ากับ OpenStack เพื่อให้สามารถอยู่ร่วมกันได้ของ Containerized Network Functions (CNF) และ Virtualized Network Functions (VNF) โดยสรุปแล้ว Tacker พัฒนาขึ้นเพื่อจัดการกับกรณีการใช้งานที่หลากหลาย รวมถึงการรวม VNF และ CNF โดยใช้ประโยชน์จากมาตรฐาน ETSI NFV

สำหรับการออกแบบสถาปัตยกรรมในโครงการ KDDI & NEC จะมุ่งเน้นการนำ Tacker มาปรับปรุงใช้งาน เนื่องจากจุดเด่นหลักของ Tacker นำมาซึ่งความ lightweight ของระบบ และมีความง่ายต่อการใช้งาน อีกทั้งสนับสนุนความเป็นมาตรฐาน ETSI NFV ที่รองรับฟังก์ชันการทำงานที่จำเป็นทั้งหมด รวมถึงการที่เป็น open source software ที่สามารถรองรับกับการใช้งานร่วมกับ VNF และ CNF จากผู้ผลิตหลายราย (Muti-Vendor) ได้ง่ายขึ้น โดยสถาปัตยกรรมของระบบประกอบด้วยโมดูลต่างๆ ดังนี้

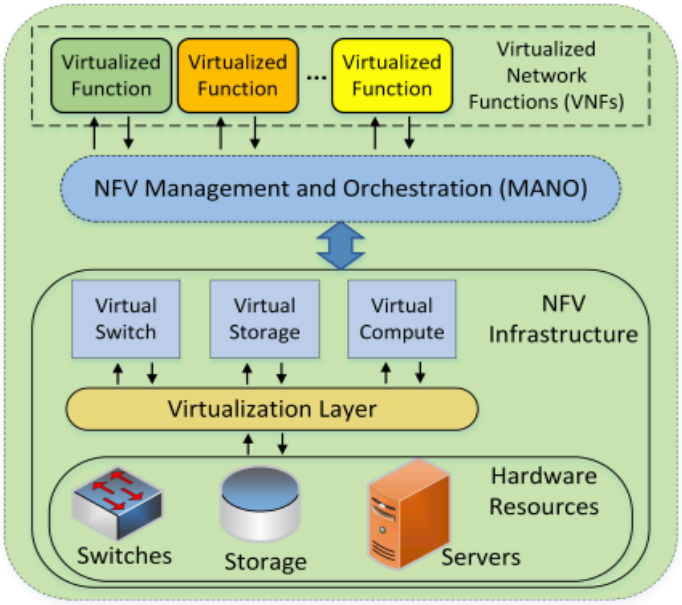
- VNF catalog management: ทำหน้าที่จัดการแค็ตตาล็อก (catalog) ของแต่ละเน็ตเวิร์คเซอร์วิส ซึ่งรวมถึงเก็บข้อมูลต่างๆ ที่เกี่ยวข้อง (VNF descriptors) ในฐานข้อมูล เพื่อให้สามารถเข้าถึงและจัดการข้อมูลดังกล่าวฯ ได้ตามต้องการ
- VNF lifecycle management: ทำหน้าที่จัดวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส ซึ่งรวมถึงการสร้าง แก้ไข และถอดถอนเน็ตเวิร์คเซอร์วิสที่ต้องการ
- VNF configuration management: ทำหน้าที่กำหนด แก้ไข หรือดำเนินการตั้งค่าในแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น
- Health Monitoring : รองรับการตรวจสอบสุขภาพของแต่ละเน็ตเวิร์คเซอร์วิส
- Self-Healing: รองรับการทำงานที่สามารถซ่อมแซมตนเองตามนโยบายที่กำหนดไว้ใน VNFD ได้โดยอัตโนมัติเมื่อพบปัญหาหรือสถานการณ์ที่ต้องการแก้ไขหรือเปลี่ยนแปลงรายละเอียดใน VNFD




***รูปที่ 3.2.9:*** *แนวคิดโดยรวมของสถาปัตยกรรม OpenStack Tacker*

### 3.3 ดำเนินการศึกษา สำรวจ และวิจัยความเป็นไปได้ในการเลือกใช้ open source software tools ต่างๆ ที่เกี่ยวข้อง

ในหัวข้อนี้จะลงลึกในรายละเอียดเกี่ยวกับเครื่องมือ open source software ต่างๆ ที่เกี่ยวข้องสำหรับใช้ในการพัฒนาแพลตฟอร์มบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง (management and orchestration platform) โดยเพื่อให้ง่ายต่อการทำความเข้าใจจะขอแบ่งเครื่องมือที่เกี่ยวข้องออกเป็น 3 หมวดหมู่หลักตามลักษณะการทำงานดังนี้ (รูปที่ 3.3.1)



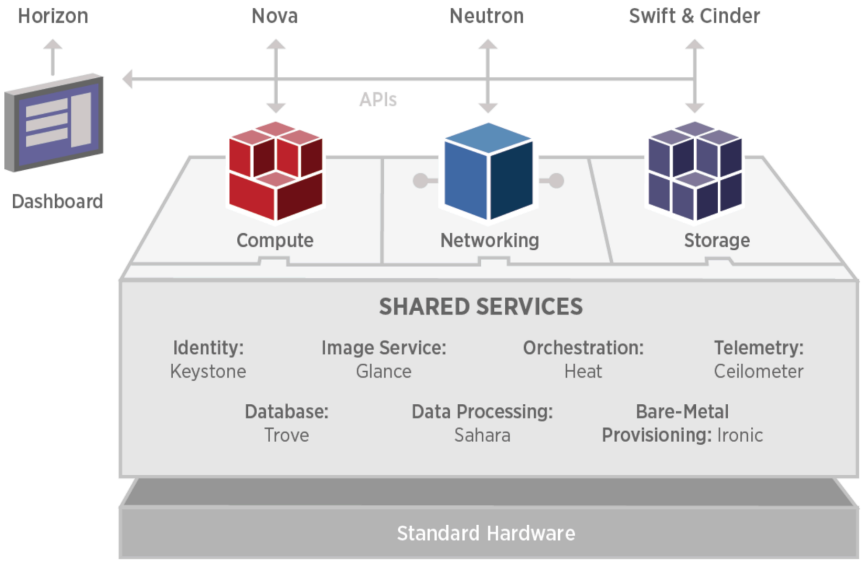

*รูปที่ 3.3.1: สถาปัตยกรรมการออกแบบระบบจำลองเครือข่ายเสมือนจริง (NFV)*

#### 3.3.1 เครื่องมือที่สนับสนุนชั้นโครงสร้างพื้นฐาน (NFV infrastructure layer)

หากอ้างอิงสถาปัตยกรรมการออกแบบระบบจำลองเครือข่ายเสมือนจริง (รูปที่ 3.3.1) จะพบว่า หน้าที่หลักของชั้นนี้ คือ การจัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรเครือข่ายต่างๆ ได้แก่ หน่วยประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูล และระบบเครือข่ายต่างๆ ที่จำเป็นต้องใช้สำหรับการจำลองเน็ตเวิร์คเซอร์วิส กล่าวอีกนัยหนึ่ง หน้าที่หลักของชั้นนี้ คือ (1) การจัดสรรทรัพยากรเครือข่ายตามการร้องขอ (2) การจัดเตรียมเครื่องเสมือนจริง (VMs) ให้พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ และ (3) ดำเนินการตรวจสอบและวิเคราะห์ประสิทธิภาพการใช้งานทรัพยากรเครือข่ายในแต่ละเน็ตเวิร์คเซอร์วิส เพื่อปรับเพิ่ม/ลดจำนวนทรัพยากรประมวลผลและเครือข่ายให้มีความเหมาะสมต่อจำนวนเวิร์คโหลดที่เข้ามาในช่วงเวลานั้นๆ โดยในปัจจุบันนี้ พบว่ามีหลายหน่วยงานที่ดำเนินการให้บริการโครงสร้างพื้นฐานระบบคลาวด์เพื่อสนับสนุนการทำงานในชั้นนี้ ได้แก่ Amazon OpenStack, Google Cloud Platform (GCP), Web Services (AWS), Microsoft Azure, และ VMware Cloud โดยจะทำการเปรียบเทียบคุณสมบัติและความสามารถต่างๆ เพื่อให้เกิดความเข้าใจเกี่ยวกับซอฟต์แวร์และพื้นฐานเทคโนโลยีที่สำคัญที่มีบทบาทในการสร้างและจัดการโครงสร้างพื้นฐานระบบคลาวด์ที่เป็นประโยชน์ต่อการออกแบบและพัฒนาระบบในขั้นถัดไป

### 3.3.1.1 OpenStack

OpenStack [38-39] เป็นแพลตฟอร์มที่เกิดจากความร่วมมือกันของ NASA และ Rackspace โดยเริ่มพัฒนาขึ้นในปี 2010 เพื่อเป็นการให้บริการโครงสร้างพื้นฐานระบบคลาวด์สนับสนุน Infrastructure as a Service (IaaS) ซึ่งเป็นซอฟต์แวร์ Open source (Apache License 2.0) ที่ใช้สำหรับบริหารจัดการเครือข่ายแบบ Cloud operating system และทำหน้าที่ให้บริการและจัดเตรียมทรัพยากรโครงสร้างพื้นฐานต่างๆ เช่น ด้านการประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูลและระบบเครือข่ายที่จำเป็น โดยมีการจัดการและควบคุมผ่านทาง Dashboard หรือ OpenStack API ซึ่งจะช่วยให้ผู้ดูแลระบบสามารถควบคุมและจัดการทรัพยากรได้ตามความต้องการของ รูปที่ 3.3.2 แสดงโครงสร้างโดยรวมของ OpenStack มีส่วนประกอบที่สำคัญหลักๆ ดังต่อไปนี้




*รูปที่ 3.3.2: แผนภาพรวมสถาปัตยกรรม OpenStack*

- Compute (NOVA): ทำหน้าที่ในการจัดเตรียมและสร้างเครื่องเสมือน (VMs) บนคลาด์เซิร์ฟเวอร์ที่เราต้องการ ซึ่งนักพัฒนาที่สร้างโปรแกรมประยุกต์บนคลาวด์สามารถเข้าถึงเครื่องเสมือนที่ได้ถูกจำลองขึ้นผ่านทาง CLI ส่วนผู้ดูแลระบบและผู้ใช้บริการสามารถเข้าถึงได้ผ่านทางเว็บ สำหรับ Hypervisor ที่ใช้งานและเป็นที่นิยม คือ KVM และ XenServer
- Networking (Neutron): โมดูลนี้มีความสำคัญในการสร้าง จัดการ และควบคุมระบบเครือข่ายพื้นฐานทั้งที่เป็น physical network และ virtual network บน OpenStack โดยผู้ใช้สามารถสร้างเครือข่ายเชิงพื้นฐาน เช่น private network หรือ shared network และกำหนดค่า IP และเครือข่ายได้ด้วยตนเอง เพื่อรองรับการเชื่อมต่อกับเครื่องเสมือนได้อย่างยืดหยุ่นและมีประสิทธิภาพ
- Storage (Swift & Cinder): เป็นแหล่งจัดเก็บข้อมูลที่มีการทำงานทั้งในรูปแบบ (1) object storage สำหรับสำรองและจัดเก็บข้อมูลขนาดใหญ่ เช่น images, photo, email, backups, และ archieves และ (2) block storage ซึ่งเป็นการทำงานร่วมกับโมดูล Nova สำหรับจัดเก็บทั้งแบบชั่วคราว ซึ่งข้อมูลเหล่านี้จะหายไปทันทีหากมีการหยุดทำงานของ instance และแบบถาวร ซึ่งข้อมูลเหล่านี้จะไม่สูญหายไป แม้ว่า instance กำลังทำงานอยู่หรือยุติการทำงานไปแล้วก็ตาม

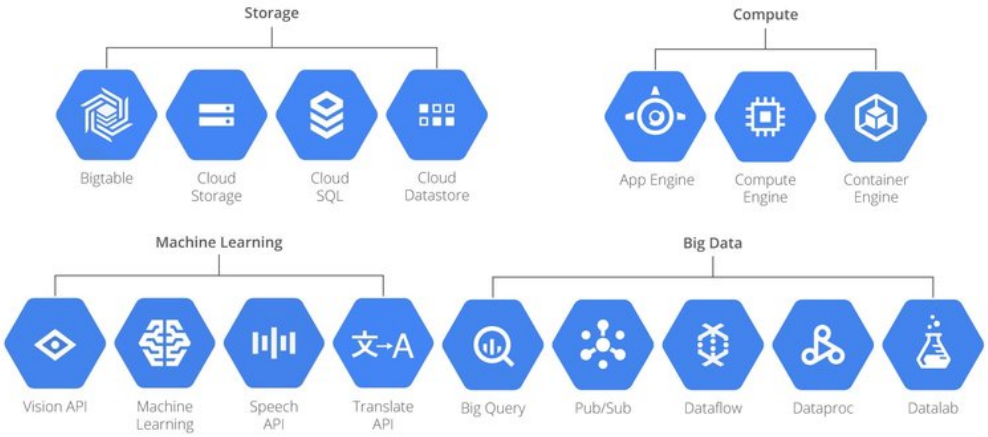
- Dashboard (Horizon): เป็น web-based interface ทำหน้าที่ในการควบคุมและจัดการเข้าถึงแต่ละ services ที่ทำงานอยู่ภายใน OpenStack โดยผู้ใช้บริการสามารถจัดการได้ด้วยตนเอง เช่น การสร้าง instance กำหนดพื้นที่จัดเก็บข้อมูล กำหนดหมายเลขไอพี ให้กับ instance ที่สร้างขึ้น เหล่านี้เป็นต้น
- Orchestration (Heat): จัดเป็น Template-driven engine ที่อนุญาตให้ผู้พัฒนาโปรแกรมสามารถสร้างและปรับแต่งการตั้งค่า template ของแต่ละ instance ได้ตามความต้องการ เพื่อรองรับการใช้งาน cloud applications ของผู้ใช้
- Identity (Keystone): ทำหน้าที่จัดการสิทธิ์ ตรวจสอบการเข้าถึง และอนุมัติการให้บริการใช้งานของแต่ละโมดูลเมื่อมีการเข้าถึงไฟล์หรือเครื่อง VMs บน OpenStack โดยจะดำเนินการตรวจสอบสิทธิ์การใช้งานเพื่อระบุตัวตนและความสามารถในเรียกใช้งานแต่ละโมดูล เพื่อเพิ่มประสิทธิภาพการรักษาความปลอดภัยบน OpenStack
- Image service (Glance): ทำหน้าที่จัดเก็บ ISO images ต่างๆ สำหรับรองรับการจำลองเครื่อง VMs บน OpenStack ในอนาคต
- Telemetry (Ceilometer): เป็นโมดูลที่ได้ถูกออกแบบมาเพื่อตรวจสอบและเก็บรวบรวมข้อมูลการใช้งานในด้านต่างๆ และวัดประสิทธิภาพการทำงานของทรัพยากรสำหรับการประเมินผล
- Database (Trove): มีเป้าหมายที่จะให้ผู้ใช้สามารถใช้งาน relational database และ non-relational database ได้อย่างรวดเร็วและง่ายดาย ไม่เกิดความยุ่งยากและซับซ้อนในการจัดการระบบฐานข้อมูล ซึ่งผู้ใช้และผู้ดูแลระบบสามารถที่จะจัดการและเรียกใช้ฐานข้อมูลหลาย ๆ ฐานข้อมูลพร้อมกัน ตามความต้องการใช้งาน

#### 3.3.1.2 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) [40-41] ถูกพัฒนาและให้บริการโดย Google โดยมีเป้าหมายเพื่อให้บริการโครงสร้างพื้นฐานบนคลาวด์ บริการเซิร์ฟเวอร์เสมือนบนคลาวด์ บริการจัดเก็บข้อมูลบนคลาวด์ ซึ่งคลอบคลุมทั้ง compute, storage, networking, machine Learning และ containers ผู้พัฒนาสามารถสร้างและพัฒนาแอปพลิเคชั่นได้อย่างยืดหยุ่น ซึ่งการให้บริการของ GCP สามารถแบ่งออกเป็น 4 ประเภทหลัก (แสดงดังรูปที่ 3.3.3)

- Compute: ทำหน้าที่ให้บริการทั้ง (1) compute engine (IaaS) ทั้งในรูปแบบ VM และServerless computing ที่ให้บริการเซิร์ฟเวอร์เสมือนบนคลาวด์ และ (2) app engine (PaaS) ที่อนุญาตให้นักพัฒนาสามารถสร้างแอปพลิเคชันและเว็บแอปพลิเคชันได้อย่างยืดหยุ่น
- Storage: สำหรับให้บริการจัดเก็บข้อมูลขนาดใหญ่บนคลาวด์ เช่น ไฟล์สื่อ ฐานข้อมูล และข้อมูลอื่นๆ รวมถึง Google Cloud SQL ที่ให้บริการฐานข้อมูล SQL บนคลาวด์ และ Google Cloud Bigtable
- Big data: มี machine learning ที่มาพร้อมกับเครื่องมือและบริการต่าง ๆ สำหรับการสร้าง และจำลองโมเดลเรียนรู้ เช่น BigQuery ที่ให้บริการการวิเคราะห์ข้อมูลแบบมหาศาล

- Machine learning: Google ได้นำเสนอโซลูชั่นและแอปพลิเคชั่นที่มีความเกี่ยวข้องกับ machine learning เพื่อช่วยให้ผู้พัฒนาสามารถสร้าง ออกแบบโมเดล และใช้อัลกอริทึม machine learning ได้อย่างง่ายได้




*รูปที่ 3.3.3: โมดูลต่างๆ บนแพลตฟอร์ม GCP*

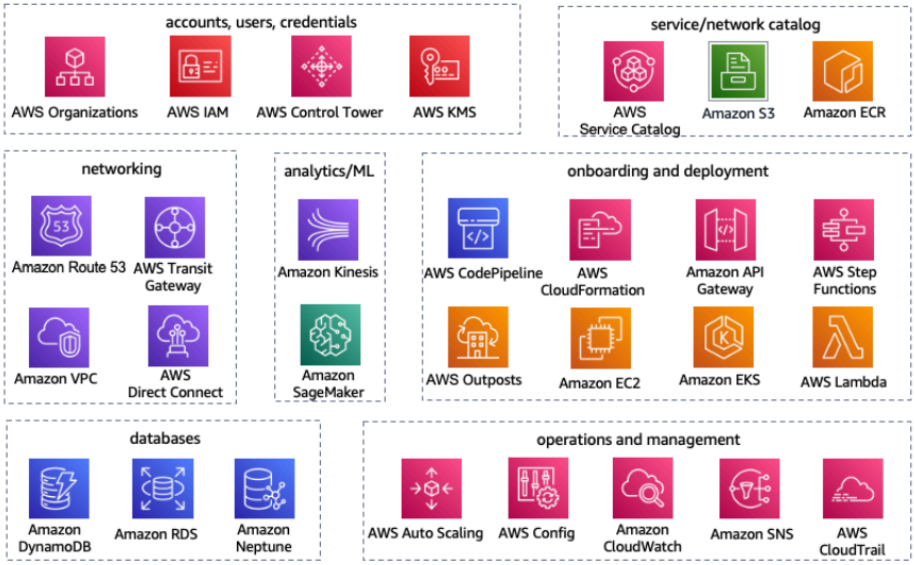
### 3.3.1.3 Amazon Web Services (AWS)

Amazon Web Services (AWS) [42-43] เป็นการให้บริการเชิงโครงสร้างพื้นฐานบนคลาวด์ (IaaS) ได้แก่ Amazon EC2, Amazon VPC, Amazon RDS และ Amazon S3 ฯลฯ และให้บริการเชิงแพลตฟอร์ม (PaaS) เช่น AWS Elastic Beanstalk, AWS Lambda,AWS App Runner และ AWS Amplify ฯลฯ โดยจะมีการคำนวณค่าใช้จ่ายตามการใช้งานจริง (Pay-as-you-go) พร้อมระบบ auto scaling ที่ช่วยปรับปรุงความจุของแต่ละ instance ตามเงื่อนไขที่ผู้ใช้กำหนด โดยการให้บริการของ AWS นั้นสามารถแบ่งกลุ่มการให้บริการตามฟังก์ชันการทำงาน (functional grouping) ได้ดังนี้

- Compute services: บริการที่เกี่ยวข้องกับการคำนวณและประมวลผล เช่น Elastic Compute Cloud (EC2), Lambda, Elastic Container Service , Elastic Kubernetes Service
- Storage services: บริการที่เกี่ยวข้องกับการจัดเก็บข้อมูล เช่น Simple Storage Service (S3), Elastic Block Store (EBS), Elastic File System (EFS), Relational Database Service (RDS), DynamoDB, Redshift, DocumentDB, Aurora
- Database services: บริการที่เกี่ยวข้องกับฐานข้อมูล เช่น RDS, DynamoDB, Redshift, DocumentDB, Aurora
- Networking services: บริการที่เกี่ยวข้องกับเครือข่าย เช่น Virtual Private Cloud (VPC), Route 53, CloudFront, API Gateway, Direct Connect, Global Accelerator
- Security services: บริการที่เกี่ยวข้องกับความปลอดภัย เช่น IAM, Key Management Service (KMS), Shield, Secrets Manager

- Management services: บริการที่เกี่ยวข้องกับการจัดการเซอร์วิส เช่น Organizations, Config, CloudWatch, CloudTrail, CodePipeline, CodeBuild, CodeDeploy, CodeCommit
- Analytics Services: บริการที่เกี่ยวข้องกับการวิเคราะห์ เช่น Redshift, QuickSight, Kinesis Data Analytics, Kinesis Data Firehose, Elasticsearch Service
- AI and Machine Learning services: บริการที่เกี่ยวข้องกับปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง เช่น SageMaker Autopilot, SageMaker AutoML

รูปที่ 3.3.4 แสดงตัวอย่างบางส่วนของเครื่องมือบน AWS ที่สนับสนุนการให้บริการด้าน cloud-native NFV orchestration service

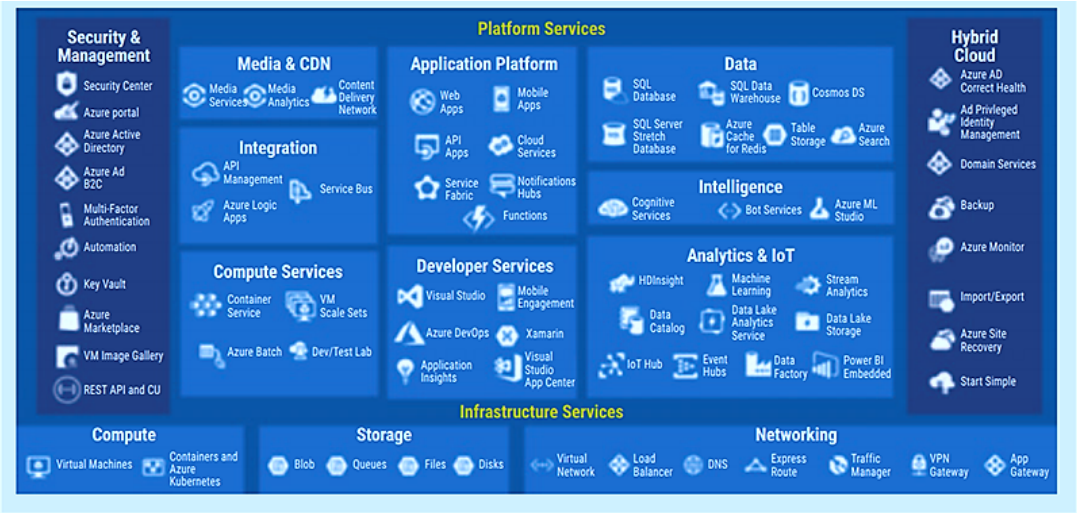



*รูปที่ 3.3.4: เครื่องมือของ AWS ที่สนับสนุนการให้บริการด้าน cloud-native NFV orchestration [36]*

### 3.3.1.4 Microsoft Azure

Microsoft Azure [44] เป็นหนึ่งในคลาวด์แพลตฟอร์มที่ได้รับความนิยมและใช้งานอย่างแพร่หลาย โดยมีวัตถุประสงค์เพื่อสร้างและพัฒนาโครงสร้างพื้นฐานคลาวด์สนับสนุนการให้บริการทั้งแบบ Infrastructure as a Service (IaaS), Platform as a Service (PaaS), และ SaaS (Software as a Service) และบริการเสริมด้านอื่นๆ ซึ่งสามารถแบ่งการให้บริการได้ออกเป็น 4 หมวดหมู่หลัก (แสดงดังรูปที่ 3.3.5)

- Infrastructure services: เป็นการให้บริการโครงสร้างพื้นฐานคลาวด์ ได้แก่ compute, storage และ networking
    - Compute: Azure มีบริการที่ช่วยในการสร้างและจัดการ VMs, containers, และแอปพลิเคชันต่างๆ บนคลาวด์ เช่น Azure virtual machines, Azure app service, และ Azure container instances




*รูปที่ 3.3.5: การให้บริการของ Microsoft Azure*

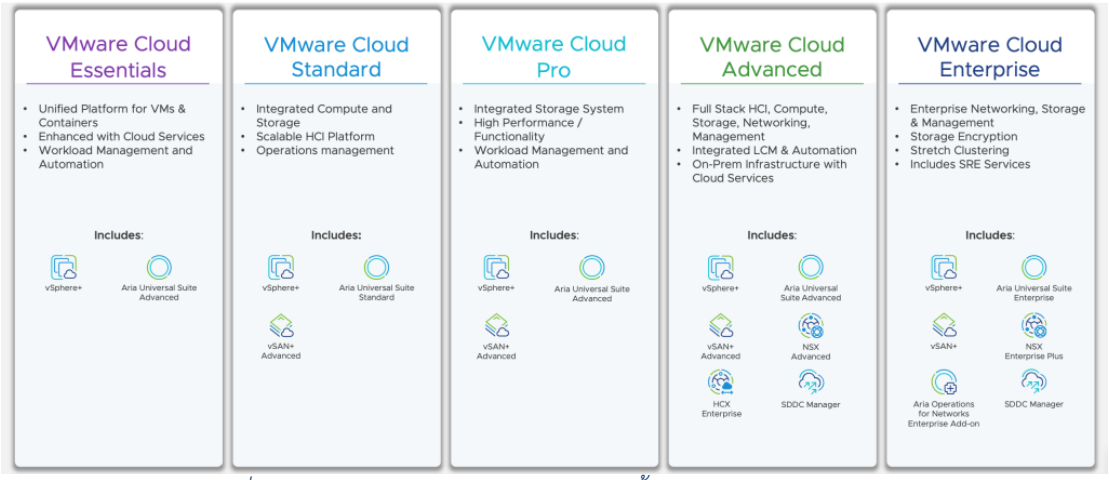
- Storage: สามารถจัดเก็บและจัดการข้อมูลในระบบคลาวด์ โดย Azure มีบริการต่างๆ เช่น Azure Blob Storage, Azure Files และ Azure Queue Storage
- Networking: บริการที่ให้ผู้ใช้สร้างและจัดการเครือข่ายในคลาวด์ได้ เช่น virtual network, load balancer, application gateway, VPN gateway

- Platform services: เป็นการจัดเตรียมโครงสร้างพื้นฐานคลาวด์เพื่องานเฉพาะทาง โดยมุ่งเน้นการให้บริการที่ช่วยให้นักพัฒนาสามารถพัฒนาซอฟต์แวร์หรือแอปพลิเคชันต่างๆ ได้อย่างสะดวก รวดเร็ว และปรับเปลี่ยนได้ตามความต้องการตามความเหมาะสม โดยเซอร์วิสที่มีให้บริการ เช่น
  - Media & CDN (Content Delivery Network): เป็นการให้บริการที่เกี่ยวกับสื่อและเน็ตเวิร์ตการจัดการข้อมูลผ่านเครือข่าย เช่น media services, media analytics, CDN ฯลฯ
  - Application platform: การให้บริการสำหรับการพัฒนาซอฟต์แวร์หรือแอปพลิเคชันต่างๆ บนคลาวด์ เช่น Web apps, Mobile Apps, API Apps ฯลฯ
  - Data: ให้บริการสำหรับการจัดเก็บและจัดการข้อมูลบนคลาวด์ เช่น SQL Server Stretch Database, Azure for Redis Cache, Table Storage, Azure Search ฯลฯ
  - Intelligence: การให้บริการสำหรับการวิเคราะห์ข้อมูลและสร้างโมเดลข้อมูลเชิงลึก เช่น Cognitive Services, Bot Services, Azure ML Studio ฯลฯ
  - Developer Services: การให้บริการเครื่องมือเฉพาะทางสำหรับอำนวยความสะดวกแก่นักพัฒนาระบบ เช่น Visual Studio, Mobile Engagement, Azure DevOps Xamarin ฯลฯ
  - Analytics & IoT: การให้บริการเครื่องมือสำหรับการวิเคราะห์ข้อมูลขนาดใหญ่และการวิเคราะห์งานด้าน IoT เช่นน Data catalog, Stream analytics, Power BI Embedded, ฯลฯ

- Security & Management: การให้บริการเครื่องมือสนับสนุนงานด้านความปลอดภัย ได้แก่ การพิสูจน์ตัวตนและการจัดการสิทธิ์ ความปลอดภัยบนระบบเครือข่าย เช่น Security Center, Azure portal, Multi-Factor Authentication, Key Vasht, ฯลฯ
- Hybrid Cloud: เป็นการบริการที่ช่วยให้ลูกค้าสามารถรวมการใช้งานระหว่าง public cloud และ private cloud เข้าด้วยกัน เพื่อให้ได้ประสิทธิภาพและความยืดหยุ่นในการดำเนินงานขององค์กร ประกอบด้วยเครื่องมือและบริการต่างๆ เช่น Azure Monitor, Import/Export, Azure site recovery และ Start Simple ฯลฯ

### 3.3.1.5 VMware Cloud

VMware [45] เป็นบริษัทซอฟต์แวร์ที่ให้บริการเทคโนโลยีเครือข่ายเสมือนและโครงสร้างพื้นฐานคลาวด์ให้กับธุรกิจทั่วโลก บริษัทนี้ก่อตั้งขึ้นในปี 1998 และเน้นการพัฒนาเทคโนโลยีเสมือนจริง (virtualization) เป็นหลัก โดยผลิตภัณฑ์หลักของ VMware ได้แก่ VMware vSphere, VMware vCenter, VMware NSX, และ VMware Cloud ซึ่งได้รับความนิยมและใช้งานกันอย่างแพร่หลาย มีให้บริการเป็นโครงสร้างพื้นฐานเทคโนโลยีคลาวด์แบบ multi-cloud ที่สามารถติดตั้งใช้งานเองได้ทั้งภายในองค์กร (on-premises) การใช้ประโยชน์จากการเช่าโครงสร้างพื้นฐานคลาวด์สาธารณะ (Public Clouds) หรือผ่านพันธมิตรของ VMware โดยวัตถุประสงค์ของ VMware คือการช่วยลดความยุ่งยากในการปรับใช้งานโดยการจัดเตรียมโครงสร้างพื้นฐานคลาวด์ที่เหมาะสมให้กับแต่ละองค์กร ผู้รับบริการสามารถใช้ประโยชน์จากเครื่องมือต่างๆ ที่มีให้บริการโดย VMware Cloud เพื่อใช้งานหรือรันแอปพิเคชันต่าง ๆ ขององค์กรได้อย่างปลอดภัย ตัวอย่างเช่น การกู้คืนระบบบนคลาวด์ในกรณีที่เกิดภัยพิบัติหรือการโจมตีด้วยแรนซัมแวร์ (ransomware) ทั้งนี้ VMware ได้แบ่งรูปแบบการให้บริการออกเป็น 5 ทางเลือกหลัก ได้แก่ Essentials, Standard, Pro, Advanced และ Enterprise (แสดงดังรูปที่ 3.3.6) โดยแต่ละรูปแบบจะมีคุณสมบัติความสามารถการให้บริการ และราคาการคิดค่าบริการที่แตกต่างกัน เพื่อตอบสนองความต้องการในการใช้งานโครงสร้างพื้นฐานคลาวด์ ตั้งแต่ระดับพื้นฐานไปจนถึงระดับองค์กรขนาดใหญ่ ขณะที่การคิดราคาในแต่ละรูปแบบจะขึ้นอยู่กับขนาดและความต้องการใช้งานขององค์กร โดยมีการคิดค่าบริการตามการใช้งานจริง (Pay-as-you-go)




***รูปที่ 3.3.6:*** *รูปแบบการให้บริการโครงสร้างพื้นฐานคลาวด์ของ VMware*

### 3.3.2 เครื่องมือที่สนับสนุนชั้นเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNF layer)

หน้าที่หลักในชั้นนี้ คือ (1) การจำลองเน็ตเวิร์คเซอร์วิสประเภทต่างๆ เช่น Firewall, IDS, DPI, ฯลฯ และจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส (VNF) ที่ได้ถูกจำลองขึ้นให้มีขีดความสามารถในการทำงานบนทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ได้กำหนดขึ้น (2) ดำเนินการตั้งค่าแต่ละเน็ตเวิร์คเซอร์วิสตามความต้องการใช้งานหรือตามนโยบาย (policy) ที่ผู้ดูแลระบบกำหนดไว้ (3) ทำการปรับลดหรือขยายเน็ตเวิร์คเซอร์วิส (scaling in/out VNFs) เมื่อประสิทธิภาพของทรัพยากรที่ใช้สำหรับการประมวลผลเกินกว่าค่า threshold ที่กำหนดไว้ (4) อัพเกรดซอฟต์แวร์ที่เกี่ยวข้อง และ (5) ถอดถอนการติดตั้งเน็ตเวิร์คเซอร์วิสเมื่อไม่มีการใช้งาน กล่าวอีกนัยหนึ่ง ชั้นนี้เปรียบเสมือนแหล่งรวมเน็ตเวิร์คเซอร์วิสแบบเสมือนประเภทต่างๆ ที่ต้องการจำลองขึ้นไว้ด้วยกัน โดยคลอบคลุมถึง (1) กลุ่มเน็ตเวิร์คเซอร์วิสที่มีความเกี่ยวข้องกับการรักษาความปลอดภัยบนเครือข่าย ได้แก่ firewall, IDS, DPI, ฯลฯ (2) กลุ่มเน็ตเวิร์คเซอร์วิสที่มุ่งเน้นการเพิ่มคุณภาพการให้บริการและการตอบสนองต่อการให้บริการเครือข่าย (QoS) ได้แก่ load balancer, WAN optimizer, web proxy, CDN, ฯลฯ และ (3) กลุ่มเน็ตเวิร์คเซอร์วิสที่เน้นการจัดเตรียมโครงสร้างพื้นฐานเครือข่ายเพื่อสนับสนุนเครือข่าย 5G ในอนาคต เช่น switch, packet gateway, HSS, MME ฯลฯ อย่างไรก็ตาม ในหัวข้อนี้จะมุ่งเน้นการนำเสนอรายละเอียดของเน็ตเวิร์คเซอร์วิสที่เกี่ยวข้องกับด้านความปลอดภัยเป็นหลัก ได้แก่ Intrusion Detection System (IDS), Deep Packet Inspection (DPI), Cyber traps, Security Information and Event Management (SIEM) และ Network traffic analyzer

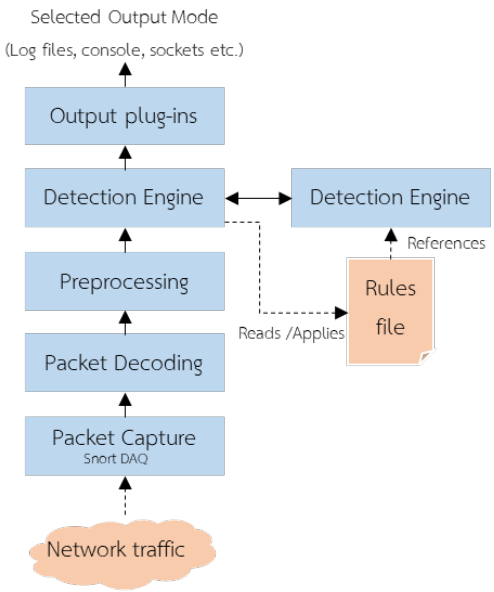
#### 3.3.2.1 Intrusion Detection System (IDS)

IDS ถือเป็นเครื่องมือที่สำคัญในการช่วยป้องกันเครือข่ายจากภัยคุกคามต่าง ๆ โดยจะทำหน้าที่เหมือนเสมือนระบบเฝ้าระวัง เพื่อคอยตรวจสอบกิจกรรมทั้งหมดที่เกิดขึ้นในระบบเครือข่าย หากพบกิจกรรมใดที่ผิดปกติหรือน่าสงสัย ระบบ IDS จะแจ้งเตือนผู้ดูแลระบบให้ทราบ เพื่อดำเนินการตรวจสอบและป้องกันเพิ่มเติม ซึ่ง IDS จะใช้ 2 วิธีหลักในการตรวจจับและวิเคราะห์ความผิดปกติ ได้แก่ (1) การตรวจจับเหตุการณ์ที่ผิดปกติ (anomaly detection) โดยอ้างอิงข้อมูลจากประวัติการใช้งานและใช้เกณฑ์ในการชี้วัดค่าทางสถิติ เพื่อเปรียบเทียบกับข้อมูลเหตุการณ์ปกติที่มีอยู่ หากกิจกรรมหรือเหตุการณ์ใดมีพฤติกรรมที่เบี่ยงเบนไปจากเดิม ให้ถือว่าเหตุการณ์เกิดขึ้นนั้นผิดปกติและถือว่าเป็นการโจมตี และ (2) การตรวจจับการบุกรุก (misuse detection) คือ การวิเคราะห์เหตุการณ์ที่เกิดขึ้นในระบบ โดยการพิจารณาจากเหตุการณ์หรือชุดของเหตุการณ์ที่ตรงกับรูปแบบเหตุการณ์ที่ได้กำหนดไว้แล้วว่าเป็นการบุกรุก หรือที่เรียกว่า Signature-based detection แนวทางนี้จะใช้ข้อมูลความรู้เกี่ยวกับพฤติกรรมที่เป็นการบุกรุก และค้นหาเพื่อตรวจจับพฤติกรรมเหล่านี้โดยตรง โดยหนึ่งในเครื่องมือการตรวจจับการบุกรุกที่เป็น open source ที่ได้รับความนิยมและมีการใช้งานกันอย่างแพร่หลาย คือ **Snort** [46-47] ซึ่งเป็นระบบตรวจจับการบุกรุกแบบระบุลักษณะ (signature-based IDS) การทำงานเป็นแบบ real-time และมีภาษาที่ใช้ในการกำหนดกฎ (rules) ในการตรวจสอบที่สามารถเข้าใจได้ง่ายและมีความยืดหยุ่นสูง รูปที่ 3.3.7 แสดงโครงสร้างของระบบ Snort ตรวจจับการบุกรุก ซึ่งประกอบด้วย 4 ส่วนหลักๆ ดังนี้

- Packet capture & decoding: เป็นส่วนที่ใช้ในการดักจับและถอดรหัสข้อมูลจากแพ็คเก็ตที่ถูกส่งผ่านเครือข่ายเพื่อทำการตรวจสอบความผิดปกติ ส่วนนี้จะเกี่ยวข้องกับการเชื่อมต่อกับอินเทอร์เฟซ

เครือข่ายเพื่อรับข้อมูลแพ็คเก็ตที่ไหลเข้า และทำการถอดรหัสข้อมูล (decoding) ซึ่งอาจมีการใช้งานเครื่องมือหรือไลบรารีต่างๆ เช่น libpcap, packet-capturing library, หรือ WinPcap เพื่อช่วยในกระบวนการจับและถอดรหัสแพ็คเก็ตข้อมูล ก่อนที่จะส่งแพ็กเก็ตที่ผ่านการถอดรหัสแล้วไปยังส่วนอื่นต่อไป

- Preprocessing: เป็นส่วนที่ใช้ในการประมวลผลและแก้ไขข้อมูลก่อนที่จะถูกส่งไปยังตัวตรวจจับ (detection engine) ส่วนนี้มีหน้าที่หลักคือการตรวจสอบและแก้ไขข้อผิดพลาดในการส่งข้อมูล (packet) เช่น การแก้ไขข้อผิดพลาดในการส่งข้อมูล TCP/IP, การตรวจจับและป้องกันการโจมตีแบบสแกน (port scanning) และการตรวจจับและป้องกันการโจมตีแบบส่งข้อมูลที่ผิดปกติ (anomaly-based attacks) โดยส่วนนี้จะมีการใช้งาน preprocessor plug-ins ต่างๆ เพื่อปรับปรุงและประมวลผลข้อมูลก่อนส่งต่อไปยัง detection engine
- Detection engine: เป็นส่วนที่ใช้ในการตรวจจับการบุกรุกในเครือข่าย โดยมีหน้าที่หลักคือการวิเคราะห์และตรวจสอบข้อมูลที่ผ่านมาว่ามีลักษณะของการโจมตีหรือไม่ โดยใช้กฎหรือลายเซ็น (rules/signatures) ที่กำหนดไว้ล่วงหน้าในการตรวจสอบ หากพบว่ามีการบุกรุกเกิดขึ้น ระบบจะทำการแจ้งเตือนไปยังผู้ที่เกี่ยวข้องต่อไป
- Output: ทำหน้าที่ในการแสดงผลลัพธ์ เก็บบันทึกข้อมูลการบุกรุก หลังจากการตรวจจับการบุกรุก โดยส่วนนี้จะทำหน้าที่เกี่ยวกับการบันทึกข้อมูลการโจมตี (logging) และการแจ้งเตือน (alerting) ซึ่งสามารถกำหนดรูปแบบและวิธีการแจ้งเตือนต่างๆ

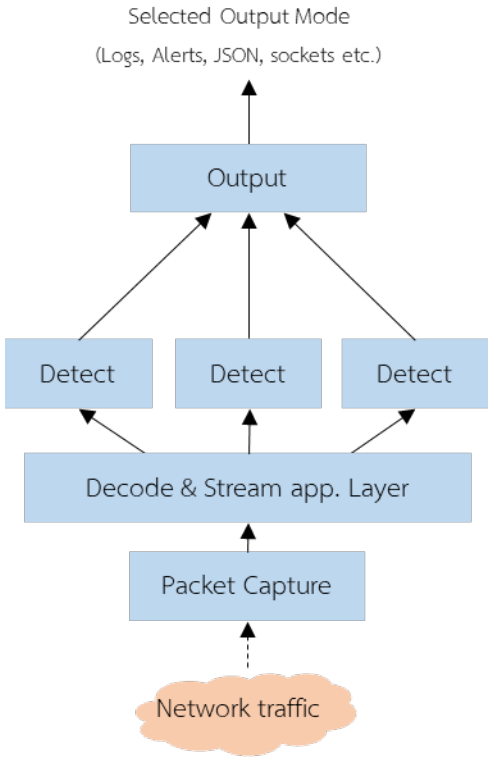



*รูปที่ 3.3.7: โครงสร้างของระบบ Snort*

### 3.4.2.2 Deep Packet Inspection (DPI)

DPI [48] เป็นเทคโนโลยีที่ใช้ในการวิเคราะห์และตรวจสอบแพ็คเก็ตข้อมูลบนเครือข่ายอินเทอร์เน็ตแบบละเอียด โดยการสแกนและวิเคราะห์ทั้งส่วนหัวของแพ็คเก็ต (packet header) และเนื้อหาข้อมูล (payload)

ทั้งหมดในแพ็คเก็ต โดยใช้เทคนิคการวิเคราะห์แบบ pattern matching เพื่อตรวจสอบเนื้อหาที่อยู่ภายในแพคเก็ต และใช้กฎ/รูปแบบที่กำหนดไว้ล่วงหน้าสำหรับอ้างอิงการตรวจสอบนี้ ซึ่งต่างจากการตรวจสอบโดย IDS ที่จะเน้นการตรวจสอบข้อมูลในส่วนหัวของแพ็คเก็ตเท่านั้น โดยไม่มีการวิเคราะห์เนื้อหาของแพ็คเก็ต ซึ่งข้อดีของ DPI จะช่วยให้ผู้ดูแลเครือข่ายสามารถระบุและแยกแยะการสื่อสารที่เกิดขึ้นในเครือข่ายได้ตามเนื้อหาที่อยู่ภายในแพคเก็ต โดยทั่วไปแล้ว DPI นิยมนำมาใช้ในการตรวจจับและป้องกันภัยคุกคามเครือข่าย เช่น การโจมตีแบบ DoS/DDoS, Man-in-the-middle และ Zero-day เป็นต้น รวมถึงสามารถใช้ในการควบคุมการใช้งานเครือข่ายตรวจจับและป้องกันการรั่วไหลของข้อมูล (data leakage) หรือการส่งข้อมูลที่ละเอียดอ่อนหรือเป็นความลับผ่านเครือข่าย เช่น การกำหนดนโยบาย QoS หรือการบล็อกเว็บไซต์ที่ไม่เหมาะสมหรือเสี่ยงต่อความปลอดภัย เป็นต้น โดยหนึ่งในเครื่องมือ open source ที่มีความสามารถในการทำ DPI และถูกนำมาใช้งานกันอย่างแพร่หลาย คือ **Suricata** [49-50] โดย Suricata นี้ถูกพัฒนาขึ้นโดย Open Information Security Foundation (OISF) มุ่งเน้นการตรวจจับและป้องกันการโจมตีในระดับเครือข่ายแบบ real-time โดยรองรับการตรวจจับแบบ signature-based และ anomaly-based ทำให้ Suricata สามารถตรวจจับการโจมตีที่รู้จักและไม่รู้จักโดยการวิเคราะห์แพ็กเก็ตข้อมูลอย่างละเอียดในชั้นของข้อมูลต่างๆ ตั้งแต่ชั้น data link ไปจนถึงชั้น application layer รูปที่ 3.3.8 โครงสร้างสถาปัตยกรรมของ Suricata ซึ่งมีรูปแบบการทำงานแบบมัลติเธรด (multithread) ร่วมกับหน่วยประมวลผลแบบหลายแกน (multi core) และหลายหน่วยประมวลผล (muti processor) ซึ่งจะช่วยเพิ่มประสิทธิภาพในการประมวลผลข้อมูลพร้อมกันได้หลายๆ งาน ทำให้สามารถวิเคราะห์และตรวจสอบข้อมูลได้อย่างรวดเร็ว โดยการทำงานจะแบ่งการประมวลผลข้อมูลออกเป็นหลายๆ ส่วน แต่ละส่วนจะทำงานพร้อมกันในกระบวนการต่าง ๆ โดยไม่ต้องรอให้กระบวนการก่อนหน้าทำงานเสร็จสิ้นก่อน ซึ่งทำให้สามารถประมวลผลแพ็คเก็ตที่เข้าสู่ระบบได้




*รูปที่ 3.3.8: โครงสร้างสถาปัตยกรรมของ Suricata*

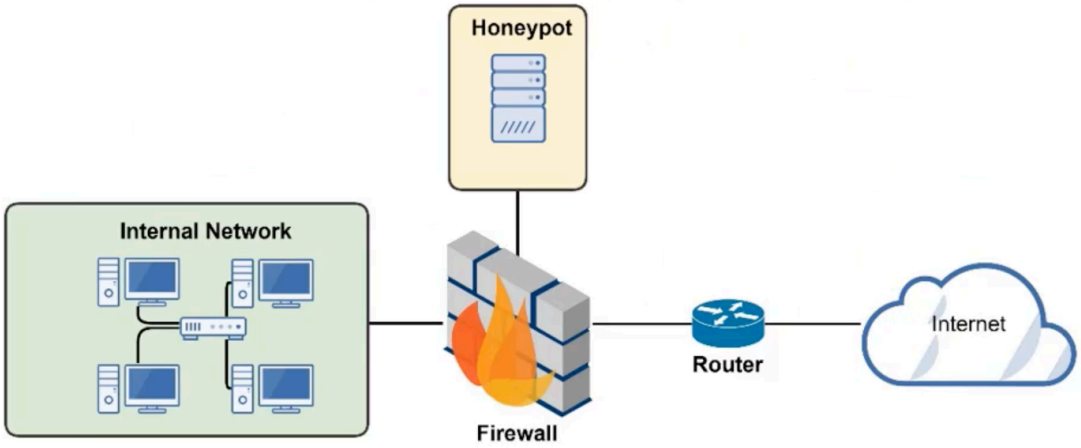
- Packet capture: ทำหน้าที่ในการรับส่งและบันทึกแพ็คเก็ตที่ถูกส่งมาผ่านอินเทอร์เฟซเครือข่ายต่าง ๆ เพื่อนำมาประมวลผล
- Decode and stream application: ทำหน้าที่ถอดข้อมูลในแพ็คเก็ตและรวมข้อมูลแพ็คเก็ตเข้าด้วยกันเพื่อเตรียมการวิเคราะห์
- Detection: ขั้นตอนนี้จะดำเนินการตรวนสอบและวิเคราะห์ข้อมูลในแพ็คเก็ตเพื่อหาความผิดปกติตามกฎที่ได้กำหนดไว้ หากพบว่ามีการกระทำที่ผิดปกติตามกฎที่กำหนดไว้ ระบบจะดำเนินการแจ้งเตือนต่อผู้ที่เกี่ยวข้องต่อไป หรือดำเนินการจัดการกับเหตุการณ์ดังกล่าวฯ ตามที่กฎที่ได้ระบุไว้
- Outputs: ทำหน้าที่แจ้งเตือนผลการตรวจจับให้ผู้ดูแลระบบทราบในรูปแบบต่างๆ เช่น การสร้างไฟล์บันทึก (log files) เป็นต้น

### 3.4.2.3 Cyber traps

Honeypot [51-52] จัดว่าเป็นเครื่องมือ open source ที่ถูกออกแบบมาเพื่อดึงดูดและดักจับผู้โจมตีที่พยายามเข้าถึงระบบหรือเครือข่ายคอมพิวเตอร์ โดยทำหน้าที่เสมือนเหยื่อล่อ (cyber traps) ที่ดึงดูดผู้ไม่ประสงค์ดีให้เข้ามาติดกับ เพื่อติดตามและศึกษาวิธีการโจมตีที่อาจจะเกิดขึ้น ซึ่งคุณสมบัติที่สำคัญของ Honeypot มีสาระสำคัญดังนี้ (1) ทำหน้าที่จำลองช่องโหว่หรือความเสี่ยงที่อาจเป็นไปได้ (simulation of vulnerabilities) เช่น เปิดพอร์ตที่มีช่องโหว่ร้ายแรง หรือสร้างระบบเสมือนที่ดูเหมือนจะมีข้อมูลอยู่ที่น่าสนใจเพื่อดึงดูดผู้โจมตีมาใช้ช่องโหว่นั้น ๆ (2) ทำหน้าที่ตรวจจับการทำงานของผู้โจมตี (detection) โดยบันทึก logs การกระทำที่เกิดขึ้น เช่น การสแกนพอร์ต การพยายามเข้าสู่ระบบ หรือการดาวน์โหลดข้อมูลที่ไม่ปกติ (3) ทำหน้าที่เป็นเหยื่อล่อ (diversionary tactics) โดยทำตัวเป็นจุดอ่อนเพื่อดึงดูดการโจมตี และเบี่ยงเบนความสนใจให้ผู้ไม่พึงประสงค์ดีพุ่งเป้าการโจมตีมาที่ Honeypot แทน โดยการทำให้เข้าใจว่าเครื่องเซิร์ฟเวอร์จริงของระบบมีการปกป้องการรักษาความปลอดภัยอย่างเต็มที่ (4) ทำหน้าที่เก็บข้อมูลที่สำคัญเกี่ยวกับการโจมตี (data collection and analysis) เช่น รูปแบบการโจมตีที่พบบ่อย หรือเทคนิคการบุกรุกที่มักจะถูกนำมาใช้ และ (5) ทำหน้าที่เก็บรวบรวมข้อมูลเพื่อการวินิจฉัย (research and forensics) สำหรับนำไปใช้ในการศึกษาวิจัยและพัฒนาความปลอดภัย วิเคราะห์แนวโน้มของการโจมตี หรือพัฒนาเทคนิคการป้องกันและตรวจจับการโจมตีต่าง ๆ ต่อไป รูปที่ 3.3.9 แสดงตัวอย่างตำแหน่งการติดตั้ง Honeypot ในระบบเครือข่าย ซึ่งโดยทั่วไปแล้วการจำลอง Honeypot นั้นสามารถทำได้ 2 วิธี ดังต่อไปนี้

- Physical Honeypots: เป็นเครื่องมือที่จำลองระบบหรือเครือข่ายจริง โดยใช้ฮาร์ดแวร์ที่แยกออกมาเป็นระบบเสมือน ซึ่งสามารถติดตั้งและทำงานเหมือนเครื่องจริงได้ ฮาร์ดแวร์ที่ใช้สามารถเป็นเครื่องเซิร์ฟเวอร์หรืออุปกรณ์เครือข่ายอื่น ๆ ที่มักจะถูกโจมตี ซึ่งระบบจะมีความเสถียรและความเชื่อถือได้สูง แต่ต้องใช้ทรัพยากรฮาร์ดแวร์จริงเพื่อทำงาน

- Virtual Honeypots: ระบบที่ถูกจำลองขึ้นมาบนแพลตฟอร์มเสมือน เช่น เซิร์ฟเวอร์ที่ทำงานบนเครื่องเสมือนหรือคอนเทนเนอร์ ซึ่งสามารถทำงานบนระบบปฏิบัติการเสมือนได้ เครื่องเสมือนที่สร้างขึ้นมาจะมีลักษณะและการทำงานเหมือนกับระบบจริง แต่ไม่ต้องใช้ทรัพยากรฮาร์ดแวร์จริง ทำให้มีความยืดหยุ่นและสะดวกในการดูแลและปรับแก้ให้เหมาะกับการใช้งานได้ง่าย




*รูปที่ 3.3.9: ตัวอย่างการจำลองและติดตั้ง Honeypot ในระบบเครือข่าย*

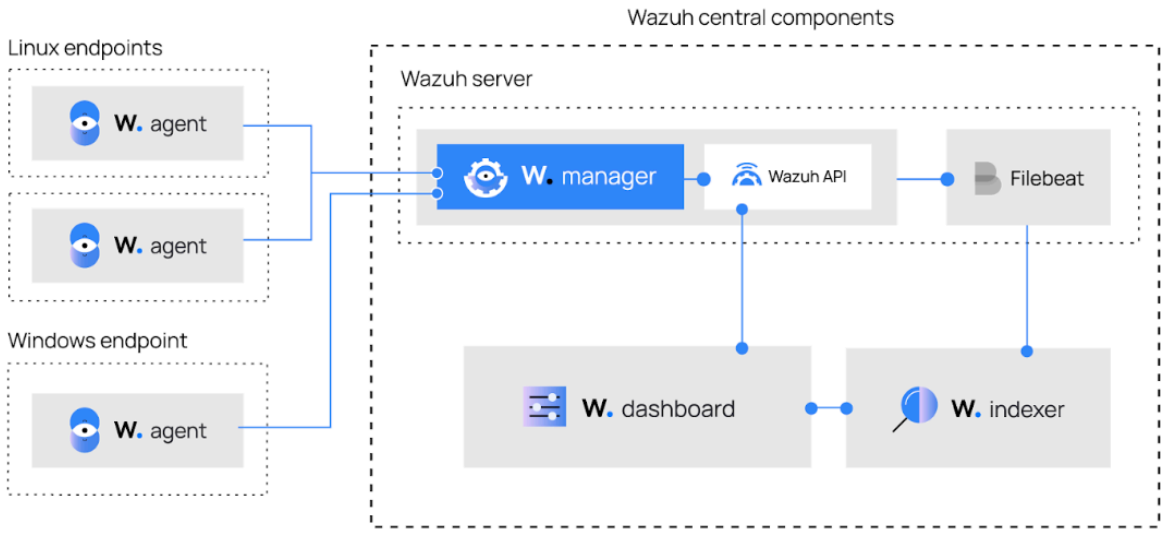
### 3.3.2.4 Security Information and Event Management (SIEM)

SIME [53] ถือว่าเป็นเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยอีกประเภทหนึ่ง ซึ่งเป็นระบบที่ใช้ในการจัดการและวิเคราะห์ข้อมูลด้านความปลอดภัยของระบบเครือข่าย โดยระบบจะทำการรวบรวมข้อมูลต่างๆ ที่มาจาก logs แบบ real-time แล้วนำมาวิเคราะห์เพื่อหาแนวโน้มหรือเหตุการณ์ที่เกี่ยวข้องกับความปลอดภัย เช่น การโจมตีจากผู้ไม่ประสงค์ดี การเข้าถึงที่ไม่ได้รับอนุญาต หรือการกระทำที่เป็นอันตรายต่อระบบ อีกทั้ง มีหน้าที่รวบรวมข้อมูลจากหลายแหล่ง เช่น OS server, application server, database server, network server รวมถึงอุปกรณ์เครือข่ายต่าง ๆ เช่น routers, switches, firewall เป็นต้น เพื่อนำมาวิเคราะห์ตรวจหาภัยคุกคามต้นเหตุที่เกิดขึ้นตามเงื่อนไขที่ผู้ดูแลระบบได้ทำการตั้งค่ากำหนดไว้ล่วงหน้า (pre-defined security poliicy) และการจัดทำรายงานอัตโนมัติ ทำให้ผู้ดูแลระบบด้านความปลอดภัยสามารถป้องกันและตอบสนองต่อภัยคุกคามต่างๆ ได้อย่างทันถ่วงที หนึ่งในตัวอย่างเครื่องมือที่ทำด้าน SIEM ที่เป็น open source คือ **Wazuh** [54] ซึ่งมีโมดูลการรักษาความปลอดภัยและการป้องกันที่หลากหลาย รูปที่ 3.3.10 แสดงสถาปัตยกรรมของ Wazuh ที่มีโครงสร้างการออกแบบโดยใช้หลักการแบบ agent-server โดย agents จะทำหน้าที่เป็นตัวรับส่งข้อมูลเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ และ server จะคอยรับข้อมูลที่ส่งมาจาก agents และดำเนินการถอดรหัส วิเคราะห์ และประมวลผลข้อมูลที่ได้รับมา

- Wazuh agents: เป็นโปรแกรมที่ออกแบบมาเพื่อใช้งานบนหลายแพลตฟอร์ม เช่น Linux, Windows, macOS, Solaris, AIX และอื่นๆ เพื่อรักษาความปลอดภัยของระบบคอมพิวเตอร์ ซึ่งมีบทบาทสำคัญในการเก็บรวบรวมบันทึกและเหตุการณ์จากแหล่งต่างๆ เช่น แอปพลิเคชัน เซิร์ฟเวอร์ หรืออุปกรณ์อื่นๆ และส่งข้อมูลเหล่านี้ต่อไปยัง Wazuh server เพื่อทำการวิเคราะห์และการตรวจสอบความเสี่ยงที่

อาจจะเกิดขึ้น เช่น การเปลี่ยนแปลงไฟล์ที่สำคัญหรือการปรับเปลี่ยนการตั้งค่าของระบบ ดังนั้น หากพบเหตุการณ์ที่เกิดความเสี่ยง ระบบสามารถทำการปิดกั้นเพื่อป้องกันการบุกรุกที่เป็นไปได้

- Wazuh server: ทำหน้าที่คอยรับข้อมูลจาก agents และดำเนินการถอดรหัส วิเคราะห์ และประมวลผลข้อมูลที่ได้รับมาตามเงื่อนไขหรือกฎการตรวจสอบการโจมตีต่างๆ (rules) ที่ได้กำหนดไว้ หากพบเหตุการณ์ที่สอดคล้องกับกฎ ระบบจะดำเนินการแจ้งเตือนว่าเป็นการโจมตี และส่งให้ Wazuh indexer เก็บข้อมูลรายละเอียดข้อมูลนี้ไว้ต่อไป
- Wazuh Indexer: เป็นส่วนที่มีหลายๆ โหนดทำงานร่วมกันในการจัดการข้อมูลที่มาจาก server โดยเก็บข้อมูลและดำเนินการในการเขียนและอ่านข้อมูลลงใน indexer ในรูปแบบเอกสาร JSON ซึ่งแต่ละเอกสารประกอบด้วยชุดของคีย์ (ชื่อฟิลด์หรือคุณสมบัติ) และค่าที่เกี่ยวข้อง ซึ่งอาจเป็นข้อความ ตัวเลข บูลีน วันที่ อาร์เรย์ของค่า ตำแหน่งทางภูมิศาสตร์ หรือข้อมูลประเภทอื่น ๆ เพื่อให้สามารถเรียกดูข้อมูลที่เกี่ยวข้องได้ง่าย และรวดเร็ว
- Wazuh dashboard: ทำหน้าที่แสดงข้อมูลด้านความปลอดภัย และเหตุการณ์ด้านความปลอดภัยแบบเว็บแอปพลิเคชันและเป็นเครื่องมือที่ช่วยให้ผู้ใช้งานสามารถวิเคราะห์ข้อมูลได้อย่างสะดวก แบ่งเป็นด้านต่าง ๆ เช่น การจัดการข้อมูลด้านความปลอดภัย ช่องโหว่ที่พบ แนวทางการปฏิบัติตามกฎระเบียบต่าง ๆ ข้อมูลการตรวจสอบความถูกต้องของไฟล์ อีกทั้งความสามารถในการสร้างรายงานและรูปแบบการแสดงผลบนแดชบอร์ดที่กำหนดเองได้

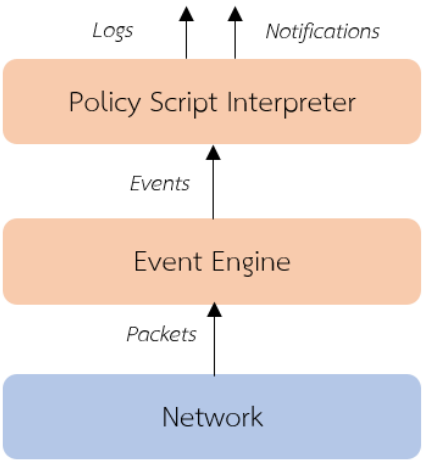



*รูปที่ 3.3.10: โครงสร้างสถาปัตยกรรมของ Wazuh*

### 3.3.2.5 Network Traffic Analyzer

Network Traffic Analyzer เป็นเครื่องมือที่ใช้วิเคราะห์ข้อมูลการจราจรในเครือข่าย เพื่อดูข้อมูลที่ผ่านเครือข่ายในรูปแบบต่างๆ เช่น ปริมาณการใช้งานเครือข่าย แบบ real-time การตรวจสอบแพ็คเก็ตข้อมูลเพื่อค้นหาปัญหาในเครือข่าย เช่น การโจมตี การตรวจจับแพ็คเก็ตที่เข้ามาในระบบเครือข่าย การตรวจจับการเชื่อมต่อที่ไม่

ปกติ การตรวจสอบประสิทธิภาพการทำงานของเว็บไซต์ หรือการวิเคราะห์แพ็คเก็ตข้อมูล (packet payloads) โดยเครื่องมือ open source ที่นิยมนำมาใช้สำหรับตรวจสอบข้อมูลหรือกิจกรรมต่างๆ ที่เกิดขึ้นในเครือข่าย คือ Zeek [55-56] โดย Zeek จะทำการสร้าง logs เพื่อระบุข้อมูลการทำงานของเครือข่ายอย่างละเอียด เช่น HTTP sessions, การร้องขอ DNS, SSL certificates, SMTP sessions เป็นต้น ซึ่ง logs สามารถเขียนได้ทั้งรูปแบบของ tab-separated และ JSON file formats ซึ่งเหมาะสำหรับการประมวลผลเพิ่มเติมด้วยเครื่องมืออื่น ๆ ทั้งนี้การเข้าถึง logs สามารถอ่านได้ 2 แบบ คือ (1) การอ่านไฟล์ข้อมูลได้จาก ไฟล์ PCAP และ (2) การอ่านข้อมูลแบบ real-time จาก Interface โดยโครงสร้างสถาปัตยกรรมของ Zeekประกอบไปด้วย 2 ส่วนหลัก (แสดงดังรูปที่ 3.3.11)

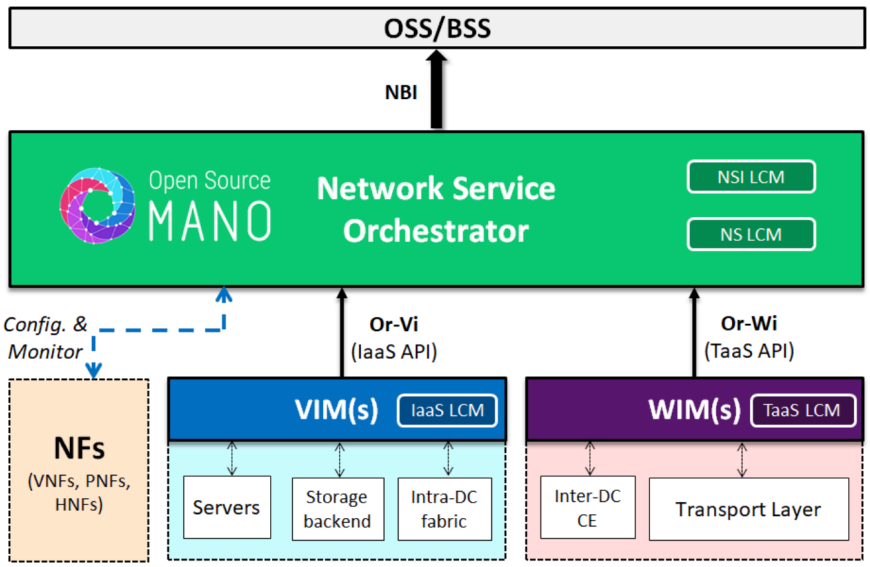



***รูปที่ 3.3.11:*** *โครงสร้างสถาปัตยกรรมของ Zeek*

- Event engine: ทำหน้าทที่ประมวลผลเพื่อสร้าง events ตามเหตุการณ์ที่เกิดขึ้นในระบบเครือข่าย โดยจะใช้กฎการตรวจจับที่ถูกกำหนดไว้เพื่อตรวจสอบข้อมูลและสร้าง events ตามเงื่อนไขที่กำหนด ตัวอย่างเช่น การตรวจจับการโจมตีด้วยช่องโหว่ในระบบ โดยเหตุการณ์ที่ถูกสร้างขึ้นจะถูกส่งต่อไปยัง Policy script interpreter
- Policy script interpreter: เป็นส่วนที่รับ events ที่ถูกสร้างขึ้นจาก event engine และทำการประมวลผลตามกฎและนโยบายที่ถูกกำหนดไว้ โดยใช้ภาษาสคริปต์แบบ Turing-complete ที่เขียนขึ้นเองหรือภาษา Zeek script เพื่อตรวจสอบเหตุการณ์ที่เกิดขึ้นและดำเนินการตามนโยบายที่กำหนด เช่น การบล็อกการเชื่อมต่อที่ไม่ปลอดภัย การส่งการแจ้งเตือนเมื่อเกิดเหตุการณ์ที่สำคัญ เป็นต้น ผลลัพธ์ที่ได้จากการประมวลผลจะถูกส่งออกเป็น logs และ notifications

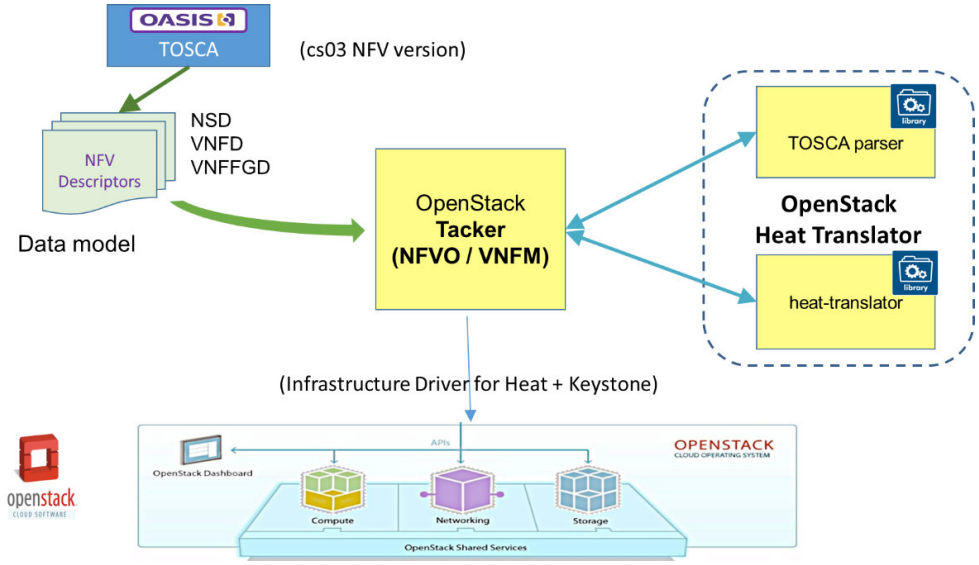
### 3.3.3 เครื่องมือที่สนับสนุนชั้นบริหารและจัดการ (MANO layer)

โดยทั่วไป ชั้นบริหารและจัดการเน็ตเวิร์คเซอร์วิส (management and orchestration layer) ทำหน้าที่ในการควบคุมลำดับการทำงานภายในระบบ และประสานให้การทำงานเป็นไปอย่างอัตโนมัติสอดคล้องกับบริบทความต้องการใช้งานหรือนโยบายที่กำหนดไว้ โดยหน้าที่หลักๆ ของชั้นนี้ คือ (1) บริหารและจัดการเน็ตเวิร์คโทโพโลยีในภาพรวม (2) ตรวจสอบสิทธิการเข้าถึงทรัพยากรเครือข่ายเมื่อมีการร้องขอ (3) ประสานงานกับโมดูลต่างๆ ที่เกี่ยวข้องทั้งในชั้น infrastructure layer เพื่อจัดสรรทรัพยากรเครือข่ายให้พร้อมสำหรับการใช้งาน และชั้น VNF layer เพื่อจำลองเน็ตเวิร์คเซอร์วิสตามการร้องขอ และ (4) บริหารและจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น โดยเครื่องมือ open source ที่ทำหน้าที่บริหารและจัดการเน็ตเวิร์คเซอร์วิส และได้ถูกนำมาใช้งานอย่างแพร่หลาย คือ Open source MANO (OSM) [28-30] และ OpenStack Tacker [57-58] ซึ่งทั้งสองเครื่องมือต่างมีเป้าหมายหลักที่เหมือนกัน คือ การพัฒนาแพลตฟอร์มสนับสนุนการบริหารและจัดการเน็ตเวิร์คเซอร์วิสที่สามารถรองรับมาตราฐานด้านโทรคมนาคมยุโรป (European Telecommunications Standards Institute: ETSI NFV) ได้ โดย OSM เป็นโครงการที่ริเริ่มโดยชุมชน ETSI และกลุ่มนักพัฒนาที่ร่วมกันสร้างขึ้นมา โดยมีจุดมุ่งหมายเพื่อออกแบบและพัฒนาโมดูลต่างๆ ที่เกี่ยวข้อง สนับสนุนการบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง รูปที่ 3.3.12 แสดงการออกแบบเชิงสถาปัตยกรรมระบบเชิงภาพรวม โดย OSM จะทำหน้าที่เป็น Network Service Orchestrator (NSO) ซึ่งเปรียบเสมือนโมดูลหลักสำคัญในการบริหารและจัดการเครือข่ายและเน็ตเวิร์คเซอร์วิส ดูแลและรับผิดชอบให้การทำงานภายในระบบเป็นไปอย่างราบรื่น อีกทั้งคอยประสานกับโมดูลอื่นๆ ที่เกี่ยวข้อง ได้แก่ (1) Virtual Infrastructure Manager (VIM) เพื่อจัดเตรียมทรัพยากรพื้นฐานที่จำเป็นต่อการจำลองเน็ตเวิร์คเซอร์วิสตามการร้องขอของผู้ใช้ (2) Network Function (NFs) เพื่อจำลองเน็ตเวิร์คเซอร์วิส ดำเนินการตั้งค่าต่างๆ ไปจนถึงการดำเนินการและติดตามรายวัน และ (3) WAN Infrastructure Manager (WIM) เพื่อสร้างการเชื่อมต่อเครือข่าย จัดสรรเน็ตเวิร์คแบนด์วิดท์ที่เหมาะสม และบังคับใช้นโยบายความปลอดภัย (security policy) ตามที่ได้ระบุไว้




*รูปที่ 3.3.12: โครงสร้างสถาปัตยกรรม OSM เชิงภาพรวม*

ในขณะที่ OpenStack Tacker ถูกพัฒนาโดยชุมชน OpenStack ที่มุ่งเน้นการทำ NFV Orchestrator และ VNF Manager สนับสนุนการจำลอง และบริหารจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงที่สามารถทำงานได้บนโครงสร้างพื้นฐานคลาวด์ ดังเช่น OpenStack โดยการออกแบบสถาปัตยกรรมของระบบนั้น (รูปที 3.3.13) เน้นการขับเคลื่อนด้วยข้อมูล (data model driven based approach) ผู้ใช้สามารถสร้างและออกแบบเน็ตเวิร์คเซอร์วิส รวมถึงการกำหนดสเปคที่ต้องการได้ด้วยตัวเองผ่านการเขียนข้อมูลลงเทมเพลตโดยใช้ OASIS TOSCA NFV standards ดังเช่น VNF descriptors, NS descriptors, VNFFG descriptors ฯลฯ ซึ่งเทมเพลตดังกล่าวฯ จะถูกนำมาใช้ในการอ้างอิงโดย OpenStack Tacker เพื่อการจำลองเน็ตเวิร์คเซอร์วิสต่อไป

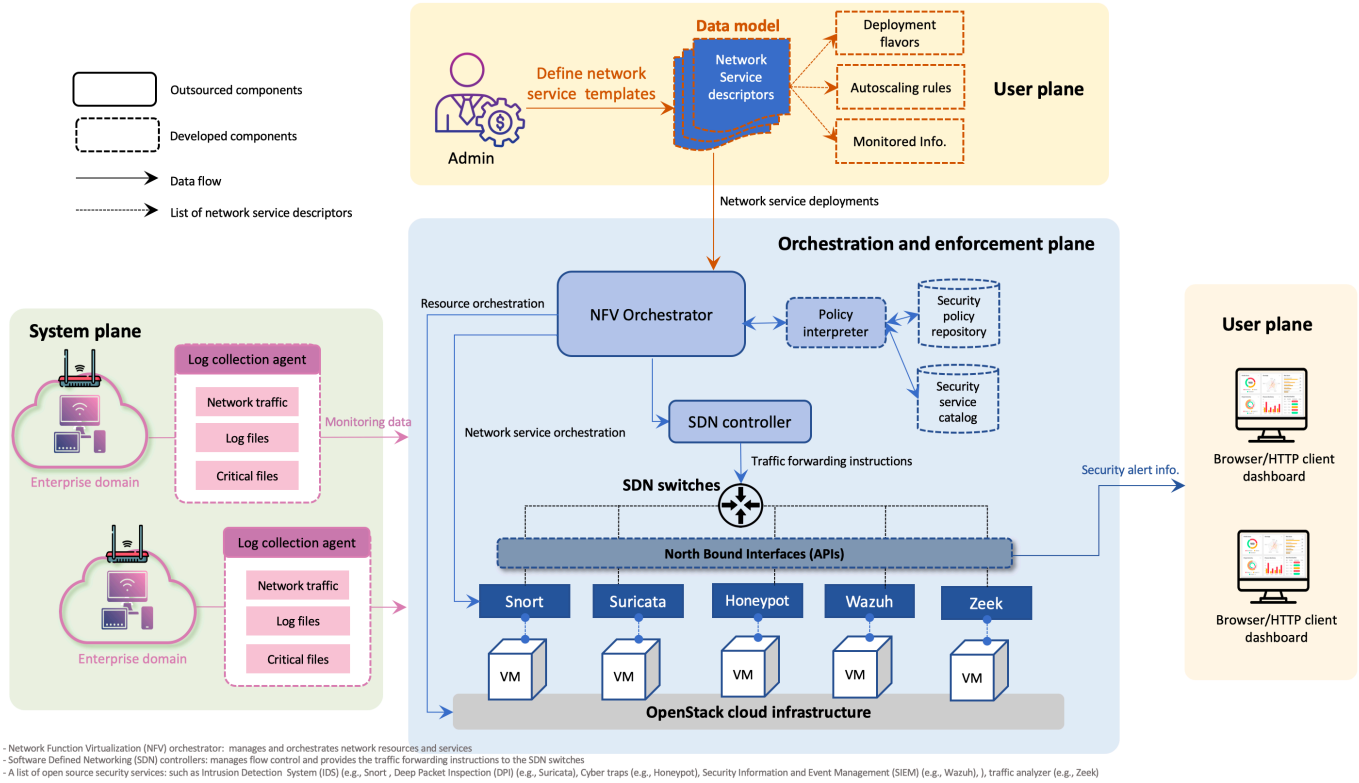



*รูปที 3.3.13: โครงสร้างสถาปัตยกรรมของ OpenStack Tacker*

### 3.4 วิจัย พัฒนา และออกแบบสถาปัตยกรรมแพลตฟอร์มเชิงภาพรวม

เพื่อให้สอดคล้องกับวัตถุประสงค์ของโครงการฯ ที่จะพลักดันในเกิดการสร้างและพัฒนางานวิจัยด้วยการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ดังเช่น NFV, SDN, และ cloud ดังนั้น โครงการนี้จึงมุ่งเป้าที่จะพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง เพื่อให้มีขีดความสามารถในการจัดสรรทรัพยากรเครือข่ายได้อย่างยืดหยุ่น และให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ[2] เพื่อให้องค์กรหรือหน่วยงานที่เกี่ยวข้องสามารถใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสที่ได้จัดเตรียมไว้ให้บริการ สำหรับใช้ในการป้องกันและรักษาความปลอดภัยบนระบบเครือข่ายขององค์กร โดยผู้ใช้ไม่จำเป็นต้องเสียเวลาในการติดตั้ง มีความรู้ทางเทคนิค หรือลงทุนในการจัดซื้อฮาร์ดแวร์แต่อย่างใด รูปที่ 3.4.1 แสดงการออกแบบสถาปัตยกรรมการโดยภาพรวม ซึ่งแบ่งการทำงานออกเป็น 3 ระดับชั้น ดังต่อไปนี้

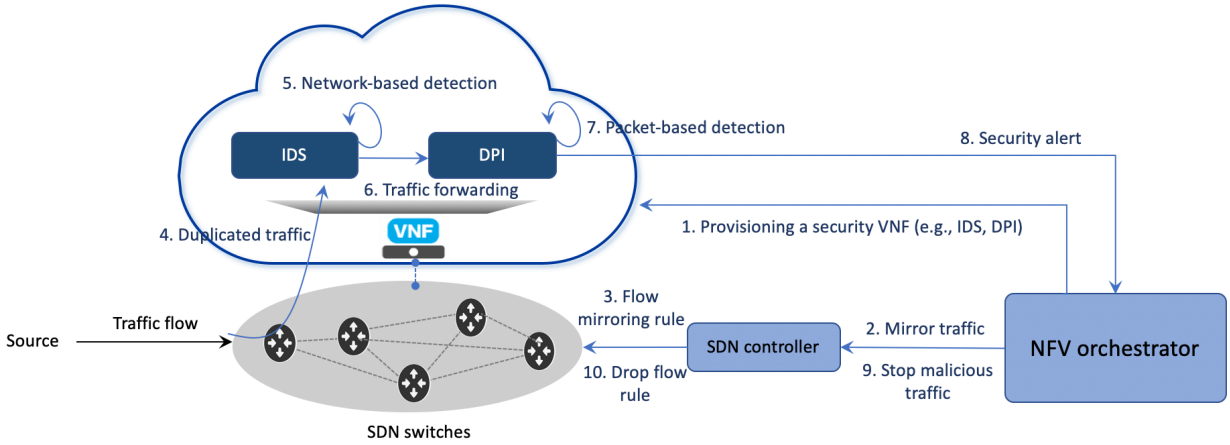
[2] สำหรับในระยะเริ่มต้นของการพัฒนาโครงการ (ระยะที่ 1: Proof of Concept: PoC) ทีมวิจัยมุ่งเน้นการจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสเสมือนจริงที่เกี่ยวข้องกับด้านการรักษาความปลอดภัยในเครือข่าย เช่น IDS, DPI, cyber traps, ฯลฯ เพื่อตอบสนองกรณีศึกษา (use case) ด้านการรักษาความปลอดภัยบนเครือข่ายเป็นหลัก




*รูปที่ 3.4.1: โครงสร้างสถาปัตยกรรมการออกแบบของระบบ*

- **System plane:** เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงกับเครือข่ายผู้รับบริการ เนื่องจากการทำงานของแพลตฟอร์มซึ่งอยู่ในชั้นถัดไปอาศัยการวิเคราะห์ข้อมูลที่มาจากเครือข่ายผู้รับบริการแบบ real-time เพื่อตรวจสอบและวิเคราะห์หาความผิดปกติถึงรูปแบบพฤติกรรมในการรับส่งแพ็กเก็ตข้อมูล (packet) ในเครือข่ายว่าก่อให้เกิดช่องโหว่หรือความเสี่ยงด้านความปลอดภัยหรือไม่ ทำให้ผู้ดูแลระบบสามารถตัดสินใจได้อย่างทันท่วงทีในการดำเนินการยกระดับความปลอดภัยให้แก่เครือข่ายภายใต้การกำกับดูแล โดยขั้นตอนพัฒนาระบบจะกำหนดให้แต่ละ network domain องค์กรที่สนใจทำการติดตั้งและเชื่อมต่อกับ Log collection engine ซึ่งเปรียบเสมือน Agent service ถูกติดตั้งที่ network domain ขององค์กรผู้รับบริการ เพื่อทำหน้าที่ในการรวบรวม logs การทำงานต่างๆ ประกอบด้วย network traffic, log files, critical files ที่สนใจ เช่น failed-login attempts, suspicious network traffic, ฯลฯ และส่งต่อไปยังแพลตฟอร์มฯ เพื่อทำการตรวจสอบและวิเคราะห์หาความผิดปกติหรือความเสี่ยงด้านความปลอดภัยที่อาจจะเกิดขึ้นต่อไป
- **Orchestration and enforcement plane:** เป็นชั้นที่แพลตฟอร์มฯ ติดตั้งอยู่ เพื่อทำหน้าที่ในการตรวจสอบและวิเคราะห์หาความเสี่ยงและช่องโหว่ด้านความปลอดภัยบนระบบเครือข่ายของผู้รับบริการจากข้อมูล logs ที่ได้รับมาจาก Log collection engine ที่ได้ถูกติดตั้งในชั้น System plane โดยรูปแบบการทำงานของแพลตฟอร์มมีหน้าที่หลัก 2 ประการ คือ

- **Orchestration service:** ทำหน้าที่ (1) จัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรเครือข่ายต่างๆ ได้แก่ หน่วยประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูล และระบบเครือข่ายต่างๆ ที่จำเป็นต้องใช้ในการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ และ (2) ทำการจำลอง ตั้งค่า และให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ เพื่อให้องค์กรที่รับบริการสามารถเรียกใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสที่จำลองขึ้นได้ เช่น การจำลองระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ดังเช่น Snort เพื่อคอยตรวจสอบและวิเคราะห์ถึงความผิดปกติบนระบบเครือข่ายของผู้รับบริการแบบ real-time จากข้อมูล logs ที่ได้รับ โดยวิธีการตรวจสอบนั้นจะอ้างอิงข้อมูลลายเซ็นหรือลักษณะเฉพาะของการบุกรุก (signature-based IDS) และนโยบายความปลอดภัย (security policy) ที่ผู้ใช้งานได้ทำการกำหนดไว้ล่วงหน้ามาช่วยในการตรวจสอบ
- **Enforcement service:** หากระบบตรวจพบความผิดปกติบนระบบเครือข่ายของผู้รับบริการ ระบบจะดำเนินการแจ้งเตือนให้ผู้ที่เกี่ยวข้องทราบ และดำเนินการออกคำสั่งให้สร้างเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย (VNFs) ที่เกี่ยวข้อง หรือปรับปรุงการตั้งค่านโยบายการรักษาความปลอดใหม่ เพื่อจัดการกับความผิดปกติที่ตรวจพบโดยทันที รูปที่ 3.4.2 แสดงตัวอย่างแนวทางการตรวจสอบและการป้องกันเมื่อตรวจพบความผิดปกติบนระบบเครือข่ายของหน่วยงานผู้ขอรับบริการ จะเห็นได้ว่าเบื้องต้นระบบจะทำการ mirror traffic จากหน่วยงานที่สนใจ (source) และส่ง traffic ดังกล่าวฯ ไปยังระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ดังเช่น Snort และระบบตรวจจับการบุกรุกเชิงลึก (Deep Packet Inspection: DPI)[3] ดังเช่น Suricata ตามลำดับ เพื่อทำการตรวจสอบและ




*รูปที่ 3.4.2: แสดงตัวอย่างการแผนภาพกระแสข้อมูลการบริหาร จัดการ และตรวสอบความปลอดภัยบนเครือข่าย*

[3] หมายเหตุ: ความแตกต่างระหว่างทั้งสอบประเภท คือ (1) การตรวจจับแบบ IDS จะใช้วิธีการตรวจสอบและวิเคราะห์ข้อมูลเครือข่ายเพื่อตรวจจับเหตุการณ์หรือกิจกรรมที่อาจก่อให้เกิดการบุกรุก โดยอ้างอิงลายเซ็นหรือลักษณะเฉพาะของการบุกรุกที่เป็นที่รู้จักและเก็บไว้ล่วงหน้า เมื่อหนึ่งเมื่อใดที่ตรวจพบเหตุการณ์ที่ตรงกับลายเซ็นเหล่านี้ ก็สามารถสรุปได้ว่ามีความเสี่ยงเกิดขึ้น ในขณะ

วิเคราะห์หาความผิดปกติที่เกิดขึ้น หลังจากนั้นหากตรวจพบถึงความผิดปกติ ระบบจะส่งการแจ้งเตือนกลับมายัง NFV orchestrator เพื่อดำเนินการออกคำสั่งไปยัง SDN controller ปรับปรุงการตั้งค่า policy และ forwarding rules ใหม่

- **User plane:** เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มและหน้าเว็บการจัดการและการแสดงผล เพื่ออำนวยความสะดวกแก่ดูและระบบและผู้รับบริการในการเรียกดูสถานะต่างๆ ที่เกี่ยวข้องกับการดำเนินการ โดยสิทธิ์ในการเข้าถึงจำแนกผู้ใช้งานออกเป็น 2 ประเภท คือ (1) แพลตฟอร์มอำนวยความสะดวกให้ผู้ดูแลระบบสามารถกำหนดหรือออกแบบ network service descriptors[4] ได้ตามที่ต้องการ รูปที่ 3.4.3 และ 3.4.4 แสดงตัวอย่างและผล network service descriptor ที่มีการระบุลักษณะเฉพาะต่างๆ เช่น ระบบปฏิบัติที่ใช้ จำนวนหน่วยความจำและหน่วยประมวลผล จำนวนพอร์ตการเชื่อมต่อ ฯลฯ ซึ่งเป็นข้อมูลสำคัญในการจำลองเครื่องเสมือน (VM) ฯลฯ และ (2) หน่วยงานผู้ขอรับบริการสามารถเรียกดูภาพรวมสถานการณ์การทำงานของเครือข่าย การแสดงผลข้อมูลที่ได้จาก

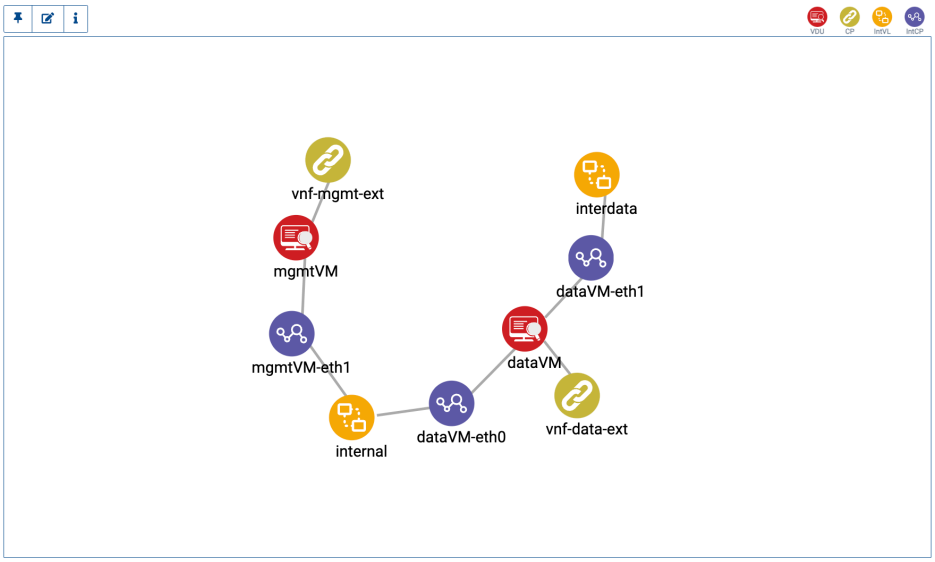
```
mgmt-cp: vnf-mgmt-ext
product-name: simple_2vm-vnf
sw-image-desc:
- id: ubuntu20.04
  image: ubuntu20.04
  name: ubuntu20.04
- id: ubuntu20.04-azure
  name: ubuntu20.04-azure
  image: Canonical:0001-com-ubuntu-server-focal:20_04-lts:latest
  vim-type: azure
- id: ubuntu20.04-gcp
  name: ubuntu20.04-gcp
  image: ubuntu-os-cloud:image-family:ubuntu-2004-lts
  vim-type: gcp
virtual-compute-desc:
- id: mgmtVM-compute
  virtual-memory:
    size: 1.0
  virtual-cpu:
    num-virtual-cpu: 1
- id: dataVM-compute
  virtual-memory:
    size: 1.0
  virtual-cpu:
    num-virtual-cpu: 1
virtual-storage-desc:
- id: mgmtVM-storage
  size-of-storage: 10
- id: dataVM-storage
  size-of-storage: 10
```

***รูปที่ 3.4.3:*** *ตัวอย่าง network service descriptor สำหรับใช้ในการจำลองเครื่องเสมือน (VM)*

ที่ (2) การตรวจจับแบบ DPI คือ การตรวจสอบแพ็กเก็ตข้อมูลเชิงลึกในระดับ packet headers และ packet contents เพื่อค้นหามัลแวร์ สแปม หรือการโจมตีที่อาจแอบแฝงมากับรูปแบบของ traffic ปกติ

[4] หมายเหตุ: Network service descriptors ในที่นี้คือ การเขียนชุดคำสั่งต่างๆ ที่เกี่ยวข้องในการจัดสรรทรัพยากรโครงสร้างพื้นฐานคลาวด์ การสร้างเครื่องเสมือน (Virtual Machines: VMs) การเชื่อมต่อเน็ตเวิร์ค และการจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง โดยชุดคำสั่งดังกล่าวฯ จะถูกเขียนให้อยู่ในรูปแบบ data sequentialization model ด้วยภาษาที่เข้าใจได้ง่าย เช่น YAML (Yet Another Markup Language)

การวิเคราะห์ภัยคุกคาม และผลการตรวจสอบความผิดปกติบนระบบเครือข่าย ฯลฯ ผ่านทาง dashboard และ โดยการติดต่อสื่อสารระหว่าง 2 ชั้นนี้จะรับส่งข้อมูลผ่าน RESTful API เป็นหลัก




*รูปที่ 3.4.4: แสดงตัวอย่างผลการจำลองเครื่องเสมือน (VM) ตาม network service descriptor ที่ได้กำหนดขึ้น*

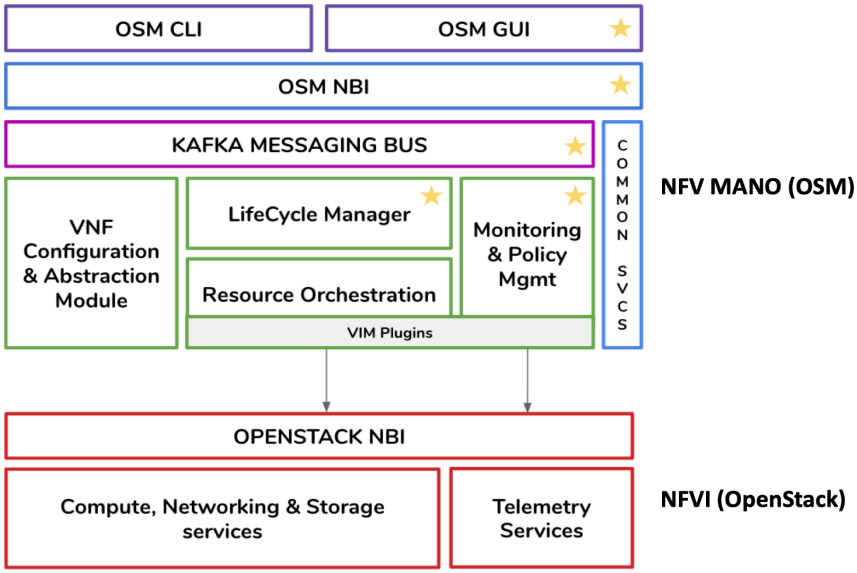
นอกเหนือจากนี้ หากมองลึกลงในรายละเอียดในส่วนที่เกี่ยวข้องกับชั้น Orchestration and enforcement plane จะพบว่าจะมีโครงสร้างสถาปัตยกรรมการออกแบบที่ประกอบด้วย 3 ระดับชั้นย่อย สนับสนุนความเป็นมาตรฐาน ETSI NFV ซึ่งประกอบด้วย (1) ชั้นโครงสร้างพื้นฐาน (NFV infrastructure layer) (2) ชั้นเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNF layer) และ (3) ชั้นบริหารและจัดการ (MANO layer) ซึ่งรายละเอียดในการออกแบบในแต่ละชั้นมีดังนี้

### 3.4.1 ชั้นออกแบบและพัฒนาระบบชั้นโครงสร้างพื้นฐาน (NFV Infrastructure layer)

อย่างที่ได้กล่าวไว้ข้างต้น หน้าที่หลักของชั้นนี้ คือ การจัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรเครือข่ายต่างๆ ได้แก่ หน่วยประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูล และระบบเครือข่ายต่างๆ ที่จำเป็นต้องใช้ในการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ โดยคลอบคลุมถึง การจัดสรรทรัพยากรต่างๆ ตามการร้องขอ การจัดเตรียมเครื่องเสมือนจริงเพื่อรองรับการรันเน็ตเวิร์คเซอร์วิสดังกล่าวฯ ตลอดจนการตรวจสอบและประเมินผลการใช้งานทรัพยากรในแต่ละเน็ตเวิร์คเซอร์วิส เพื่อปรับเพิ่ม/ลดจำนวนทรัพยากรประมวลผลและเครือข่ายให้มีความเหมาะสมต่อจำนวนเวิร์คโหลดที่เข้ามาในช่วงเวลานั้นๆ เพื่อสนับสนุนให้เกิดการใช้งานทรัพยากรที่มีอยู่ได้อย่างมีประสิทธิภาพ โดยโครงการนี้มุ่งเน้นการเลือกเครื่องมือ open source framework ที่ได้รับการยอมรับเป็นวงกว้าง มีกลุ่มชุมชนนักพัฒนาที่คอยตรวจสอบช่องโหว่ความปลอดภัย และปรับปรุง framework ให้เป็นปัจจุบันอยู่เสมอ นอกเหนือจากนี้ การเลือกใช้ open source framework ทำให้ผู้พัฒนาสามารถนำไปต่อยอด หรือปรับปรุงให้สอดคล้องตามวัตถุประสงค์การใช้งานหรือให้บริการได้อย่างอิสระ โดย OpenStack ถือเป็น open source framework ที่ผู้พัฒนาเลือกที่จะนำมาใช้เป็นส่วนหนึ่งในการวิจัยและพัฒนาแพลตฟอร์มฯ เพื่อทำหน้าที่เป็น NFVI

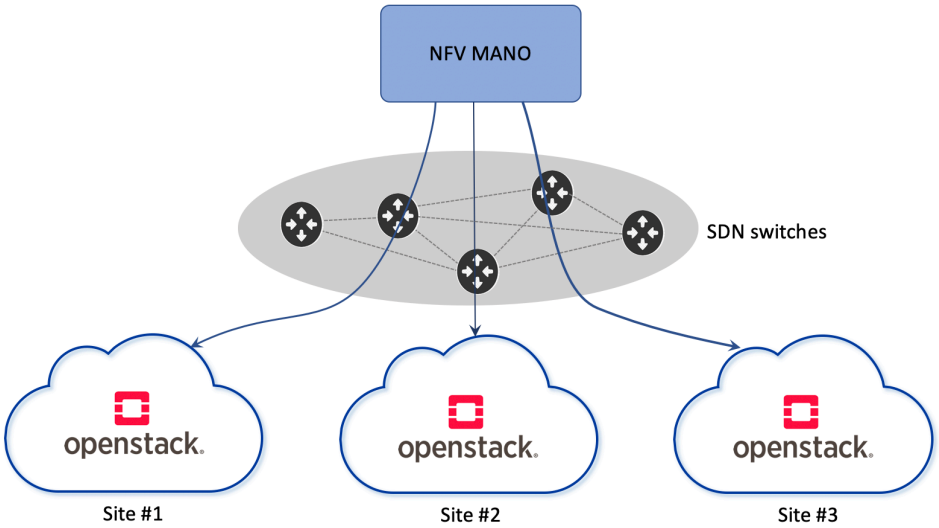
ในการจัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรต่างๆ ที่จำเป็น สนับสนุนการทำงานของเน็ตเวิร์คเซอร์วิสที่จะถูกจำลองขึ้นในลำดับถัดไป โดยความสามารถของ OpenStack มีดังนี้

- การจัดการทรัพยากรการคำนวณ: OpenStack ให้บริการด้านการจัดการเซิร์ฟเวอร์เสมือน (virtual servers) หรือ VMs ผ่านโมดูล Nova ซึ่งช่วยในการสร้าง การจัดการ และการปรับขนาดของ instances (เซิร์ฟเวอร์เสมือน) สามารถทำได้อย่างรวดเร็ว
- การจัดการเครือข่าย: หลังจากที่ VM instance ได้ถูกสร้างขึ้น โมดูล Neutron จะทำหน้าที่ในการจัดการเครือข่ายเสมือน รวมถึงการจัดการ IP addresses, subnets และระบบการเชื่อมต่อ เช่น VPNs และ Firewalls เป็นต้น
- การจัดการการจัดเก็บข้อมูล (Storage Management): OpenStack มีโมดูล Cinder สำหรับการจัดเก็บข้อมูลแบบ block storage และ Swift สำหรับ object storage ทำให้สามารถจัดเก็บข้อมูลที่เกี่ยวข้องกับเน็ตเวิร์คเซอร์วิส (VNFs) ได้อย่างมีประสิทธิภาพ
- การจัดการแพลตฟอร์มและการปรับขนาด: OpenStack มีความสามารถในการขยายหรือลดขนาดของทรัพยากรตามความต้องการใช้งาน ซึ่งเป็นสิ่งสำคัญสำหรับ NFVI ที่ต้องรองรับประสิทธิภาพการประมวลผลของแต่ละเน็ตเวิร์คเซอร์วิส (VNFs) ที่มีการเปลี่ยนแปลงอยู่เสมอ
- Interfaces และ APIs ที่มีความยืดหยุ่น: OpenStack มี APIs ที่หลากหลายและมีความยืดหยุ่น ทำให้สามารถรวมกับ VNFs และระบบจัดการอื่น ๆ ได้ง่าย
- การรองรับการทำงานแบบ Multi-tenant: OpenStack สนับสนุนการทำงานแบบ multi-tenant ซึ่งมีความสำคัญในการให้บริการ NFV ที่มีการแยกข้อมูลและทรัพยากรตามลูกค้าหรือหน่วยงานต่าง ๆ




*รูปที่ 3.4.5: แสดงการเชื่อมต่อระหว่างชั้น NFV MANO (OSM) และชั้น NFVI (OpenStack)*

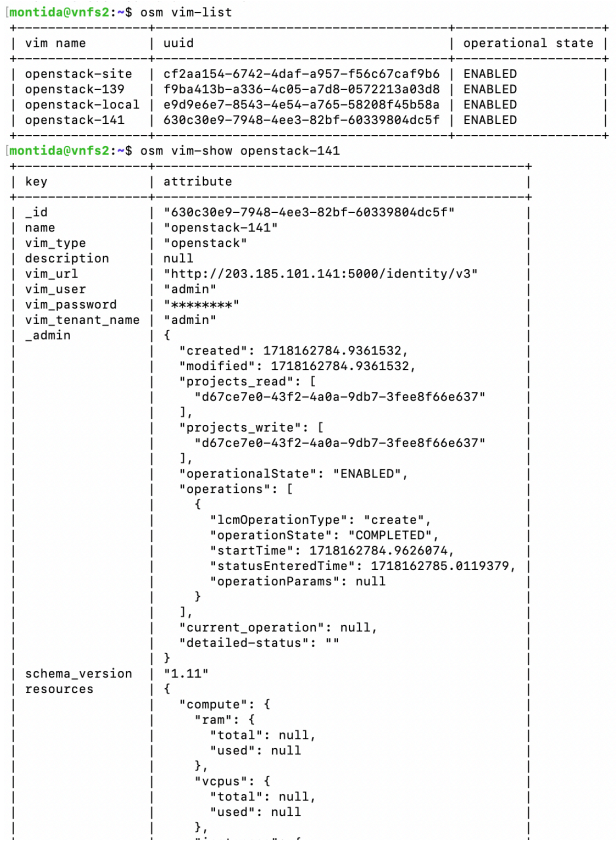
คุณสมบัติเหล่านี้ OpenStack จึงสามารถทำหน้าที่เป็น NFVI ได้อย่างมีประสิทธิภาพ รองรับการทำงานของเน็ตเวิร์คเซอร์วิสต่างๆ และการจัดการเครือข่ายเสมือนได้อย่างครอบคลุม โดยโครงสร้างการเชื่อมต่อระหว่างชั้นสนับสนุนโครงสร้างพื้นฐาน (NFV Infrastructure) เช่น Openstack และชั้นบริหารและจัดการเน็ตเวิร์คเซอร์วิส (NFV Management and Orchestration: NFV MANO) เช่น OSM แสดงดังรูปที่ 3.4.5 ซึ่งมีการติดต่อสื่อสารกันผ่าน OpenStack Northbound Interface (NBI) ทั้งนี้เราสามารถติดตั้ง จำลอง และใช้งาน OpenStack ได้หลายๆ ที่พร้อมกัน (ดังรูปที่ 3.4.6 และ 3.4.7) เพื่อลดความเสี่ยงที่อาจจะเกิดขึ้น เพิ่มความยืดหยุ่น และรองรับการขยายงานในอนาคต




*รูปที่ 3.4.6: แสดงตัวอย่างการจำลองและใช้งานโครงสร้างพื้นฐาน (NFV infrastructure) ด้วยแพลตฟอร์ม OpenStack ในหลายๆ ที่พร้อมกัน*

### 3.4.2 ออกแบบและพัฒนาระบบในส่วนที่เป็นชั้นเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNF layer)

อีกหนึ่งเหตุผลหลักสำคัญของการประยุกต์ใช้เทคโนโลยีการจำลองระบบเสมือนจริง คือ การแก้ไขปัญหาและลดข้อจำกัดที่ประสบในระบบเครือข่ายแบบดั้งเดิมที่เกี่ยวข้องทั้งในเรื่องค่าใช้จ่ายด้านการลงทุน ความยุ่งยากซับซ้อนในการเข้าถึงและจัดการทรัพยากรเครือข่ายผ่านอุปกรณ์ทางกายภาพ และความล่าช้าในการจำลองเน็ตเวิร์คเซอร์วิส ดังนั้น เพื่อแก้ปัญหาดังกล่าวฯ แนวคิดหลักของงานวิจัยนี้ คือ การจำลองเน็ตเวิร์คเซอร์วิสต่าง ๆ ซึ่งเดิมจำเป็นต้องอาศัยการติดตั้งอุปกรณ์ฮาร์ดแวร์ที่มีหน้าที่การทำงานแบบจำเพาะในการดำเนินการ เช่น router, firewall, Network Address Translator (NAT), ฯลฯ โดยนำเน็ตเวิร์คเซอร์วิสต่าง ๆ ดังกล่าวมาออกแบบพัฒนาในรูปแบบซอฟต์แวร์แอพพลิเคชั่นที่สามารถทำงานได้บนอุปกรณ์ฮาร์ดแวร์หรือเครื่องเซิร์ฟเวอร์ทั่วไปได้ ลดการพึ่งอุปกรณ์เน็ตเวิร์คฮาร์ดแวร์จำเพาะและค่าใช้จ่ายในการบำรุงรักษาอุปกรณ์ระยะยาว ซึ่งรูปแบบการจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNFs) นั้นสามารถนำไปติดตั้งบนโครงสร้างพื้นฐานเครือข่ายเสมือน (Network Functions



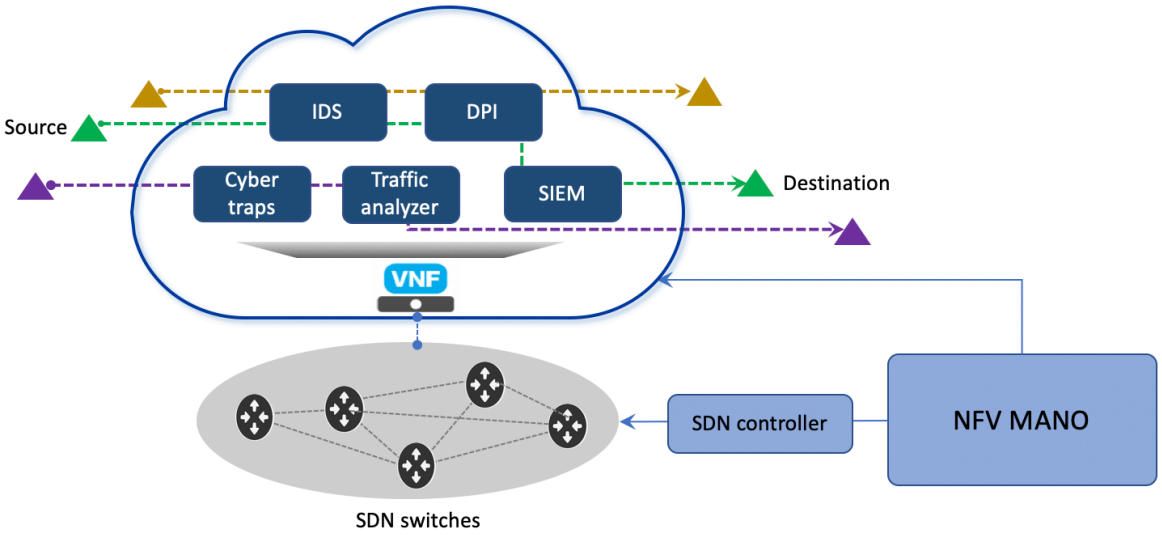

***รูปที่ 3.4.7:*** *หน้า CLI แสดงรายละเอียดและข้อมูล OpenStack (cloud-based infrastructure) ที่ระบบได้ดำเนินการเชื่อมต่อ*

Virtualization Infrastructure : NFVI) ดังเช่น OpenStack เพื่อสร้างสภาวะแวดล้อมที่เหมาะสมต่อการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการต่อไป

กล่าวได้ว่า ชั้นนี้เปรียบเสมือนแหล่งรวมเน็ตเวิร์คเซอร์วิสประเภทต่างๆ ที่ผู้ดูแลระบบต้องการที่จะจำลองขึ้นเพื่อรองรับการให้บริการแก่ผู้ใช้ไว้ด้วยกัน (รูปที่ 3.4.8 แสดงตัวอย่างการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่าง) โดยหน้าที่รับผิดชอบหลักในชั้นนี้ คือ (1) การสร้างและให้บริการเน็ตเวิร์คเซอร์วิสประเภทต่างๆ หรือที่เรียกว่า VNFs as a Service (VNFaaS) ตามการร้องขอของผู้ใช้ (2) จัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้นให้มีขีดความสามารถในการทำงานบนทรัพยากรโครงสร้างพื้นฐานคลาวด์ รวมถึงดำเนินการตั้งค่าต่างๆ ตามนโยบาย (policy) ที่ผู้ดูแลระบบกำหนดไว้ (3) อัพเกรดซอฟต์แวร์ที่เกี่ยวข้อง (4) ปรับลดหรือขยายจำนวนเน็ตเวิร์คเซอร์วิส (scaling in/out VNFs) และ (5) ถอดถอนการติดตั้งเน็ตเวิร์คเซอร์วิสเมื่อไม่มีการใช้งาน คุณสมบัติเหล่านี้ จะช่วยอำนวยความสะดวกให้แก่หน่วยงานผู้รับบริการ (customers) ในการติดตั้งและจัดการเน็ตเวิร์เซอร์วิสได้อย่างยืดหยุ่น ลดความยุ่งยากในการทำความเข้าใจรายละเอียดด้านเทคนิคเชิงลึก ซึ่งต้องอาศัยผู้เชี่ยวชาญเฉพาะทางในการดูแล โดยโมเดลการให้บริการเน็ตเวิร์คเซอร์วิสแบ่งประเภทผู้ใช้งานออกเป็น 2 กลุ่ม ดังนี้

- กลุ่มผู้ให้บริการจะมีหน้าที่รับผิดชอบ ดังนี้

- จัดเตรียมเน็ตเวิร์คเซอร์วิสประเภทต่าง ๆ เช่น เน็ตเวิร์คเซอร์วิสที่เกี่ยวข้องกับด้านความปลอดภัย ได้แก่ firewall, IDS, DPI, ฯลฯ




*รูปที่ 3.4.8: แสดงตัวอย่างการจำลองเน็ตเวิร์คเซอร์วิส (VNFs) ด้านความปลอดภัยประเภทต่างๆ และบริบทความต้องการใช้งานของผู้รับบริการที่ต้องการใช้งาน VNFs แต่ละประเภทแตกต่างกันออกไป*

- บำรุงรักษาเน็ตเวิร์คเซอร์วิส เช่น ดำเนินการตั้งค่า rules หรือ policy ต่างๆ ที่เกี่ยวข้อง อัพเดตซอฟต์แวร์ หรืออัพเดตแพทช์ (Patch update) เพื่อปิดช่องโหว่หรือแก้ไขข้อผิดพลาด
- เลือก จัดการ ปรับเปลี่ยนเน็ตเวิร์คเซอร์วิสให้เหมาสมตามลักษณะการใช้งานและความต้องการของผู้รับบริการ ทั้งในแง่ประเภทเน็ตเวิร์คเซอร์วิสและจำนวนการ deployment
- บริหารต้นทุนและค่าใช้จ่ายในการให้บริการ

- กลุ่มผู้รับบริการจะมีหน้าที่รับผิดชอบ ดังนี้
  - ให้ข้อมูลเกี่ยวกับระบบที่เข้ารับบริการ เช่น การให้สิทธิ์เข้าถึง logs การทำงานของเครือข่าย
  - ให้ข้อมูลเกี่ยวกับความต้องการในการใช้งาน เช่น การระบุเงื่อนไข (conditions) หรือนโยบาย (policy) ในการติดตั้ง VNF ในสถานการณ์การทำงานรูปแบบต่าง ๆ
  - จัดเตรียมสภาพแวดล้อมของระบบให้รองรับเชื่อมต่อกับแพลตฟอร์มของผู้ให้บริการ

สำหรับตัวอย่างเครื่องมือ open source ที่ทางโครงการสนใจและเลือกใช้พัฒนาเพื่อสนับสนุนการจำลองและให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ในชั้นนี้ ประกอบด้วย

- Intrusion Detection System (IDS): Snort ถือเป็น opensource framework สำหรับการพัฒนา IDS ที่ได้รับยอมรับอย่างแพร่หลาย ทำหน้าที่ตรวจจับและป้องกันการโจมตีแบบต่าง ๆ บนเครือข่าย
- Deep Packet Inspection (DPI): Suricata เป็น opensource framework ที่ใช้แป็นเครื่องมือในการวิเคราะห์และตรวจจับรูปแบบการโจมตีบนเครือข่ายทั้งที่เป็นแบบรู้จัก (signature based

detection) และแบบไม่รู้จัก (unknown anomalies) โดยการวิเคราะห์แพ็กเก็ตข้อมูลเชิงลึกตั้งแต่ชั้น data link ไปจนถึงชั้น application layer

- Cyber traps: Honeypot เป็น open source framework เพื่อทำหน้าที่เป็นเหยื่อล่อและดักจับผู้โจมตีที่พยายามเข้าถึงระบบเครือข่าย โดยการตรวจจับและเก็บข้อมูลที่สำคัญในการโจมตี เพื่อประโยชน์ต่อการวินิจฉัยและปรับปรุงแนวทางการป้องกันระบบเครือข่ายต่อไปในอนาคต
- Traffic analyzer: Zeek เป็น open source framework ที่ใช้วิเคราะห์ข้อมูลการจราจรในเครือข่าย เพื่อตรวจจับเหตุการณ์ที่ผิดปกติหรือน่าสงสัย
- SIEM: Wazuh เป็น open source framework ที่ใช้ในการจัดการ รวบรวมข้อมูลจาก logs วิเคราะห์ข้อมูลด้านความปลอดภัย ตรวจจับการบุกรุก และจัดทำรายงาน ช่วยให้ผู้ดูแลระบบด้านความปลอดภัยสามารถป้องกันและตอบสนองต่อภัยคุกคามต่างๆ ได้อย่างทันถ่วงที

### 3.4.3 ออกแบบและพัฒนาระบบในส่วนที่เป็น NFV MANO layer

หากอ้างอิงโครงสร้างสถาปัตยกรรมการออกแบบเทคโนโลยี NFV ชั้นบริหารและจัดการเน็ตเวิร์คเซอร์วิส (MANO layer) ถือว่าเป็นแกนหลักสำคัญของการขับเคลื่อนระบบ รับผิดชอบในการควบคุมลำดับการทำงานภายในระบบ และประสานให้การทำงานเป็นไปอย่างอัตโนมัติสอดคล้องกับความต้องการใช้งานหรือนโยบายที่กำหนดไว้ โดยคลอบคลุมถึง การบริหารและจัดการเน็ตเวิร์คโทโพโลยีในภาพรวม ประสานงานกับโมดูลต่างๆ ที่เกี่ยวข้องทั้งในชั้น NFVI เพื่อจัดสรรทรัพยากรโครงสร้างพื้นฐานให้พร้อมสำหรับการใช้งาน และชั้น VNF เพื่อจำลองเน็ตเวิร์คเซอร์วิสตามการร้องขอ อีกทั้ง ดูแลวัฏจักรการทำงานของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น เริ่มตั้งแต่การติดตั้ง เพิ่ม/ลดจำนวนหน่วยการทำงาน ไปจนถึงการถอดถอดการติดตั้งเมื่อยกเลิกใช้งาน รูปที่ 3.4.1 แสดงหน่วยการทำงานต่างๆ ที่สำคัญ โดยบทบาทหน้าที่และความสัมพันธ์ระหว่างหน่วยการทำงานต่างๆ มีรายละเอียดดังนี้

- NFV MANO Stack: ซึ่งประกอบด้วย 3 หน่วยการทำงานสำคัญ ได้แก่
    - NFV Orchestrator (NFVO): ซึ่งเป็นเสมือนหน่วยการทำงานแกนหลักของระบบ ทำหน้าที่สำคัญในการบริหารและจัดการทรัพยากรเครือข่าย (resource orchestration) และเน็ตเวิร์คเซอร์วิส (service orchestration) ทั้งหมดในระบบ โดยเริ่มตั้งแต่ออกคำสั่งในการจัดเตรียมทรัพยากรโครงสร้างพื้นฐานที่จำเป็นสำหรับการจำลองเน็ตเวิร์คเซอร์วิส การสร้างและการเชื่อมต่อเน็ตเวิร์คเซอร์วิสตามการร้องขอ ติดตามสถานะการทำงาน ประสานการทำงานกับหน่วยการทำงานต่างๆ ที่เกี่ยวข้องให้การทำงานภายในระบบเป็นไปอย่างราบรื่น ไปจนถึงการถอดถอนการติดตั้งและยกเลิกใช้งาน นอกจากนี้ NFVO ยังรองรับการเชื่อมต่อกับหน่วยการทำงานอื่นๆที่เกี่ยวข้อง ได้แก่ Policy interpreter เพื่อการกำหนดมาตรการความปลอดภัยบนเครือข่าย และ RESTful API รองรับการนำเข้าข้อมูลการตั้งค่าจากผู้ใช้งานหรือผู้ดูแลระบบ และการส่งข้อมูลภาพรวมการทำงานและสถานะหน่วยการทำงานภายในเพื่อนำไปแสดงผลแก่ผู้ใช้งาน

- VNF Manager (VNFM): ทำหน้าที่เป็นผู้จัดการและรับผิดชอบในการดูแลวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส ตั้งแต่การสร้าง ตั้งค่า อัพเดท และถอดถอนเน็ตเวิร์คเซอร์วิส (VNF instance) โดยจะทำงานภายใต้การควบคุมของ NFVO
- Virtualized Infrastructure Manager (VIM): ทำหน้าที่เป็นผู้จัดการ รับผิดชอบในการจัดเตรียมทรัพยากรโครงสร้างพื้นฐานต่างๆ ที่จำเป็นต่อการจำลอง VNF instance โดยจะประสานการงานร่วมกับแพลตฟอร์มที่ให้บริการโครงสร้างพื้นฐานคลาวด์ต่างๆ (NFVI) เช่น OpenStack, AWS, GCP, ฯลฯ โดยทำงานภายใต้การควบคุมของ NFVO

- Policy interpreter: ทำหน้าที่ตีความนโยบายความปลอดภัยที่ได้กำหนดและจัดเก็บภายใน Security policy repository มีการทำงานร่วมกับ NFVO เพื่อให้แน่ใจว่าการทำงานมีความสอดคล้องและเป็นไปตามนโยบายความปลอดภัยที่กำหนดไว้
- Security service catalog: ทำหน้าที่จัดการแค็ตตาล็อก (catalog) ของแต่ละเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย ซึ่งรวมถึงการเก็บข้อมูลต่างๆ ที่เกี่ยวข้อง (VNF descriptors) ในฐานข้อมูล เพื่อให้สามารถเข้าถึงและจัดการข้อมูลดังกล่าวฯ ได้ตามต้องการ โดย NFVO จะเรียกใช้ VNF descriptors จาก security service catalog นี้ และอ้างอิงการตั้งค่านโยบายความปลอดภัยจาก Security policy repository ในการจำลองเน็ตเวิร์คเซอร์วิสที่เกี่ยวข้อง
- Virtual Network Functions (VNFs): เป็นการจำลองเน็ตเวิร์คเซอร์วิสหรือฟังก์ชันเครือข่ายต่างๆ ได้โดยไม่ต้องพึ่งพาฮาร์ดแวร์เฉพาะ ตัวอย่างเช่น จำลองฟังก์ชันของ firewall หรือ IDS ด้วยคุณลักษณะของ VNFs ที่รองรับการทำงานบนโครงสร้างพื้นฐานแบบเสมือน จึงสามารถปรับเปลี่ยนตามปริมาณทรัพยากรที่มีอยู่ ความต้องการผู้ใช้งาน หรือรูปแบบการบริการของให้บริการได้อย่างรวดเร็ว โดยไม่จำเป็นต้องติดตั้งหรือเปลี่ยนแปลงฮาร์ดแวร์ สามารถลดต้นทุนในการพึ่งพาฮาร์ดแวร์ที่มีค่าใช้จ่ายสูง โดย VNF ทำงานร่วมกับ NFVO ผ่าน VNFM เพื่อประสานงานในการติดตั้ง กำหนดการตั้งค่า และการจัดการ VNF ให้เป็นไปแบบอัตโนมัติ
- NFV Infrastructure (NFVI) มีหน้าที่ในจัดสรรทรัพยากร เช่น ทรัพยากรการประมวลผล การเชื่อมต่อเครือข่าย พื้นที่จัดเก็บข้อมูล และสร้างสภาพแวดล้อมเสมือน เช่นการสร้างและการจัดการ virtual machines (VMs) หรือ containers เพื่อรองรับการทำงานของ VNF ช่วย VNF ให้สามารถทำงานได้อย่างเป็นอิสระจากฮาร์ดแวร์ โดย NFVI ทำงานร่วมกับ NFVO ผ่าน VIM เพื่อประสานงานในการจัดสรรทรัพยากรพื้นฐานต่างๆ ที่จำเป็น ตรวจสอบการเข้าถึงทรัพยากร ตรวจสอบประสิทธิภาพการใช้งานทรัพยากร และการจัดการทรัพยากรให้เป็นไปแบบอัตโนมัติ
- SDN controller: ทำหน้าหน้าที่ควบคุมและจัดการไหลของ traffic บนเครือข่ายไปยัง VNFs ที่ได้ถูกจำลองขึ้นให้ตรงตามข้อกำหนดและนโยบายการรักษาความปลอดภัยที่ได้กำหนดไว้ โดย SDN controller จะรับคำสั่งที่เกี่ยวข้องกับการตั้งค่าต่างๆ เช่น policy/configuration rules จาก NFVO ผ่าน NBI

- Kafka/RESTful APIs: เป็น Nortbound Interface (NBI) ของระบบ ซึ่งทำหน้าที่ติดต่อกับ User plane เพื่อรับเข้าข้อมูลการตั้งค่าต่างๆ เช่น security policy จากผู้ใช้งาน และส่งต่อไปยัง Policy interpreter เพื่อทำการบันทึกและปรับเปลี่ยนนโยบายความปลอดภัยบนเครือข่ายให้สอดคล้องกับเงื่อนไขใหม่ที่กำหนด นอกจากนี้ Kafka/RESTful APIs ยังทำหน้าที่ส่งข้อมูลภาพรวมและสถานะของหน่วยการทำงานที่ได้จาก VNFs เช่น ผลการตรวจจับความผิดปกติ ประเภทภัยคุกคาม และความเสี่ยงต่างๆ ที่ตรวจพบ ฯลฯ ไปยัง User plane เพื่อนำไปจัดแสดงในรูปแบบ dashboard ให้แก่ผู้ใช้งาน

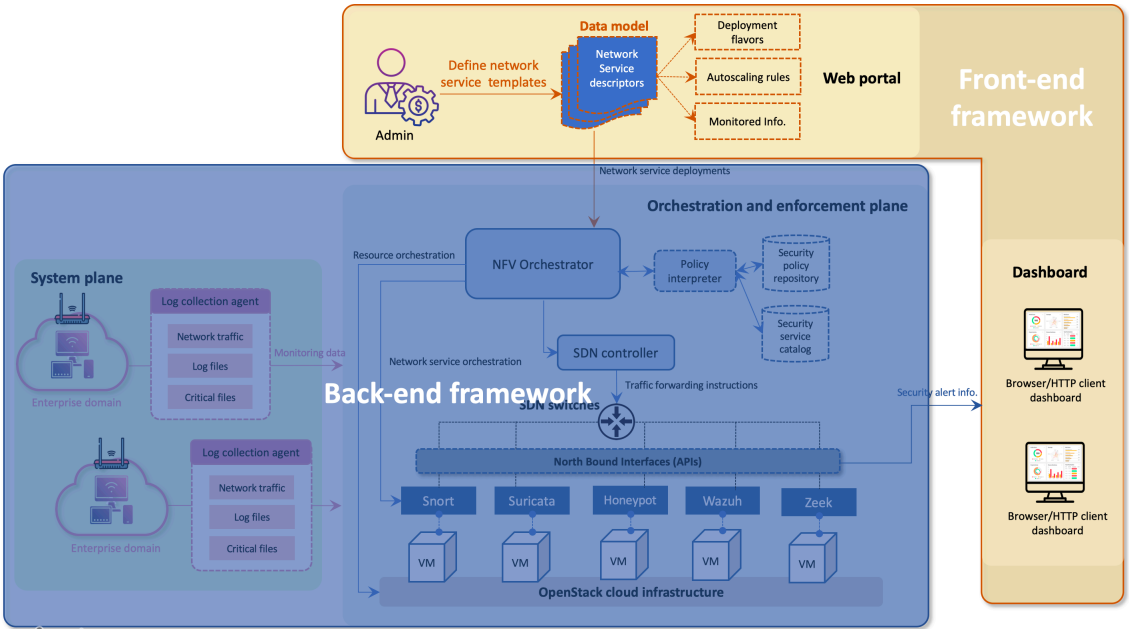
โดย Opensource MANO (OSM) ถือเป็นเครื่องมือ opensource framework ที่ทางโครงการเลือกใช้ในการวิจัยและพัฒนาเพื่อสนับสนุนการบริการและจัดการเน็ตเวิร์คเซอร์วิส เนื่องจาก OSM ได้รับการสนับสนุนโดย European Telecommunications Standards Institute (ETSI) ซึ่งเป็นหน่วยงานมาตรฐานสำคัญในอุตสาหกรรมโทรคมนาคม ทำให้มั่นใจได้ว่า OSM สอดคล้องกับมาตรฐานที่ยอมรับกันอย่างกว้างขวาง นอกจากนี้ OSM ยังมีการออกแบบเพื่อรองรับการทำงานร่วมกับแพลตฟอร์มเทคโนโลยีอื่น ๆ และการเชื่อมต่อกับหน่วยการทำงานภายนอกได้อย่างมีประสิทธิภาพ จึงมีความเหมาะสมในการเลือกนำมาต่อยอดการพัฒนาในส่วนของ NFV MANO Layer ของโครงการนี้

### 3.4.4 ออกแบบและพัฒนา web portal และส่วนติดต่อกับผู้ใช้งาน (API)

Web Portal เป็นส่วนของ User plane ทำหน้าที่ติดต่อกับผู้ใช้งาน ซึ่งสามารถแบ่งออกได้เป็น 2 ประเภท ได้แก่ เจ้าหน้าที่ผู้ดูแลระบบ และหน่วยงานผู้รับบริการใช้งานระบบ โดยผู้ใช้งานแต่ละประเภทจะมีสิทธิเข้าถึงข้อมูลในระดับที่แตกต่างกัน

- ผู้ดูแลระบบ: ในส่วนเจ้าหน้าที่ผู้ดูแลระบบสามารถเข้าถึงหน้าการจัดการข้อมูลหน่วยงานผู้ใช้บริการ เช่น การเพิ่ม ลบ แก้ไขข้อมูลผู้ใช้งาน นอกจากนี้ยังสามารถเข้าดูภาพรวมสถานการณ์ทำงานของระบบโดยรวม สามารถเข้าถึงรายการภัยคุกคามที่ตรวจพบบนเครือข่าย แต่ไม่สามารถเข้าถึงข้อมูลที่เป็นความลับหรือข้อมูลในรายละเอียดเชิงลึกของแต่ละผู้ใช้งานได้

- ผู้รับบริการ: ในส่วนของหน่วยงานผู้ใช้บริการ สามารถเข้าถึงข้อมูลการแสดงผลต่างๆ ที่เกี่ยวข้อง เช่น สามารถเข้าดู สร้าง และแก้ไขการตั้งค่านโยบายความปลอดภัยบนเครือข่ายได้ สามารถเข้าถึง Dashboard ซึ่งแสดงภาพรวมสถานะของเครือข่ายของตนเอง สามารถเข้าถึงข้อมูลสถานะของบริการฟังก์ชั่นความปลอดภัยบนเครือข่ายที่แพลตฟอร์มได้ทำการติดตั้งในกรณีที่ตรวจพบภัยคุกคาม สามารถได้รับการแจ้งเตือนภัยคุกคามและรูปแบบการดำเนินการเพื่อลดความเสี่ยง และสามารถออกรายงานสรุปผลข้อมูลภัยคุกคามที่ตรวจพบย้อนหลังได้




*รูปที่ 3.4.9: แสดงโครงสร้างการทำงานที่เกี่ยวข้องกับ web portal*

รูปที่ 3.4.9 แสดงหน่วยการทำงานต่างๆ ที่เกี่ยวข้องกับ web portal ซึ่งแบ่งออกได้เป็น 3 ส่วนหลัก ได้แก่

- Front-end framework: ประกอบด้วย 2 ส่วนนการทำงานหลัก (1) web portal ซึ่งเป็นส่วนการจัดการข้อมูลหน่วยงานผู้ใช้บริการ และรับเข้าข้อมูลจากผู้ใช้งาน เช่น ข้อมูลการตั้งค่านโยบายความปลอดภัย และ (2) dashboard ทำหน้าที่นำข้อมูลมาแสดงผลให้แก่ผู้ใช้งาน เช่น การแสดงภาพรวมสถานการณ์การทำงานของเครือข่าย ผลการตรวจสอบความผิดปกติบนระบบเครือข่าย รวมถึงการจัดทำรายงาน ทั้งนี้ข้อมูลที่นำมาแสดงผลบน web portal และรูปแบบคำสั่งการทำงานที่รองรับ จะมีความแตกต่างตามบริบทผู้ใช้งาน เช่น เจ้าหน้าที่ผู้ดูแลระบบมีสิทธิ์เข้าถึงหน้าการจัดการผู้ใช้งานระบบ (user management) ในขณะที่ผู้ใช้งานหรือผู้รับบริการจะไม่มีสิทธิ์เข้าถึงหน้าดังกล่าวฯ
- Back-end framework เป็นหน่วยการทำงานต่างๆ ที่ทำงานร่วมกันในระบบหลังบ้านเพื่อสนับสนุนการบริหารและจัดการเน็ตเวิร์คเซอร์วิสแบบเสมือนด้านความปลอดภัย โดยระบบจะรับการตั้งค่าต่างๆ เช่น การตั้งค่าเครือข่าย นโยบายความปลอดภัย ฯลฯ ผ่านทาง web portal และผลที่ได้จากการวิเคราะห์และประมวลผลจะถูกนำมาแสดงผลผ่านทาง dashboard

- Application Programming Interfaces (APIs): เป็น NBI ของระบบ ซึ่งทำหน้าเป็นตัวกลางในการรับ-ส่งข้อมูลระหว่าง front-end framework (user plane) และ back-end framework (orchestration and enforcement plane) เช่น การรับข้อมูลการปรับปรุงตั้งค่านโยบายความปลอดภัยจากผู้ใช้งาน และนำส่งไปยัง Policy intepreter เพื่อตีความนโยบายความปลอดภัยที่ได้กำหนด โดยทำงานร่วมกับ NFVO เพื่อปรับปรุงเงื่อนไขการตรวจสอบและให้แน่ใจว่าเงื่อนไขการทำงานของแต่ละ VNFs ที่ถูกสร้างขึ้นมีการทำงานที่สอดคล้องกับนโยบายความปลอดภัยที่กำหนดไว้ต่อ สามารถรับมือภัยคุกคามบนเครือข่ายให้สอดคล้อง และ การรับข้อมูลผลการตรวจสอบและวิเคราะห์จากแต่ละ VNFs มาแสดงผลบน dashboard

**3.5 บทสรุป**

ปัจจุบันพบว่าหลายๆ ประเทศทั้งในทวีปยุโรปและเอเชียต่างให้ความสนใจที่จะพัฒนาและขับเคลื่อนงานวิจัยที่เกี่ยวข้องกับเทคโนโลยีการจำลองระบบเสมือนจริง ไปสู่การใช้งานจริงภายใต้ทุนอุดหนุนวิจัยร่วมกัน ดังเช่น โครงการ T-NOVA, ANASTACIA, Open source MANO (OSM), ONAP (Open Network Automation platform), PALANTIR, และ KDDI & NEC coporation เนื่องจากประเทศส่วนใหญ่ต่างเห็นพ้องต้องกันว่าเทคโนโลยีการจำลองระบบเสมือนจริงจะนำมาซึ่งการเปลี่ยนแปลงโครงสร้างเครือข่ายครั้งสำคัญ อันสามารถลดข้อจำกัดต่างๆ ที่ประสบในระบบเครือข่ายแบบดั้งเดิมได้ ทั้งในเรื่อง การช่วยลดค่าใช้จ่ายสำหรับการจัดซื้อและบำรุงรักษาอุปกรณ์เครือข่ายที่เกิดขึ้นภายในองค์กร การช่วยเพิ่มประสิทธิภาพในการตอบสนองการให้บริการและความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส ความสามารถในการจัดสรรและใช้งานทรัพยากรได้อย่างมีประสิทธิภาพ มีความยืดหยุ่น เหมาะสม และสอดคล้องกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ ซึ่งในหัวข้อนี้ จะมุ่งเน้นการดำเนินงานออกเป็น 2 ระยะ โดยระยะที่ 1 ดำเนินการศึกษา และทำความเข้าใจหลักการออกแบบของแต่ละโครงการฯ แนวคิดการออกแบบสถาปัตยกรรมของระบบ คุณสมบัติและคอมโพเนนท์ต่างๆ เพื่อประโยชน์ต่อการนำมาปรับใช้และคัดเลือกเครื่องมือ (open source tools) ที่เหมาะสมต่อการพัฒนาแพลตฟอร์มฯ ต่อไป สำหรับการดำเนินงานในระยะที่ 2 มีความเกี่ยวข้องกับการวิจัย พัฒนา และออกแบบสถาปัตยกรรมแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง ดังนั้น เพื่อให้สอดคล้องกับกรณีศึกษา (security use case) ว่าด้วยแนวทางการป้องกันและรักษาความปลอดภัยในองค์กร โดยแนวคิดการออกแบบสถาปัตยกรรมระบบ จะแบ่งการทำงานออกเป็น 3 ระดับชั้นด้วยกัน ดังนี้ (1) System plane ซึ่งเป็นส่วนที่ทำหน้าที่เชื่อมต่อแพลตฟอร์มฯ กับเครือข่ายผู้รับบริการ เพื่อทำหน้าที่เฝ้าติดตาม และรวบรวม ข้อมูล logs การทำงานต่างๆ ที่สนใจ และส่งต่อไปยังแพลตฟอร์มฯ เพื่อทำการตรวจสอบและวิเคราะห์หาความผิดปกติ หรือความเสี่ยงด้านความปลอดภัยที่อาจจะเกิดขึ้นต่อไป (2) Orchestration and enforcement plane ซึ่งถือว่าเป็นแกนหลักสำคัญของการพัฒนาแพลตฟอร์มฯ เพื่อทำหน้าที่ในการบริหาร จัดการ และประสานการทำงานต่างๆ ภายในระบบให้เป็นไปอย่างราบรื่นและมีประสิทธิภาพ ทั้งในส่วนของการควบคุม ดูแล และจัดสรรทรัพยากรเครือข่าย การจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (resource and network service deployment) และทำหน้าที่ในการตรวจสอบความเสี่ยงหรือช่องโหว่ด้านความปลอดภัยที่อาจจะเกิดขึ้นด้วยการวิเคราะห์ข้อมูล logs และ (3) User plane เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มฯ กับผู้ใช้งาน เพื่ออำนวยความสะดวกให้แก่ (1) ผู้ดูแลระบบในการ

สร้างและออกแบบพิมพ์เขียว (network service descriptor) ของแต่ละเน็ตเวิร์คเซอร์วิสได้ตามที่ต้องการ รวมไปถึงการเรียกดูสถานะๆ ต่างๆ ที่เกิดขึ้นภายในระบบ ดังเช่น สถานะใช้งานทรัพยากร ฯลฯ และ (2) ผู้ใช้ซึ่งเป็นหน่วยงานที่รับบริการ ต่อการเรียกดูข้อมูลการตรวจจับ ข้อมูลที่ได้จากการวิเคราะห์ภัยคุกคาม และผลการตรวจอสอบข้อมูลผ่าน dashboard ในรูปแบบกราฟแสดงผลประเภทต่างๆ

# บทที่ 4
# กิจกรรมที่ 3: ทดสอบประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน

## 4.1 บทนำ

จากแนวทางการออกแบบสถาปัตยกรรมแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงเชิงภาพรวมที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา ซึ่งแนวทางการออกแบบได้แบ่งการทำงานออกเป็น 3 ระดับชั้นใหญ่ๆ ด้วยกัน ได้แก่ (1) System plane เป็นส่วนที่ทำหน้าที่เชื่อมต่อกับเครือข่ายผู้รับบริการ โดยมี log collection agent ทำหน้าที่ในการรวบรวมและจัดส่ง logs การทำงานต่างๆ ที่เกี่ยวข้องไปยังชั้นถัดไป (2) Orchestration and enforcement plane รับผิดชอบในการควบคุมดูแล บริหาร จัดสรรทรัพยากรเครือข่าย และการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ให้พร้อมใช้งาน รวมถึงการทำหน้าที่ตรวจสอบและวิเคราะห์ความเสี่ยงด้านภัยคุกคามที่อาจจะเกิดขึ้น และ (3) User plane เป็นส่วนการติดต่อกับผู้ใช้งานภายนอก เพื่ออำนวยความสะดวกให้แก่ ผู้ดูแลระบบในการสร้างและออกแบบพิมพ์เขียว (network service descriptor) ทีเกี่ยวข้องกับเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ได้ตามที่ต้องการ รวมไปถึงการเรียกดูสถานะๆ ต่างๆ ที่เกิดขึ้นภายในระบบ ดังเช่น สถานะใช้งานทรัพยากร ฯลฯ และผู้ใช้ซึ่งเป็นหน่วยงานที่รับบริการ ต่อการเรียกดูข้อมูลการตรวจจับ ข้อมูลที่ได้จากการวิเคราะห์ภัยคุกคาม และผลการตรวจอสอบข้อมูลผ่าน dashboard ในรูปแบบกราฟแสดงผลประเภทต่างๆ

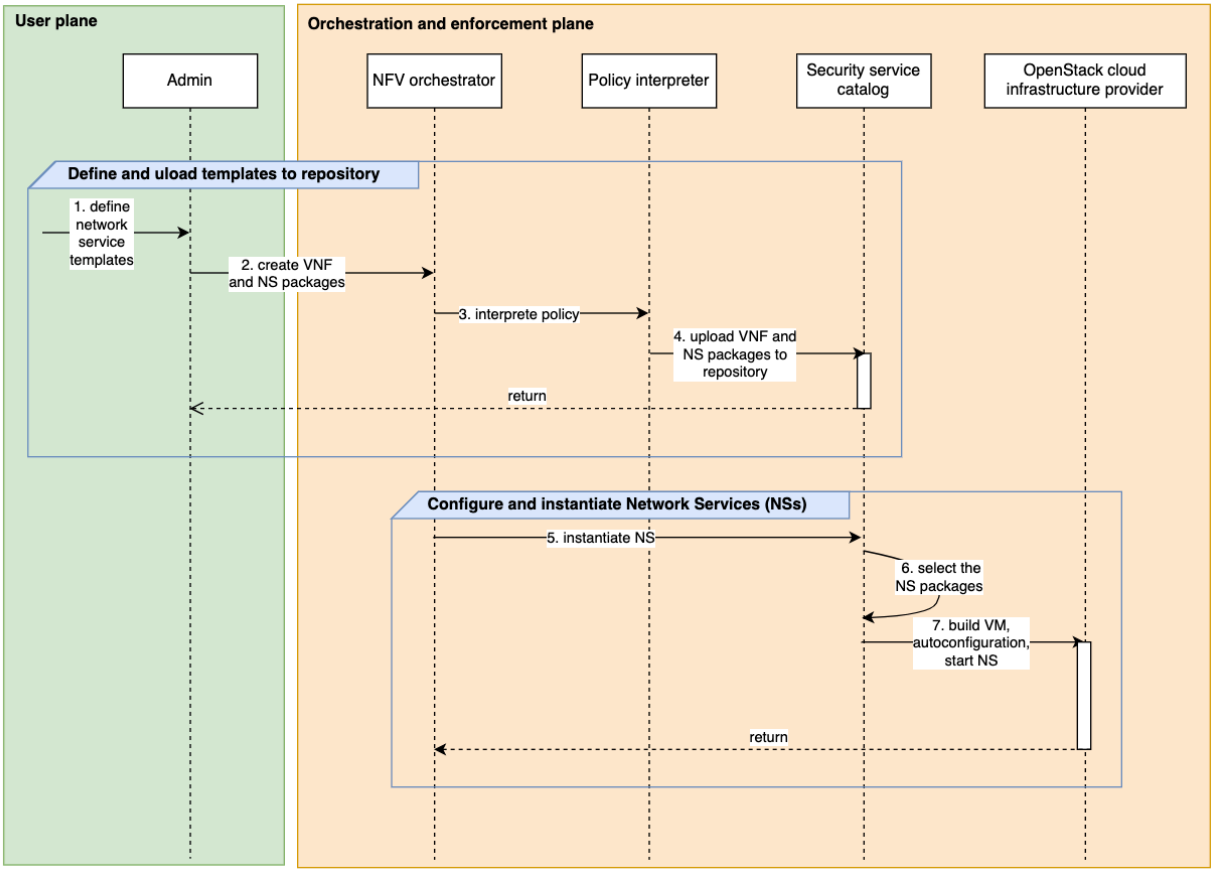
ในหัวข้อนี้จะพูดถึงในรายละเอียดของแต่ละองค์ประกอบที่เกี่ยวกับกับการพัฒนาแพลตฟอร์มฯ ตั้งแต่ขั้นตอนเริ่มต้นของการตั้งค่า ทดสอบฟังก์ชั่นการทำงาน และการวัดประสิทธิภาพของระบบแบบองค์รวม

## 4.2 การทำงานเชิงหน้าที่ (funcitonality)

เพื่อให้ง่ายต่อการทำความเข้าใจ การทำงานของแพลตฟอร์มฯ จะแบ่งออกเป็น 2 ระยะหลักๆ ดังนี้

- Pre-configuration and deployment time: โดยในระยะนี้จะเกี่ยวข้องกับการสร้างและออกแบบพิมพ์เขียวเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (ได้แก่ Suricata, Snort, Zeek) การตั้งค่านโยบายความปลอดภัยให้กับแต่ละเน็ตเวิร์คเซอร์วิส การจัดเตรียมทรัพยากรโครงสร้างพื้นฐาน การตั้งค่าการเชื่อมต่อเครือข่าย และการจำลองเน็ตเวิร์คเซอร์วิสให้พร้อมสำหรับการใช้งาน รูปที่ 4.2.1 แสดงภาพรวมการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้อง เพื่อสร้าง ออกแบบ ตั้งค่า และจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยดังกล่าวฯ สำหรับรองรับการตรวจสอบและวิเคราะห์ภัยคุกคามที่จะเกิดขึ้นในระยะต่อไป
- Run-time: ระยะนี้มีความเกี่ยวข้องกับการตรวจสอบและวิเคราะห์หาความผิดปกติจากข้อมูล logs ที่ได้รับมาจาก log collection agent ซึ่งได้ถูกติดตั้งไว้ที่ network gateway ของเครือข่ายผู้รับบริการ เพื่อตรวจหาความผิดปกติหรือความเสี่ยงที่อาจก่อให้เกิดภัยคุกคาม ซึ่งผลลัพธ์ทีได้จากการประมวลผลของแต่ละเน็ตเวิร์คเซอร์วิส จะถูกนำมาแสดงผลในรูปแบบต่างๆ ที่ผู้ใช้สามารถเข้าใจได้ง่าย เช่น กราฟแสดงผล

ผ่าน dashboard รูปที่ 4.2.2 แสดงภาพรวมการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้องกับการวิเคราะห์และตรวจสอบภัยคุกคาม

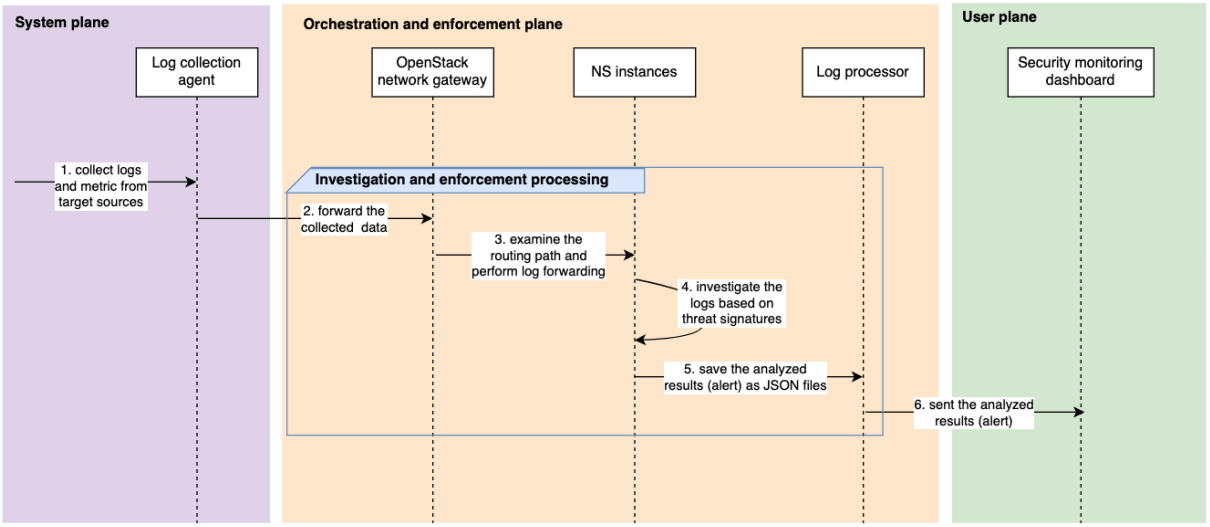



***รูปที่ 4.2.1:*** *แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้อง ซึ่งประกอบด้วย (1) Admin ผู้ที่รับผิดชอบในการออกแบบพิมพ์เขียว (network service templates) ด้านความปลอดภัยแต่ละประเภท ในที่นี้คือ Suricata, Snort และ Zeek (2) NFV orchestrator ทำหน้าที่รับและประมวลผลพิมพ์เขียวที่ได้ออกแบบ และอัพโหลดพิมพ์เขียวที่ถูกต้องไว้ใน repository กลาง (security service catalog) (3) Policy interpreter ทำหน้าที่ตีนโยบายความปลอดภัยที่ได้กำหนดหรือการตั้งค่าต่างๆ ที่เกี่ยวข้อง (4) Security service catalog ทำหน้าที่เสมือนแหล่งจัดเก็บกลาง (repository) ที่รวบรวมพิมพ์เขียวของแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกสร้างขึ้นและพร้อมใช้งาน (5) OpenStack cloud infrastructure provider รับผิดชอบในการจัดสรรทรัพยากรโครงสร้างพื้นฐาน ดังเช่น VM, CPU, RAM, ฯลฯ ที่จำเป็นเพื่อรอบรับการจำลองเน็ตเวิร์คเซอร์วิส*

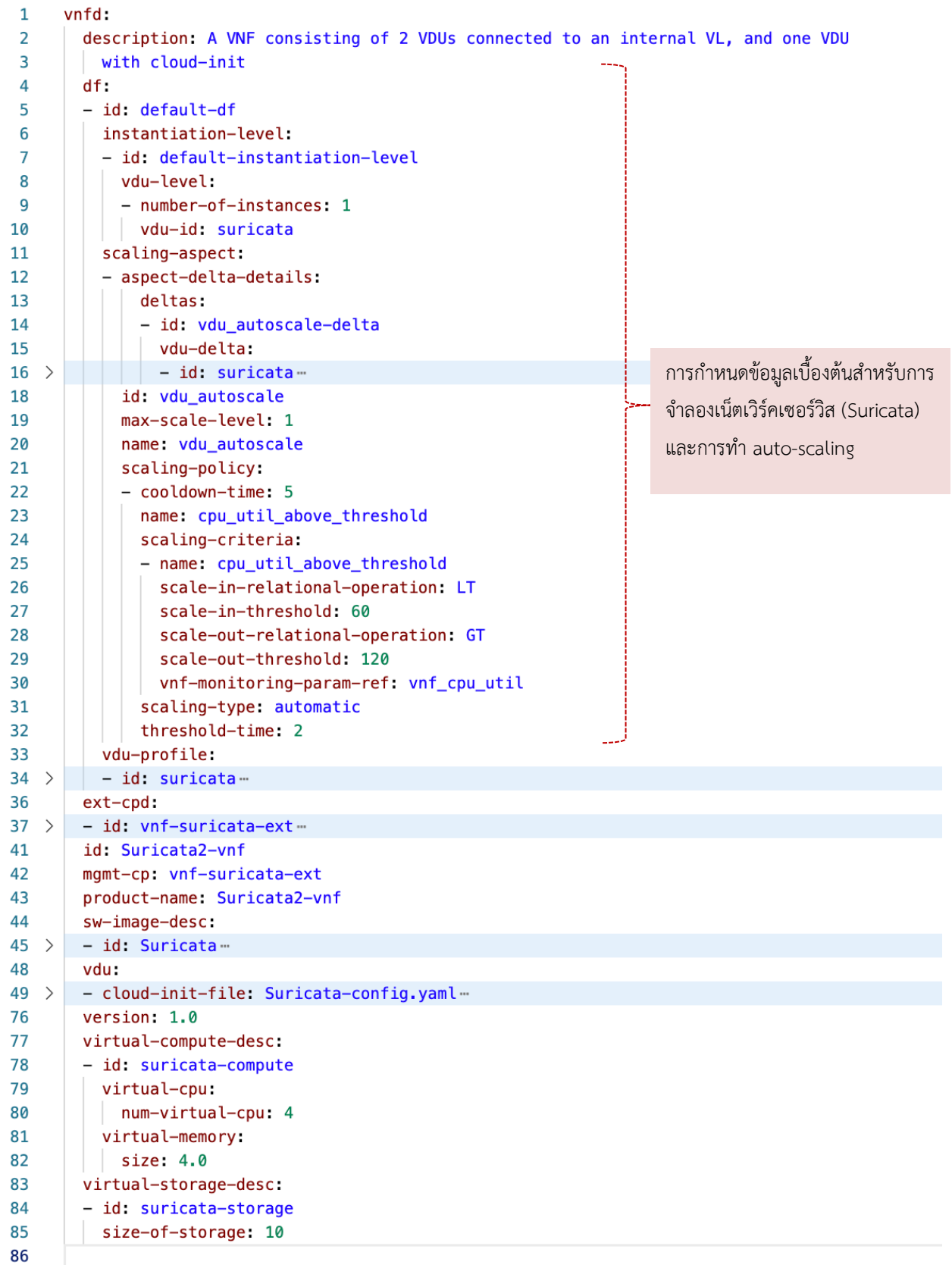
### 4.2.1 การออกแบบพิมพ์เขียว (Network service templates)

สถาปัตยกรรมแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงได้ถูกออกแบบมาเพื่อสนับสนุน model-driven environment รองรับมาตรฐาน ETSI NFV เพื่ออำนวยความสะดวกให้แก่ผู้ดูแลในการสร้างและออกแบบพิมพ์เขียว กำหนดการตั้งค่า และจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิสได้ตามที่ต้องการ ดังเช่น เน็ตเวิร์คเซอร์วิสด้านความปลอดภัย ได้แก่ Suricata, Snort และ Zeek โดยข้อดีของการสร้างแบบพิมพ์เขียวที่ขับเคลื่อนด้วยโมเดลนี้ (model-driven based network service templates) มีบทบาทสำคัญอย่างมากที่จะช่วยให้ผู้ดูแลสามารถจำลองและส่งมอบการให้บริการเน็ตเวิร์คเซอร์วิสประเภทใหม่ๆ ได้รวดเร็วยิ่งขึ้น มีความ

ยืดหยุ่นในการปรับปรุงการตั้งค่าต่างๆ และเพิ่มความสะดวกให้แก่ผู้ดูแลที่จะสามารถอ้างอิงพิมพ์เขียวเหล่านี้เพื่อประโยชน์ต่อการนำมาใช้ซ้ำ รูปที่ 4.2.3, 4.2.4, และ 4.2.5 แสดงตัวอย่างโมเดลการออกแบบพิมพ์เขียวสำหรับการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย ดังเช่น Suricata




***รูปที่ 4.2.2:*** *แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้องกับการตรวจสอบและวิเคราะห์ความเสี่ยงด้านภัยคุกคาม ซึ่งประกอบด้วย (1) Log collection agent ซึ่งได้ถูกติดตั้งที่ network gateway ของหน่วยงานผู้รับบริการเพื่อรวบรวมข้อมูล logs การทำงานต่างๆ (2) OpenStack network gateway ทำหน้าที่ในการจัดการเส้นทางและส่งต่อข้อมูล logs ที่ได้รับมาไปยังเน็ตเวิร์คเซอร์วิสด้านความที่กำหนด (3) NS instances คือเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ได้ถูกจำลองขึ้น ได้แก่ Suricata, Snort และ Zeek ซึ่งทำหน้าที่ในการประมวล วิเคราะห์ และตรวจหาความผิดปกติหรือความเสี่ยงที่ก่อให้เกิดภัยคุกคามจากข้อมูล logs (4) Log processor ทำหน้าที่รวบรวมข้อมูลและส่งผลลัพธ์ที่ได้จากการวิเคราะห์ไปแสดงผลผ่านเว็บ และ (5) Security monitoring dashboard ทำหน้าที่ในการนำผลลัพธ์มาจัดจำแนกและแสดงผลในรูปแบบกราฟรายงานต่างๆ เพื่อให้ง่ายต่อการทำความเข้าใจ*




*รูปที่ 4.2.3: แสดงการกำหนดข้อมูลเบื้องต้นสำหรับการจำลอง Suricata*

```yaml
vdu:
- cloud-init-file: Suricata-config.yaml
  id: suricata
  int-cpd:
  - id: suricata-eth0-int
    virtual-network-interface-requirement:
    - name: suricata-eth0
      position: 1
      virtual-interface:
        type: PARAVIRT
  monitoring-parameter:
  - id: vnf_cpu_util
    name: vnf_cpu_util
    performance-metric: cpu_utilization
  - id: vnf_memory_util
    name: vnf_memory_util
    performance-metric: average_memory_utilization
  - id: vnf_packets_sent
    name: vnf_packets_sent
    performance-metric: packets_sent
  - id: vnf_packets_received
    name: vnf_packets_received
    performance-metric: packets_received
  name: suricata
  sw-image-desc: Suricata
  virtual-compute-desc: suricata-compute
  virtual-storage-desc:
  - suricata-storage
version: 1.0
virtual-compute-desc:
- id: suricata-compute
  virtual-cpu:
    num-virtual-cpu: 4
  virtual-memory:
    size: 4.0
virtual-storage-desc:
- id: suricata-storage
  size-of-storage: 10
```

การตั้งค่า Suricata และสร้าง metrics สำหรับ monitoring

***รูปที่ 4.2.4:*** *แสดงการตั้งค่า Suricata และการสร้าง metrics สำหรับ monitoring โดย NFV orchestrator จะตรวจสอบโดยอ้างอิงค่า threshold ที่กำหนดในการปรับลดหรือขยายเน็ตเวิร์คเซอร์วิส (Suricata)*

*รูปที่ 4.2.5: แสดงการตั้งค่าเน็ตเวิร์คให้กับ Suricata*

### 4.2.2 ขั้นตอนการทำงาน

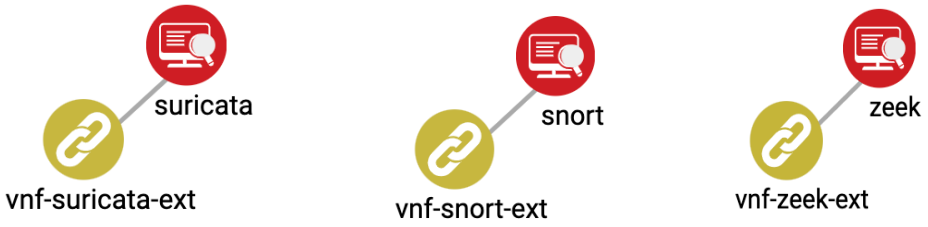
ขั้นตอนการทำงานจะแบ่งออกเป็น 3 ระยะหลักๆ ดังนี้

**ระยะที่ 1: การสร้างและออกแบบพิมพ์เขียว**

สำหรับการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย ดังเช่น Snort, Suricata และ Zeek จากรูปที่ 4.2.1 ผู้ดูแลระบบจะดำเนินการออกแบบพิมพ์เขียวและอัพโหลดพิมพ์เขียวที่ถูกต้องไว้ใน repository กลาง โดย

การติดตั้งและจำลองเน็ตเวิร์คเซอร์วิสดังกล่าวฯ บน โครงสร้างพื้นฐาน OpenStack cloud นั้น เกิดจากการบริหารและจัดการผ่าน NFV orchestrator ซึ่งแต่ละเน็ตเวิร์คเซอร์วิสมีการสร้างและออกแบบพิมพ์เขียวในรูปของ .yaml ไฟล์ โดยมีข้อมูล ดังนี้

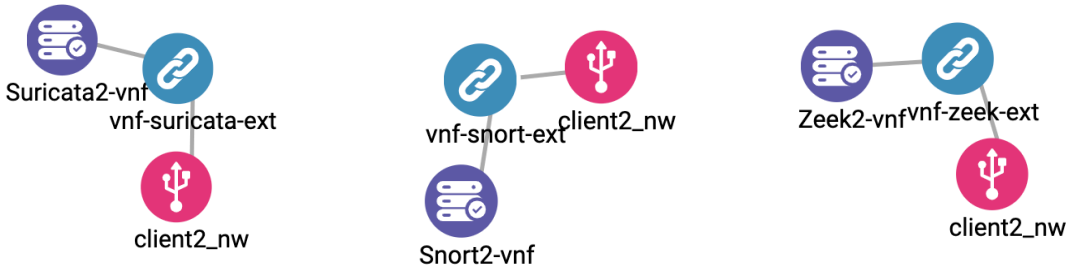
1. *VNF package:* คือ การกำหนดทรัพยาการโครงสร้างพื้นฐานที่จำเป็นให้กับเครื่อง VM ที่ต้องการสร้าง ได้แก่ การกำหนด CPU Core, RAM, Storage การทำ auto scaling และพอร์ตที่ใช้สำหรับเชื่อมต่อรวมไปถึง Cloud-init file สำหรับกำหนดค่าเริ่มต้นให้กับเครื่อง VM ที่ใช้สำหรับการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ เช่น Suricata, Snort และ Zeek รูปที่ 4.2.6 แสดงตัวอย่าง network topology ของแต่ละ VNF package สำหรับการจำลอง Suricata, Snort และ Zeek ตามลำดับ และรูปที่ 4.2.3 แสดงตัวอย่างข้อมูลทั้งหมดใน VNF package ที่ใช้สำหรับการจำลอง Suricata ซึ่งจากรูปจะเห็นได้ว่าในบรรทัดที่ 11-32 จะมีการกำหนดขีดความสามารถในการทำ auto scaling เอาไว้ด้วย ซึ่งเป็นฟังก์ชั่นที่เขียนเอาไว้เพื่อทำการปรับขยาย/ลดจำนวนเน็ตเวิร์คเซอร์วิสตามประสิทธิภาพของการใช้งาน หรือเมื่อประสิทธิภาพของทรัพยากรที่ใช้สำหรับการประมวลผลเกินกว่าค่า threshold ที่กำหนดไว้ ซึ่งคุณสมบัติดังกล่าวฯ ถือว่าเป็นจุดเด่นหนึ่งของแพลตฟอร์ฯ ในการจัดสรรและใช้งานทรัพยากรให้เกิดประโยชน์สูงสุด โดยมีการกำหนดข้อมูลสำคัญที่เกี่ยวข้องกับการทำ auto-scaling ไว้ดังนี้




*รูปที่ 4.2.6: ภาพแสดง network topology ของ Suricata, Snort และ Zeek โดย -ext แสดง external link connection*

   a. Scaling-policy จะมีการกำหนด cooldown-time ไว้ที่ 5 นาที เพื่อพักการทำ scaling หลังจากครั้งล่าสุด 5 นาที
   b. Scaling-criteria คือ การกำหนดเงื่อนไขในการทำ scaling และพารามิเตอร์ที่ใช้ในการกำหนดการ scale คือ cpu_util โดยเงื่อนไขของการ scale คือ เมื่อ cpu_util มีการใช้งานสูงกว่า 60% เกินเวลา Threshold-time ก็จะทำการ scale-out และเมื่อมีการใช้งาน cpu_util ต่ำกว่า 10% เกินเวลา Threshold-time ก็จะทำการ scale in
   c. Threshold-time คือ เวลาที่กำหนดสำหรับเงื่อนไขในการทำ scaling (นาที)
2. *NS package* คือ การสร้างการเชื่อมต่อระหว่างเครื่อง VM ที่ใช้สำหรับการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ กับวง Network ใน OpenStack ที่ได้เตรียมเอาไว้ รูปที่ 4.2.7 แสดงตัวอย่างการเชื่อมต่อเน็ตเวิร์คเซอร์วิส ได้แก่ Suricata, Snort และ Zeek เข้ากับวง Network ‘client2’

3. *Cloud-init file* คือ ไฟล์ที่ใช้สำหรับการตั้งค่าคอนฟิกให้กับเครื่อง VM ที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิส โดยข้อมูลการตั้งค่าจะมีความแตกต่างกันไปขึ้นอยู่กับเน็ตเวิร์คเซอร์วิสแต่ละประเภท




*รูปที่ 4.2.7: ภาพแสดง network topology ของ Suricata, Snort และ Zeek ซึ่งทั้งสามประเภทจะถูกเชื่อมต่อเข้ากับ Network วงเดียวกันคือ 'client2'*

**ระยะที่ 2: การตั้งค่าเน็ตเวิร์คเซอร์วิส**

เมื่อ OpenStack ได้รับข้อมูลจาก NFV Orchestrator สำหรับการสร้างเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ OpenStack จะอ้างอิงข้อมูลที่ระบุไว้ใน VNF package และ NS package โดยขั้นตอนแรกเป็นการสร้างเครื่อง VM ขึ้นมา โดยใช้ image ที่มีการระบุไว้ใน VNF Package เพื่อใช้เป็นระบบปฏิบัติการสำหรับการสร้างเครื่อง VM โดยแต่ละ image จะมีการติดตั้งค่าเริ่มต้น (pre-configuration) สำหรับการจำลองเน็ตเวิร์คเซอร์วิสและแพ็คเกจที่จำเป็นไว้ล่วงหน้า รูปที่ 4.2.5 แสดงตัวอย่างการตั้งค่าที่แตกต่างกันไปในแต่ละเน็ตเวิร์คเซอร์วิสที่ต้องการผ่าน Cloud-init file (.yaml file) ข้อมูลด้านล่าง ดังนี้

1. การตั้งค่า Suricata service
   1.1 การตั้งค่า interface: ในขั้นตอนแรกหลังจากการติดตั้ง Suricata service จะต้องมีการตั้งค่าพอร์ตที่ใช้สำหรับการฟังข้อมูลทราฟฟิก ซึ่งสามารถเข้าไปตั้งค่าได้ที่ /etc/suricata/suricata.yaml และไปยังบรรทัด af-packet เพื่อตั้งค่า interface ที่ต้องการ (แสดงดังรูปที่ 4.2.8)
   1.2 การตั้งค่า rule: โดยปกติแล้ว Suricata จะมี default_rule มาให้อยู่ถ้าเกิดต้องการที่จะเพิ่ม rule ที่ต้องการก็สามารถเข้าไปยัง /var/lib/suricata/rules/ เพื่อสร้าง file rule เพิ่มเติม และนำมาเพิ่ม rule-file ในไฟล์ suricata.yaml (แสดงดังรูปที่ 4.2.9)
   1.3 Log file: ผลการตรวจสอบภัยคุกคามตาม rules ที่กำหนด จะถูกจัดเก็บใน log file โดยบันทึกอยู่ใน /var/log/suricata/ ซึ่งจะมีไฟล์ที่สำคัญหลักๆ คือ fast.log จะเป็นไฟล์ที่แสดงข้อความ alert ที่มาจากการกำหนดค่า rule ต่างๆ เอาไว้ก่อนหน้า และ eve.json เป็นไฟล์ จะเป็นไฟล์ที่เก็บข้อมูล log แบบละเอียดในรูปของ .json file เพื่อที่จะนำไปใช้กับ fluent-bit ต่อไป
   1.4 การเปิดใช้งาน: สามารถตรวจสอบการเปิดใช้งานได้จากการใช้คำสั่ง sudo systemctl status suricate.service รูปที่ 4.2.10 แสดงสถานการณ์เปิดใช้งาน ถ้าข้อความแสดงผลเป็น active แสดงว่า Service ได้ถูกเปิดใช้งานเรียบร้อยแล้ว

2. การตั้งค่า Snort service

2.1 การตั้งค่า log: เพื่อให้ log ที่ออกมาเป็น json file จึงต้องเข้ามาตั้งค่าในตัวไฟล์ของ Snort service ซึ่งบันทึกอยู่ใน /usr/local/etc/snort/snort.lua โดยจะอยู่ในส่วนของ configuration output โดยสามารถเพิ่ม fields ที่ต้องการสำหรับวิเคราะห์ได้ และ log บันทึกอยู่ใน /var/log/snort/alert_json.txt (แสดงดังรูปที่ 4.2.11)

2.2 การติดตั้ง rule: Snort3 community มี rule public ให้จึงสามารถโหลดมาแล้วบันทึกไว้ใน /usr/local/etc/rules/ เพื่อนำไปเรียกใช้งานต่อไป

```
af-packet:
  - interface: dummy0
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    #  * cluster_flow: all packets of a given flow are sent to the same socket
    #  * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    #  * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #  socket. Requires at least Linux 3.14.
    #  * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    #  more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
    # cluster_rollover has been deprecated; if used, it'll be replaced with cluster_flow.
    cluster-type: cluster_flow
    # In some fragmentation cases, the hash can not be computed. If "defrag" is set
    # to yes, the kernel will do the needed defragmentation before sending the packets.
    defrag: yes
    # To use the ring feature of AF_PACKET, set 'use-mmap' to yes
    use-mmap: yes
```

*รูปที่ 4.2.8: ตัวอย่างการตั้งค่าพอร์ตที่จะใช้สำหรับการฟังข้อมูล traffic*

```
default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
  - /var/lib/suricata/rules/local.rules
```

*รูปที่ 4.2.9: ตัวอย่างการเพิ่ม rules ใหม่และกำหนด path ไปยัง default rule*

```
ubuntu@suricata2:~$ sudo systemctl status suricata.service
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: active (running) since Thu 2024-07-04 09:06:47 UTC; 8min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 10 (limit: 4645)
     Memory: 225.3M
        CPU: 3min 15.445s
     CGroup: /system.slice/suricata.service
             └─1100 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid --af-packet -D -vv

Jul 04 09:06:46 suricata2 systemd[1]: Starting LSB: Next Generation IDS/IPS...
Jul 04 09:06:47 suricata2 suricata[1076]: Starting suricata in IDS (af-packet) mode... done.
Jul 04 09:06:47 suricata2 systemd[1]: Started LSB: Next Generation IDS/IPS.
```

*รูปที่ 4.2.10: แสดงสถานะการเปิดใช้งาน Suricata service*

2.3 การเรียกใช้งาน: สามารถใช้คำสั่ง sudo snort -c /usr/local/etc/snort.lua -R /usr/local/etc/rules/(rule file) -l /var/log/snort/ -i dummy0

โดย /usr/local/etc/snort.lua คือไฟล์ config ของตัว snort

/usr/local/etc/rules/(rule file) คือ path rule ที่เราต้องการเรียกใช้

-l /var/log/snort/ คือ path ที่เราจะบันทึก log file

3. การตั้งค่า Zeek service

3.1 การตั้งค่า log: โดยแรกเริ่มนั้น Zeek ไม่ได้ให้ผล log ออกมาในรูป json file จึงต้องมีการตั้งค่า log เพื่อให้ผลลัพธ์ออกมาอยู่ในรูป json (แสดงดังรูปที่ 4.2.12)

3.2 การตั้งค่า interface: ตั้งค่าพอร์ตที่ใช้สำหรับฟังข้อมูลทราฟฟิค โดยสามารถแก้ไขในไฟล์ /opt/zeek/etc/node.cfg และแก้ไขในส่วนที่เป็น interface (แสดงดังรูปที่ 4.2.13)

3.3 การเริ่มใช้งาน: หลังจากการตั้งค่าทุกอย่างเรียบร้อยแล้วจึงใช้คำสั่ง sudo zeekctl deploy เพื่อเริ่มการใช้งาน และสามารถตรวจสอบสถานะโดยใช้คำสั่ง sudo zeekctl status

```
-- 7. configure outputs
--------------------------------------------------------------------------
alert_json =
{
        fields = 'timestamp seconds action class b64_data dir dst_addr dst_port eth_dst eth_len eth_src eth_type gid icmp_code icmp>
        file = true,
        limit = 100,
}

appid_listener =
{
    json_logging = true,
    file = '/var/log/snort/appid-output.json',
}
```

*รูปที่ 4.2.11: แสดงตัวอย่างการตั้งค่า log file บน Snort service*

```
echo "redef LogAscii::json_timestamps = JSON::TS_ISO8601;" | sudo tee -a /opt/zeek/share/zeek/policy/tuning/json-logs.zeek
echo "redef LogAscii::json_include_unset_fields=T;" | sudo tee -a /opt/zeek/share/zeek/policy/tuning/json-logs.zeek
echo "@load policy/tuning/json-logs.zeek" | sudo tee -a /opt/zeek/share/zeek/site/local.zeek
```

*รูปที่ 4.2.12: แสดงตัวอย่างการตั้งค่า log file บน Zeek*

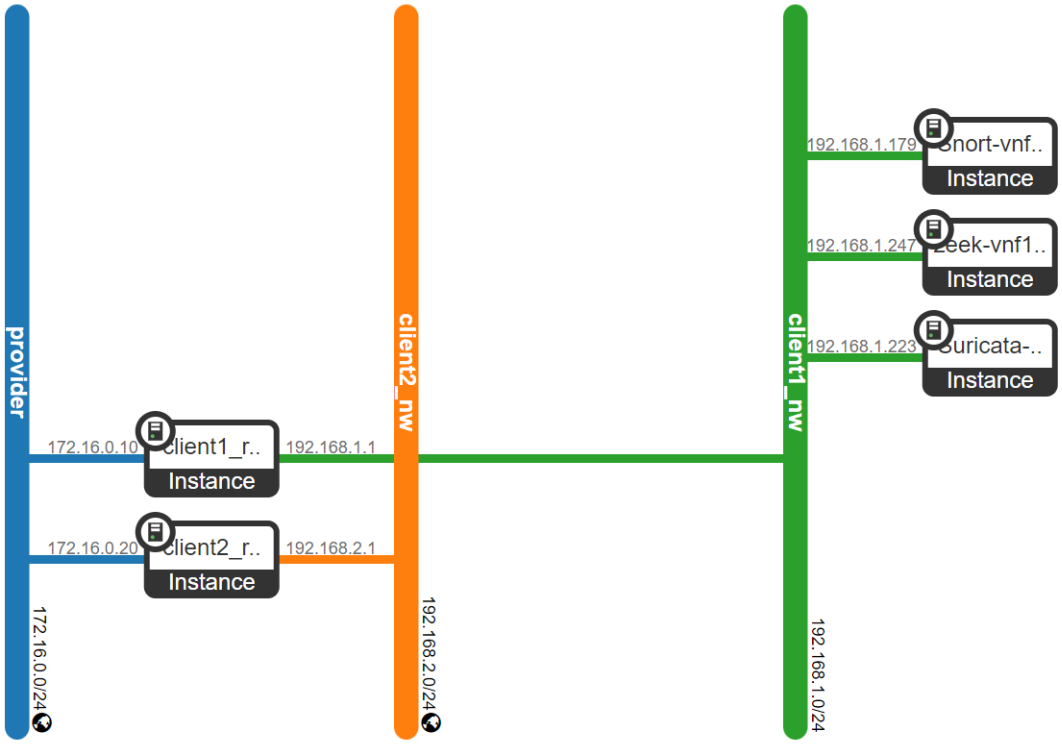
```
  GNU nano 6.2                                    /opt/zeek/etc/node.cfg *
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=dummy0
```

*รูปที่ 4.2.13: แสดงตัวอย่างการตั้งค่า interface บน Zeek service*

เมื่อขั้นตอนการจำลองครื่อง VM และการตั้งค่าเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแต่ละประเภทเสร็จสิ้น ผลลัพธ์การสร้างโทโพโลยีเครือข่ายบน OpenStack (แสดงดังรูปที่ 4.2.14) ซึ่งประกอบด้วย 3 ส่วนหลักๆ ดังนี้

- Provider ทำหน้าที่เชื่อมต่อและติดต่อสื่อสารกับเน็ตเวิร์คภายนอก โดยจะทำหน้าที่รับและแปลง traffic ip ที่วิ่งเข้ามาจากภายนอกจาก public ip และเปลี่ยนเป็น private ip
- client router คือ router ที่ทำหน้าที่คอยจัดการเส้นทางให้กับข้อมูลที่ผ่านเข้าออกภายในวงเน็ตเวิร์ค และทำการตัดสินใจว่าจะส่งทราฟฟิกข้อมูลดังกล่าวฯ ไปยังเครือข่ายภายในวงใด
- client_nw คือ การจำลองสร้างเครือข่ายภายใน (internal network) ให้แต่ละกลุ่มลูกค้า โดย internal network แต่ละวงจะมีการจำลองและเชื่อมต่อกับเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแต่ละประเภท ได้แก่ Suricata, Snort, และ Zeek สำหรับรองรับการให้บริการลูกค้าแต่ละกลุ่ม โดยข้อดีของการออกแบบ internal network แยกออกจากกัน โดยแต่ละวงจะมีการติดตั้งและจำลองเน็ตเวิร์คเซอร์วิสฯ แยกตามกลุ่มลูกค้าแต่ละกลุ่ม จะช่วยเพิ่มประสิทธิภาพในการประมวลผลและวิเคราะห์ภัยคุกคามได้เร็วขึ้น ช่วยแก้ปัญหาคอขวดในกระบวนการทำงาน ทำให้ระบบสามารถรองรับการทำงานที่มีปริมาณงานเป็นจำนวนมากได้เป็นอย่างดี

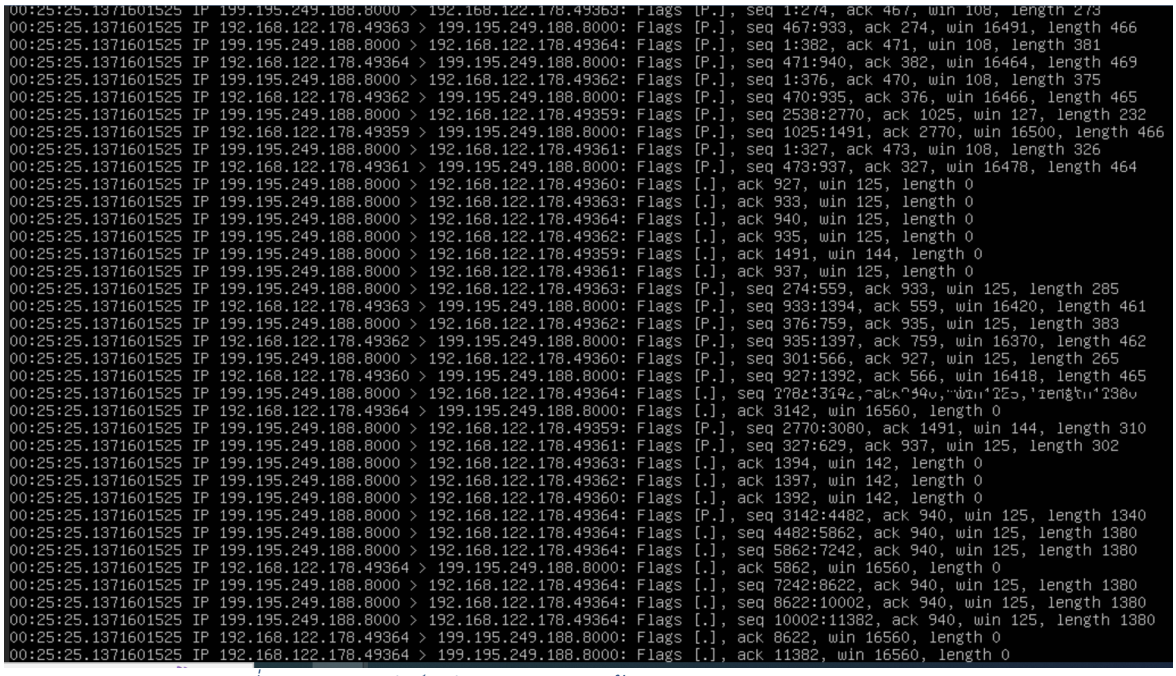



*รูปที่ 4.2.14: โครงสร้างการเชื่อมต่อเครือข่าย (network topology) ของแต่ละเน็ตเวิร์คเซอร์วิสบน OpenStack*

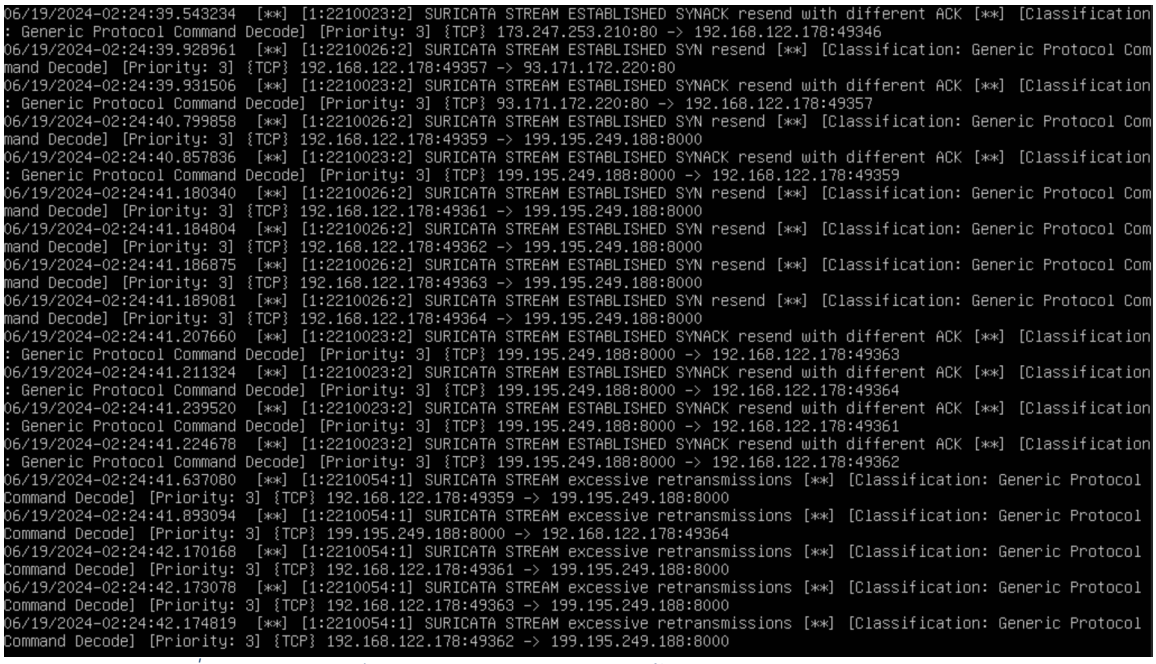
**ระยะที่ 3: การทดสอบใช้งาน**

ขั้นตอนต่อมาคือการตรวจสอบการใช้งานเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแต่ละประเภทที่ได้จำลองขึ้น ได้แก่ Suricata, Snort และ Zeek เพื่อแสดงให้เห็นถึงความสามารถในการใช้งาน โดยข้อมูล logs ที่นำมาใช้ในการทดสอบอ้างอิงจาก Public Dataset [60]

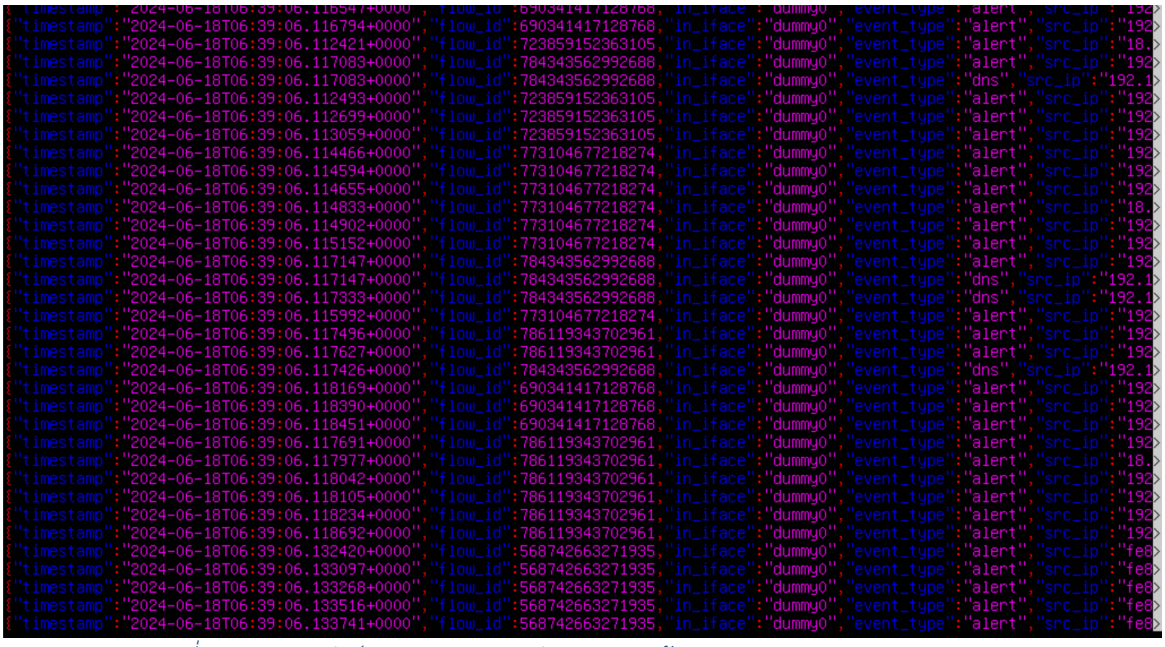
1. Suricata service: สำหรับแนวทางการทดสอบใช้งานเราจะใช้ คำสั่ง sudo tcpreplay-edit -i dummy0 -v 2013-06-18-Neutrino-EK-traffic.pcap เพื่อทำการจำลองทราฟฟิกที่พอร์ต duumy0 และตรวจเช็คผลการแจ้งเตือนผ่าน fast.log และ eve.json โดยเมื่อใช้งานคำสั่ง ผลลัพธ์ที่ได้จะแสดงดังรูปที่ 4.2.15, 4.2.16 และ 4.2.17 ตามลำดับ
2. Snort service: สำหรับการทดสอบเราจะใช้ คำสั่ง sudo tcpreplay-edit -i dummy0 -v 2013-06-18-Neutrino-EK-traffic.pcap เพื่อทำการจำลองทราฟฟิกที่พอร์ต duumy0 และตรวจเช็คผลการแจ้งเตือนผ่าน alert_json.txt โดยเมื่อใช้งานคำสั่งจะได้ผลลัพธ์แสดงดังรูปที่ 4.2.18
3. Zeek service: ในการทดสอบเราจะใช้ คำสั่ง sudo tcpreplay-edit -i dummy0 -v 2013-06-18-Neutrino-EK-traffic.pcap เพื่อทำการจำลองทราฟฟิกที่พอร์ต duumy0 และตรวจเช็คผลการแจ้งเตือนผ่าน log file โดยเมื่อใช้งานคำสั่งจะได้ผลลัพธ์แสดงดังรูปที่ 4.2.19 และ 4.2.20



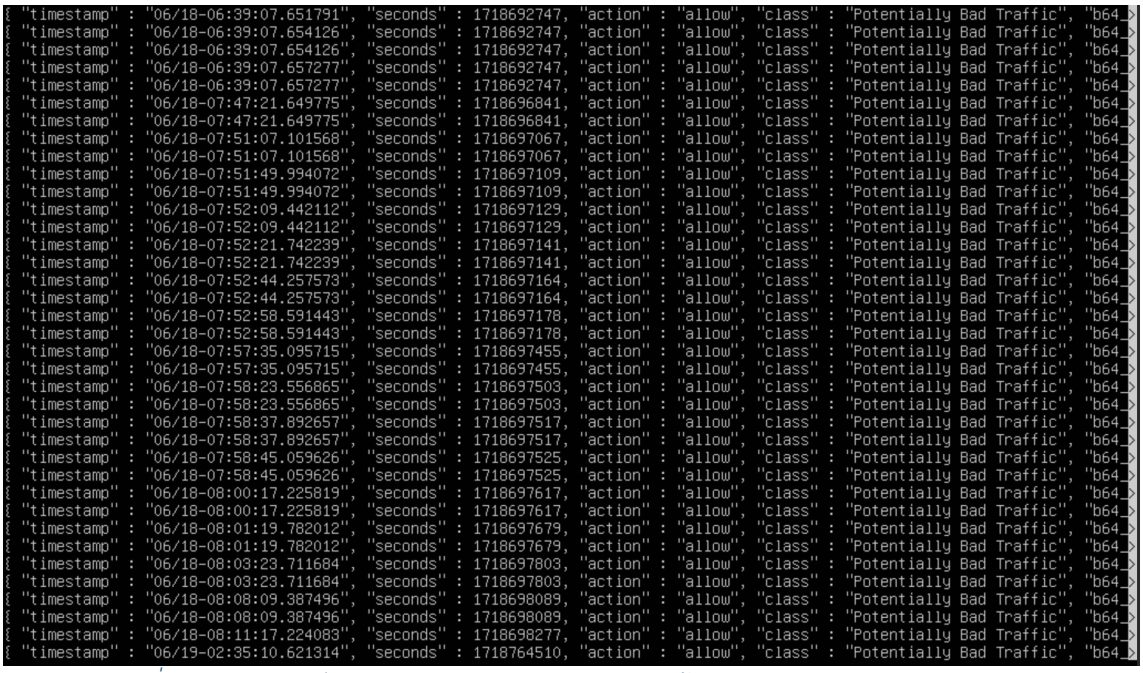

***รูปที่ 4.2.15:*** *ผลลัพธ์หลังจากการจำลองข้อมูลทราฟฟิกจาก public dataset*



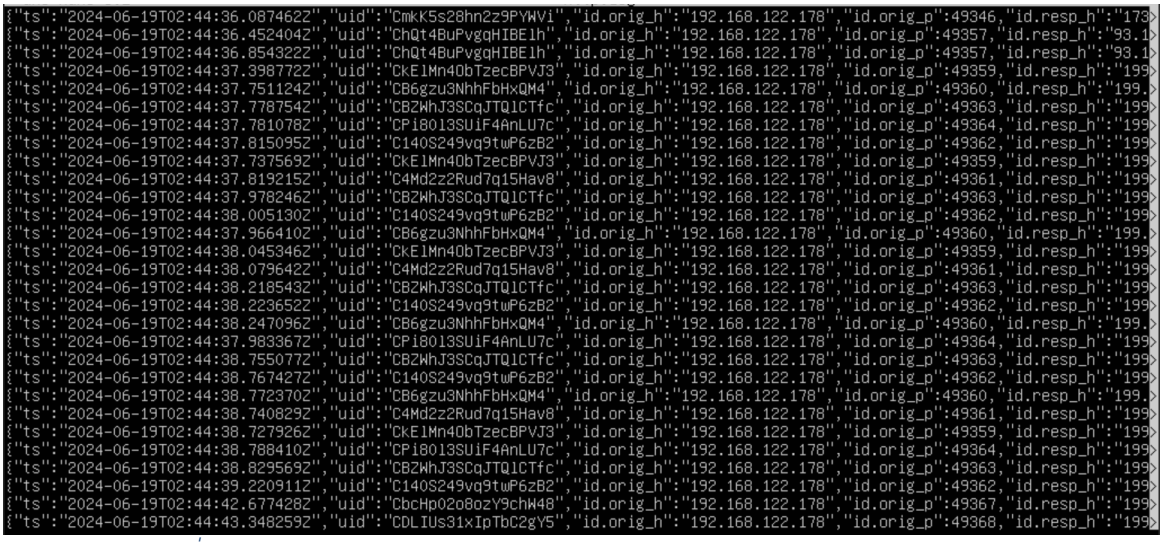

***รูปที่ 4.2.16:*** *ผลลัพธ์ของ fast.log หลังการจำลองข้อมูลทราฟฟิกจาก public dataset*




***รูปที่ 4.2.17:*** *ผลลัพธ์ของ eve.json หลังจากจำลองข้อมูลทราฟฟิกจาก public dataset*




***รูปที่ 4.2.18:*** *ผลลัพธ์ของ alert_json.txt หลังการจำลองข้อมูลทราฟฟิกจาก public dataset*




***รูปที่ 4.2.19:*** *ผลลัพธ์ของ http.log หลังจากจำลองข้อมูลทราฟฟิกจาก public dataset*

***รูปที่ 4.2.20:*** *ไฟล์ log หลังจากการจำลองข้อมูลทราฟฟิกจาก public dataset*

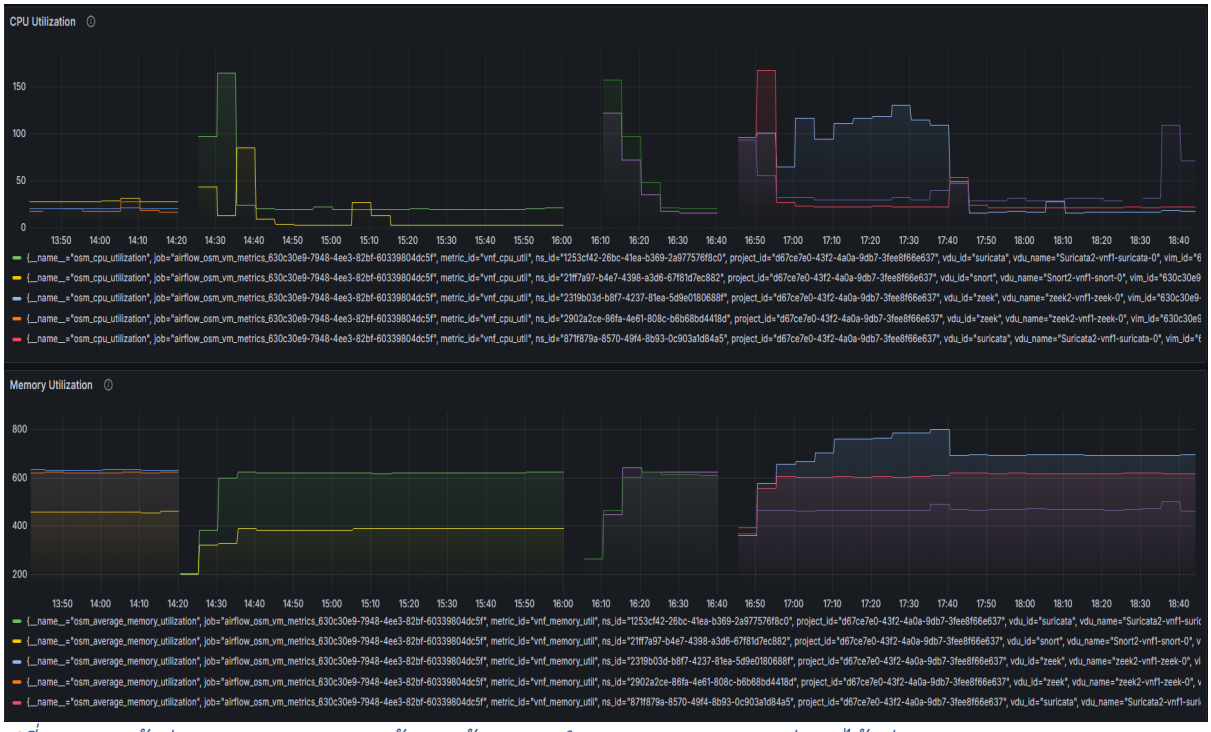
## 4.3 ด้านการใช้งาน

จากขั้นตอนการตั้งค่าต่างๆ ที่เกี่ยวข้องในระบบหลังบ้าน (ดังที่กล่าวถึงในหัวข้อที่ 5.2) ในหัวข้อนี้จะพูดถึงรายละเอียดการใช้งานและส่วนแสดงผลของระบบ ซึ่งข้อมูลที่นำมาแสดงสามารถแบ่งออกได้เป็น 3 กลุ่มข้อมูลหลักๆ ได้ดังนี้

### 4.3.1 สถานะการทำงานของระบบ

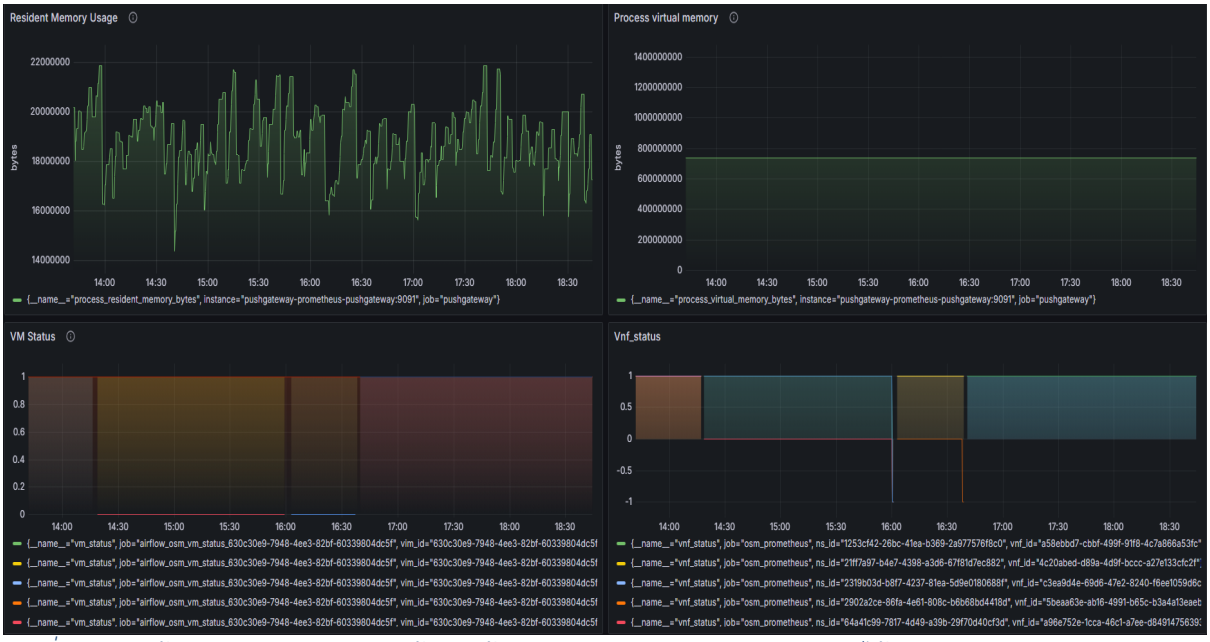
สถานะการทำงานของระบบจะเป็นข้อมูลที่เกี่ยวข้องกับสถานะทำงานของระบบหลังบ้าน ซึ่งโดยผู้ดูแลระบบสามารถติดตามและตรวจสอบสถานะการทำงานของ instances ต่างๆ ได้ดังนี้ (แสดงดังรูปที่ 4.3.1 และ 4.3.2)

- *CPU utilization* แสดงถึงการใช้งานของ CPU โดยจะแสดงให้เห็นว่าทรัพยากรการประมวลผลของแต่ละ instance ในระบบนั้นถูกใช้งานไปเท่าไหร่ ผู้ดูแลสามารถติดตามและเห็นการใช้งาน CPU ว่ามีการใช้งานมากน้อยเพียงใดในแต่ละช่วงเวลา
- *Memory utilization* แสดงถึงการใช้งานของ RAM โดยจะแสดงให้เห็นว่าทรัพยากรหน่วยความจำของแต่ละ instance ในระบบนั้นถูกใช้งานไปเท่าไหร่ เพื่อติดตามว่าหน่วยความจำถูกใช้มากน้อยเพียงใดในแต่ละช่วงเวลา
- *Resident memory usage* แสดงถึงการใช้งานหน่วยความจำจริงใน RAM ของ process แต่ละตัวที่กำลังทำงานอยู่ และสามารถใช้ข้อมูลนี้ในการตรวจสอบว่ามีการใช้งานหน่วยความจำมากเกินไปหรือไม่
- Process virtual memory แสดงถึงการใช้งานหน่วยความจำเสมือน (Virtual Memory) ของ process แต่ละตัวที่กำลังทำงานอยู่
- VM status แสดงถึงสถานะการทำงานของ Virtual Machines (VMs) แต่ละตัว โดย status = 1 หมายถึงกำลังทำงานอย่างถูกต้องหรือปกติ status = 0 หมายถึงสถานะที่ไม่มีการกำหนดหรือสถานะที่ไม่มีค่าหรือสถานะที่ไม่มีการเปลี่ยนแปลงและ status = -1 หมายถึงมีปัญหาหรือผิดปกติในการทำงาน เช่น การเชื่อมต่อที่ขัดข้อง หรือเกิดข้อผิดพลาดขึ้น




***รูปที่ 4.3.1:*** *หน้าต่าง dashboard แสดงข้อมูลแจ้งสถานะทำงานของ instances ต่างๆ ได้แก่ CPU utilization และ memory utilization*

- VNF status แสดงถึงสถานะการทำงานของ Virtual Network Function (VNFs) แต่ละตัว โดย status = 1 หมายถึงกำลังทำงานอย่างถูกต้องหรือปกติ status = 0 หมายถึงสถานะที่ไม่มีการกำหนดหรือสถานะที่ไม่มีค่าหรือสถานะที่ไม่มีการเปลี่ยนแปลงและ status = -1 หมายถึงมีปัญหาหรือผิดปกติในการทำงาน เช่น การเชื่อมต่อที่ขัดข้อง หรือเกิดข้อผิดพลาดขึ้น




*รูปที่ 4.3.2: หน้าต่าง dashboard แสดงข้อมูลแจ้งสถานะทำงานของ instances ต่างๆ ได้แก่ Resident memory usage, Process virtual memory, VM status และ VNF status*

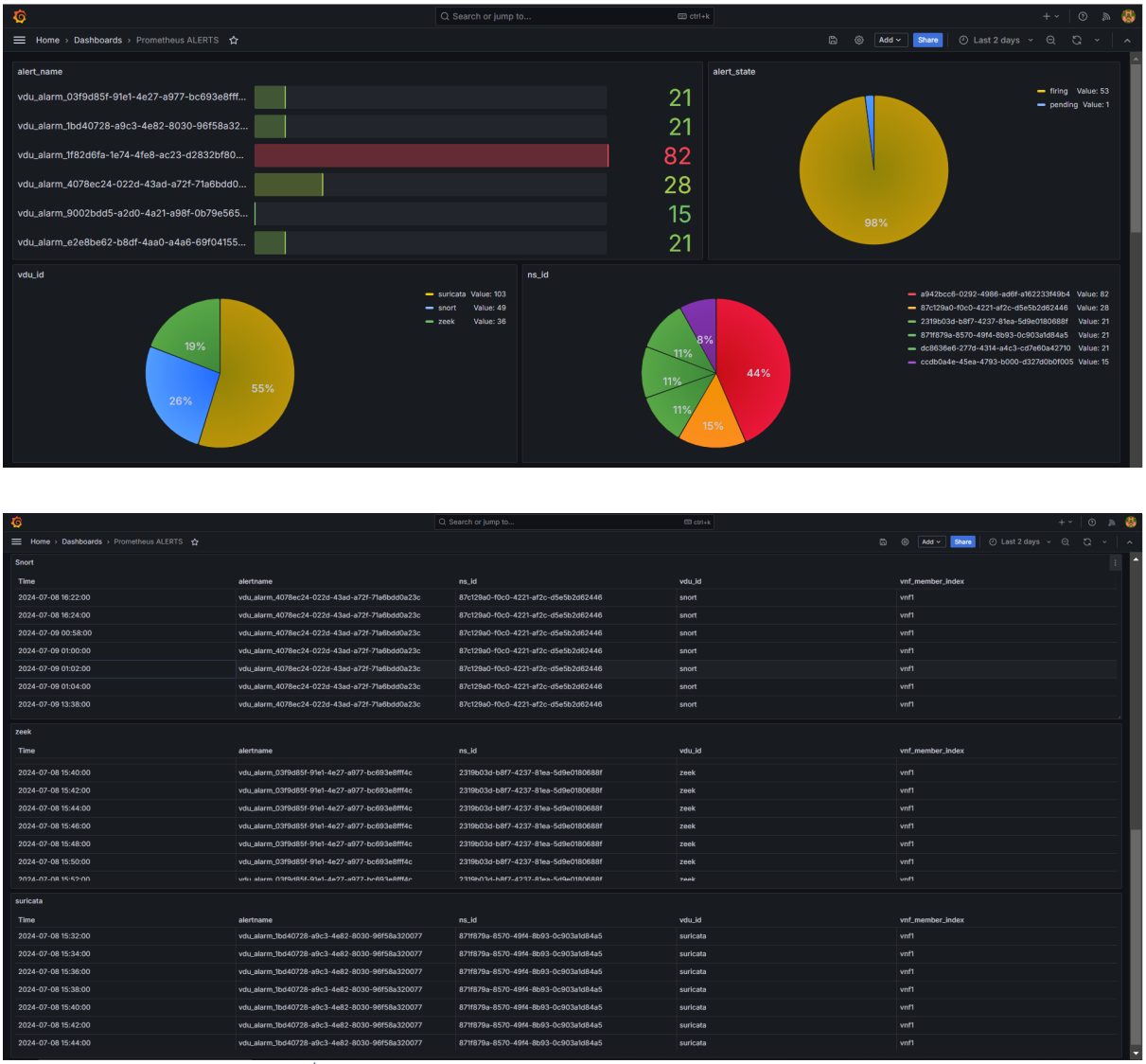
### 4.3.2 สถานะการการแจ้งเตือนของระบบ

นอกเหนือจากขีดความสามารถของระบบในการแสดงข้อมูลการทำงานของ instances ต่างๆ (ดังที่ได้กล่าวมาข้างต้น) ระบบยังเพิ่มขีดความสามารถในการตรวจสอบ ติดตามประสิทธิภาพการดำเนินงาน และแจ้งเตือน (alerts) แก่ผู้ดูแลหากพบว่ามีความผิดปกติภายในระบบเกิดขึ้น โดยสามารถอธิบายรายละเอียดของแต่ละ panel ได้ดังนี้ (แสดงดังรูปที่ 4.3.3)

- Time แสดงวันที่และเวลาที่ข้อมูลหรือเหตุการณ์ที่เกิดการแจ้งเตือน (Alerts) ที่เกิดขึ้น เพื่อใช้ติดตามเวลาที่แน่นอนของเหตุการณ์ที่เกิดขึ้น
- Alert Name แสดงถึงชื่อของการแจ้งเตือนที่เกิดขึ้นในระบบ เช่น vdu_alarm_1f82d6fa-1e74-4fe8-ac23-d2832bf80b65
- Alert State แสดงถึงสถานะของการแจ้งเตือนที่กำลังเกิดขึ้น เช่น Firing, Pending
    - Pending หมายถึง เมื่อเงื่อนไขการแจ้งเตือนถูกทริกเกอร์และถึงค่า threshold ที่กำหนด แต่ยังไม่ถึงเวลาที่กำหนดเพื่อส่งการแจ้งเตือน และถ้าเงื่อนไขยังคงถูก ทริกเกอร์อยู่ในระยะเวลาที่กำหนด จะเปลี่ยนสถานะเป็น Firing

- Firing หมายถึง เมื่อเงื่อนไขการแจ้งเตือนยังคงถูกทริกเกอร์อยู่และระยะเวลาที่กำหนดหมดลง Alert manager จะทำการส่งการแจ้งเตือนไปยังระบบหรือผู้ใช้งานตามที่กำหนด

- VDU Id แสดงถึงของ Virtual Deployment Unit (VDU) ที่เกี่ยวข้องกับการแจ้งเตือน ในที่นี้ตัวอย่างเช่น Snort, Zeek, และ Suricata
- NS Id แสดงถึงรหัสของ Network Service (NS) ที่เกี่ยวข้องกับการแจ้งเตือน
- VNF Member Index แสดง VNF member ที่เกี่ยวข้อง







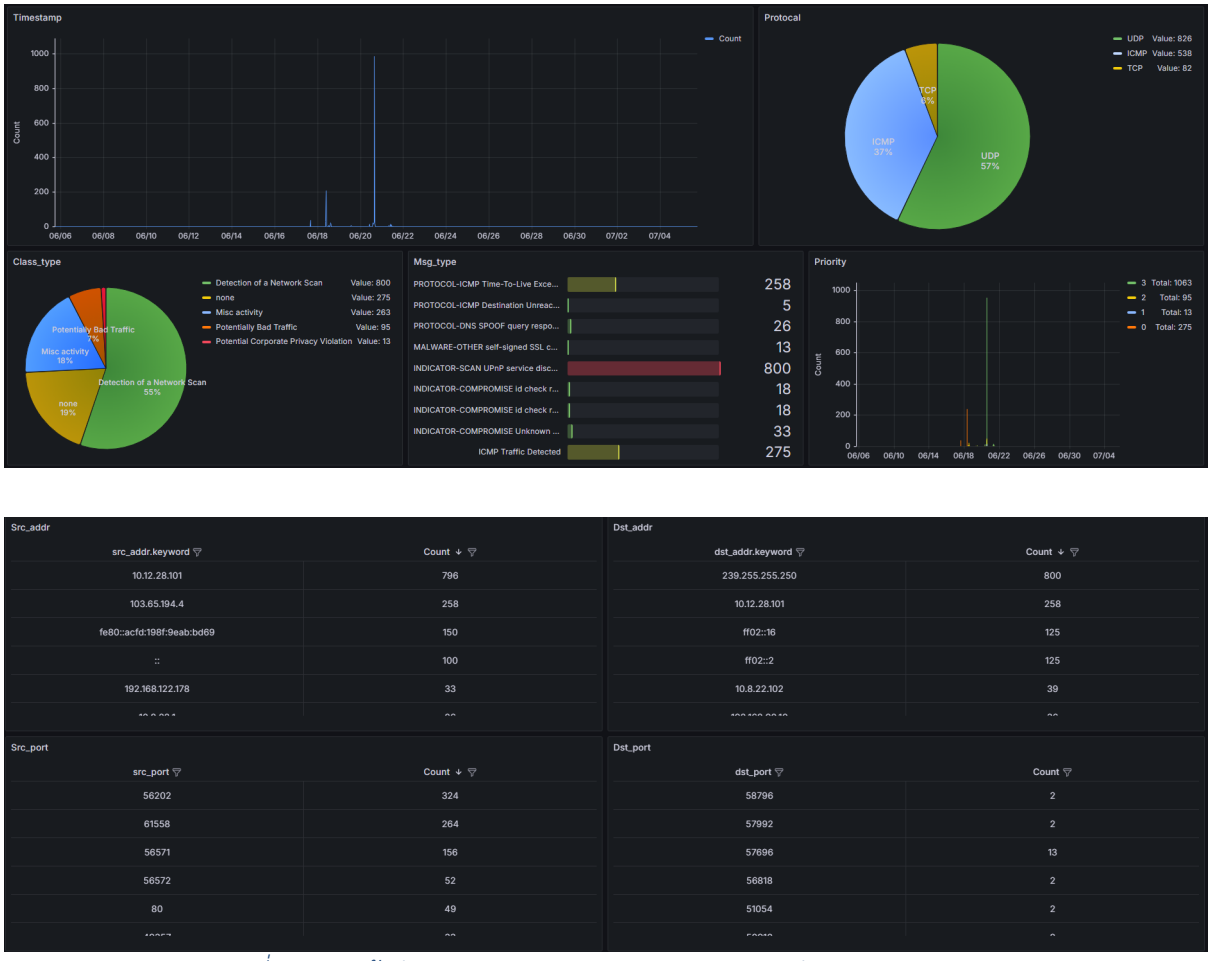
*รูปที่ 4.3.3: หน้าต่าง dashboard แสดงผลการแจ้งเตือน (alerts)*

### 4.3.3 ผลการวิเคราะห์ภัยคุกคาม

ข้อมูลที่เกี่ยวข้องกับผลการวิเคราะห์และตรวจสอบภัยคุกคามที่มาจากเน็ตเวิร์คเซอร์วิสที่ได้จำลองขึ้น ในที่นี้ได้แก่ Suricata, Snort และ Zeek

1. **Snort:** ผลการวิเคราะห์ข้อมูลภัยคุกคามจาก Snort service ประกอบด้วยข้อมูลที่สำคัญดังต่อนี้ (แสดงดังรูปที่ 4.3.4)

- Timestamp แสดงวันที่และเวลาที่ข้อมูลหรือเหตุการณ์ถูกบันทึก เพื่อใช้ติดตามเวลาที่แน่นอนของเหตุการณ์ที่เกิดขึ้น
- Protocol แสดงประเภทของโปรโตคอลที่ถูกใช้ในการสื่อสารเครือข่าย เช่น UDP, TCP, ICMP
- Class Type แสดงประเภทของเหตุการณ์หรือภัยคุกคามที่ถูกตรวจพบ ช่วยในการจำแนกและจัดหมวดหมู่เหตุการณ์ที่เกิดขึ้น
- Message Type แสดงข้อความที่อธิบายถึงเหตุการณ์หรือภัยคุกคามที่ถูกตรวจพบ
- Priority แสดงลำดับความสำคัญของเหตุการณ์หรือภัยคุกคาม โดยการจำแนกประเภทของ Snort ใช้ค่าความสำคัญ 1-4 โดย 1 คือระดับรุนแรงที่สุด และ 4 คือระดับรุนแรงน้อยที่สุด
- Source IP Address แสดงที่อยู่ IP ต้นทาง
- Destination IP Address แสดงที่อยู่ IP ปลายทาง
- Source Port แสดงพอร์ตที่อยู่ต้นทาง
- Destination Port แสดงพอร์ตที่อยู่ปลายทาง




*รูปที่ 4.3.4: หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Snort*

2. **Suricata:** ผลการวิเคราะห์ข้อมูลภัยคุกคามจาก Suricata service ประกอบด้วยข้อมูลที่สำคัญดังต่อนี้ (แสดงดังรูปที่ 4.3.5)
    - Timestamp แสดงวันที่และเวลาที่ข้อมูลหรือเหตุการณ์ถูกบันทึก เพื่อใช้ติดตามเวลาที่แน่นอนของเหตุการณ์ที่เกิดขึ้น
    - Protocol แสดงประเภทของโปรโตคอลที่ถูกใช้ในการสื่อสารเครือข่าย เช่น UDP, TCP, ICMP
    - Application Protocol แสดงประเภทของโปรโตคอลของแอปพลิเคชันที่เกี่ยวข้องในเหตุการณ์ เช่น HTTP, DNS
    - Alert Severity แสดงถึงลำดับความสำคัญของเหตุการณ์ Suricata มีระดับความรุนแรง 3 ระดับ คือ 1, 2 และ 3 โดย 3 เป็นระดับต่ำสุด
    - Signature แสดงถึงกฎที่ใช้ตรวจจับเหตุการณ์หรือภัยคุกคามที่ถูกตรวจพบ ทำให้รู้ถึงลักษณะของภัยคุกคาม
    - Source IP Address แสดงที่อยู่ IP ต้นทาง
    - Destination IP Address แสดงที่อยู่ IP ปลายทาง
    - Source Port แสดงพอร์ตที่อยู่ต้นทาง
    - Destination Port แสดงพอร์ตที่อยู่ปลายทาง
    - Direction แสดงถึงทิศทางของเครือข่าย
    - Action แสดงถึงการดำเนินการกับแพ็กเก็ตหลังจากตรวจพบ เช่น pass, drop, reject, alert
    - Category แสดงหมวดหมู่ของเหตุการณ์หรือภัยคุกคามที่ถูกตรวจพบ

***รูปที่ 4.3.5:*** *หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Suricata*

3. **Zeek:** ผลการวิเคราะห์ข้อมูลภัยคุกคามจาก Zeek service ประกอบด้วยข้อมูลที่สำคัญดังต่อนี้ (แสดงดังรูปที่ 4.3.6)
   - Timestamp แสดงวันที่และเวลาที่ข้อมูลหรือเหตุการณ์ถูกบันทึก เพื่อใช้ติดตามเวลาที่แน่นอนของเหตุการณ์ที่เกิดขึ้น
   - Message Error แสดงข้อความข้อผิดพลาดที่เกี่ยวกับเหตุการณ์ที่เกิดขึ้น
   - DNS Host แสดงชื่อโฮสต์ที่เกี่ยวข้อง เพื่อช่วยในวิเคราะห์การทำงานของ DNS ในเครือข่าย
   - Protocol แสดงประเภทของโปรโตคอลที่ถูกใช้ในการสื่อสารเครือข่าย เช่น UDP, TCP, ICMP
   - Application Service แสดงประเภทของโปรโตคอลของแอปพลิเคชันที่เกี่ยวข้องในเหตุการณ์ เช่น HTTP, DNS
   - MD5 Hash แสดงค่าแฮช MD5 เพื่อใช้ตรวจสอบความสมบูรณ์ของข้อมูลที่เกี่ยวข้องกับภัยคุกคาม
   - SHA-1 Hash แสดงค่าแฮช SHA-1 เพื่อใช้ตรวจสอบความสมบูรณ์ของข้อมูลที่เกี่ยวข้องกับภัยคุกคาม

- Source IP Address แสดงที่อยู่ IP ต้นทาง
- Destination IP Address แสดงที่อยู่ IP ปลายทาง
- Source Port แสดงพอร์ตที่อยู่ต้นทาง
- Destination Port แสดงพอร์ตที่อยู่ปลายทาง

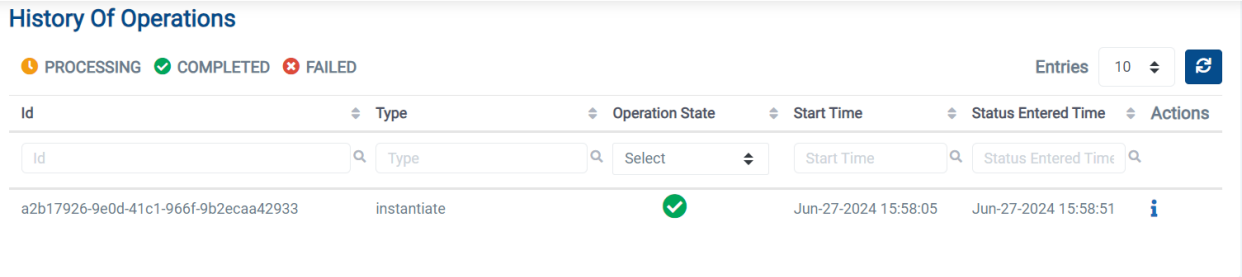
*รูปที่ 4.3.6: หน้าต่าง dashboard แสดงผลการตรวจสอบภัยคุกคามจาก Zeek*

## 4.4 ด้านประสิทธิภาพ (performance)

สำหรับการวัดประสิทธิภาพของระบบจะใช้เกณฑ์ในการพิจารณาด้วยกัน 3 ด้าน ได้แก่ เวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (deployment time) เวลาที่ใช้ในการตั้งค่าข้อมูลเบื้องต้นในแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น (configuration time) และเวลาแฝงโดยรวมทั้งหมดจากต้นทางถึงปลายทาง (End-to-end latency) โดยแนวทางการทดสอบได้ค่าทรัพยากรของเครื่อง VM ให้เหมือนกันคือ CPU จำนวน 4 core และ RAM จำนวน 4 GB

### 4.4.1 Deployment time

Deployment time เป็นการวัดเวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (NS intances) โดยผู้ดูแลระบบสามารถตรวจสอบเวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิสได้จากแถบข้อมูล Action ของหน้า OSM GUI (แสดงดังรูปที 4.4.1) โดยค่าเฉลี่ยเวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิสแต่ละประเภท ได้แก่ Suricata, Snort และ Zeek แสดงดังรูปที่ 4.4.2




*รูปที่ 4.4.1: ภาพแสดงข้อมูลที่เกี่ยวข้องกับระยะเวลาที่ใช้ในการจำลองและติดตั้งเน็ตเวิร์คเซอร์วิส (NS instances)*

### 4.4.2 Configuration time

Configuration time คือเวลาที่ใช้ในการตั้งค่าข้อมูลเบื้องต้นในแต่ละเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น ในที่นี้คือเวลาตั้งแต่เริ่มต้นที่เครื่อง VM ใช้ในการตั้งค่าข้อมูลเบื้องต้นที่อ่านจาก cloud-init ไฟล์ โดยสามารถเช็คข้อมูลเวลาได้จาก Logs ของ VM แต่ละเครื่องผ่าน openstack ดังรูปที่ 4.4.3 โดยค่าเฉลี่ยเวลาที่ใช้ในการตั้งค่าเน็ตเวิร์คเซอร์วิสแต่ละประเภท แสดงดังรูปที่ 4.4.4

### 4.4.3 End-to-end latency

End-to-end latency คือเวลาแฝงโดยรวมทั้งหมดจากต้นทางถึงปลายทาง โดยจะนับจากเวลาที่ใช้ส่ง log และ alert log จากตัวเน็ตเวิร์คเซอร์วิส (NS instances) ไปแสดงผลที่ Wazuh dashboard โดยการทดสอบมีดังนี้

1. ติดตั้ง pcap dataset ด้วยคำสั่ง wget และ unzip

   >> wget https://www.malware-traffic-analysis.net/2024/01/08/2024-01-08-GootLoader-infection-traffic.pcap.zip

   >> unzip -P infected_20240108 2024-01-08-GootLoader-infection-traffic.pcap.zip

2. replay traffic ไปยังพอร์ตที่ได้ตั้งค่าเอาไว้เพื่อจับข้อมูลทราฟฟิกด้วยคำสั่งพร้อมกับใช้คำสั่ง sudo tcpdump –i dummy0 เพื่อตรวจดูเวลาที่บันทึกข้อมูลทราฟฟิก

   >> sudo tcpreplay-edit -i dummy0 2024-01-08-GootLoader-infection-traffic.pcap

3. ตรวจดูข้อมูล log ที่ส่งไปยัง WAZUH แล้วเช็ค timestamp ที่ได้รับ จากนั้นนำมาหาความต่างของเวลากับ timestamp ของ alert log จากสมการดังด้านล่าง

$$Latency\ (s) = (T_{wazuh} - T_{alert}) + (T_{alert} - T_{trafficcome})$$

   แต่จากการทดลองทั้งหมดเวลาที่บันทึก $T_{alert}$ กับ $T_{trafficcome}$ มักจะเป็นเวลาเดียวกัน (ไม่มี latency) จึงไม่นำมาพิจารณาในส่วนนี้

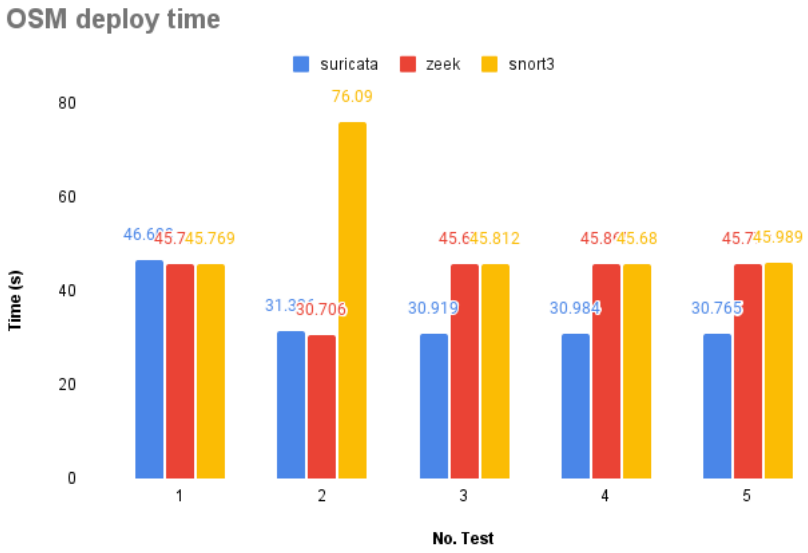
$$(T_{alert} - T_{trafficcome})$$

โดย $T_{wazuh}$ คือเวลาที่บันทึกเมื่อ log ส่งไปถึง Wazuh dashboard

$T_{alert}$ คือเวลาที่ alert ถูกบันทึกลงใน log ของแต่ละเน็ตเวิร์คเซอร์วิส

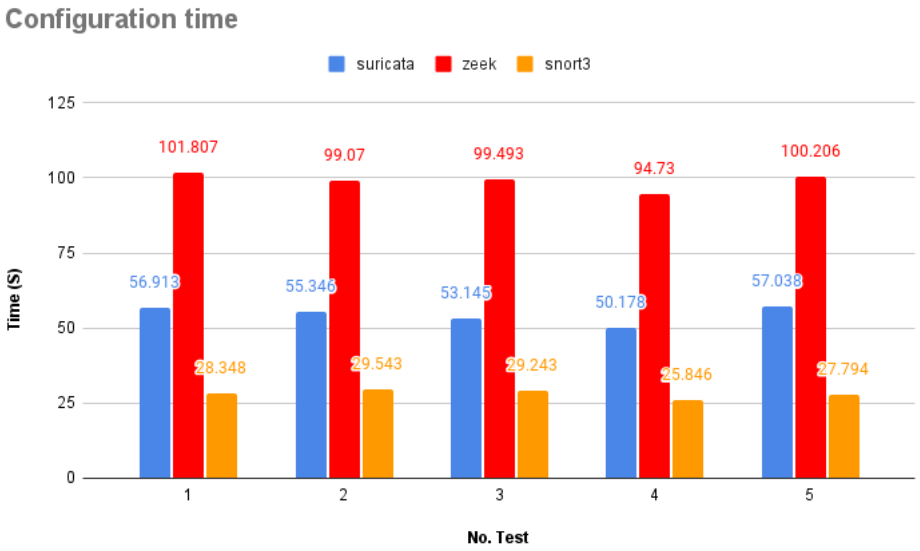
$T_{trafficcome}$ คือเวลาที่ traffic ที่ทำการประมวลผลเกิดการ alert

4. รูปที่ 4.4.5 และ รูปที่ 4.4.6 แสดงข้อมูล log alert ที่แสดงผ่าน Wazuh dashboard และผลการทดสอบ end-to-end latency

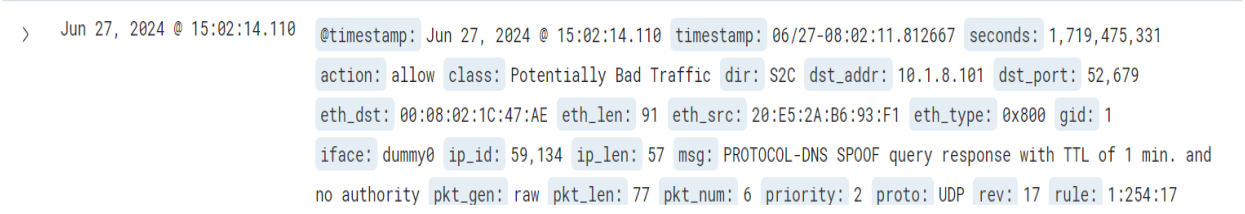



*รูปที่ 4.4.2: กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในการติดตั้งเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ได้แก่ Suricata, Snort และ Zeek (**หมายเหตุ:เนื่องจากว่าการติดตั้งนั้นต้องทำเป็นลำดับขั้นจึงทำให้มีความแตกต่างของเวลา ซึ่งในความเป็นจริงแล้วเวลาที่ใช้ควรจะใกล้เคียงกัน และก็ขึ้นอยู่กับการใช้งานทรัพยากรของเครื่องเซิฟเวอร์ในขณะนั้นด้วย)*

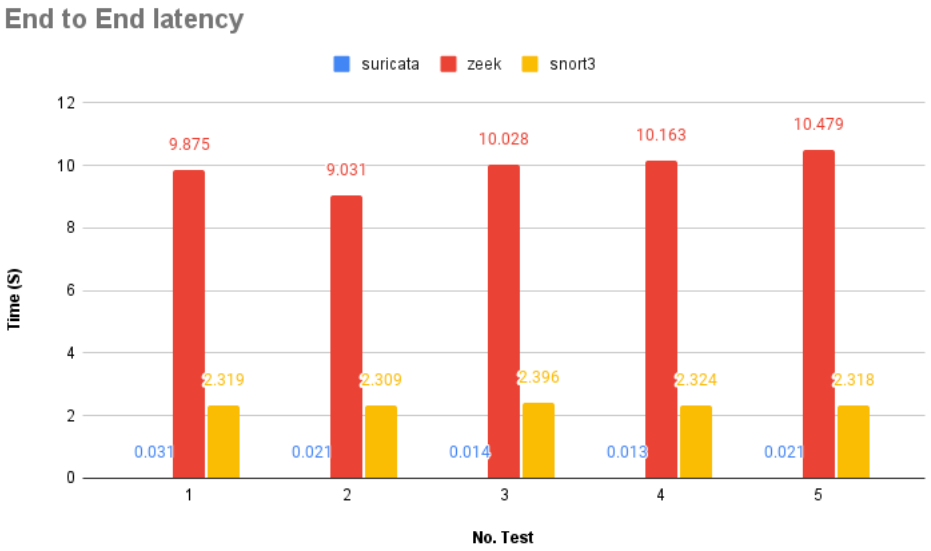
*รูปที่ 4.4.3: ภาพแสดงข้อมูลเกี่ยวกับระยะเวลาที่ใช้ในการตั้งค่าเบื้องต้นจากไฟล์ cloud-init*




***รูปที่ 4.4.4:*** *กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในตั้งค่าตั้งค่าเบื้องต้นในแต่ละเน็ตเวิร์คเซอร์วิส (**หมายเหตุ (1) สาเหตุที่ Zeek ใช้เวลานานนั้นเกิดจากการที่ต้องมีการเปิดใช้งานโดยใช้คำสั่ง sudo zeekctl deploy; (2) สาเหตุที่ Suricata ใช้เวลานานกว่า Snort ก็มาจากการ configuration ใน Suricata.yaml ที่มีการใช้คำสั่ง sudo systemctl restart suricata.service ส่งผลให้ใช้เวลาเพิ่มขึ้นในส่วนนี้)*




***รูปที่ 4.4.5:*** *แสดงข้อมูลเกี่ยวกับ log alert ที่แสดงผ่าน Wazuh dashboard*

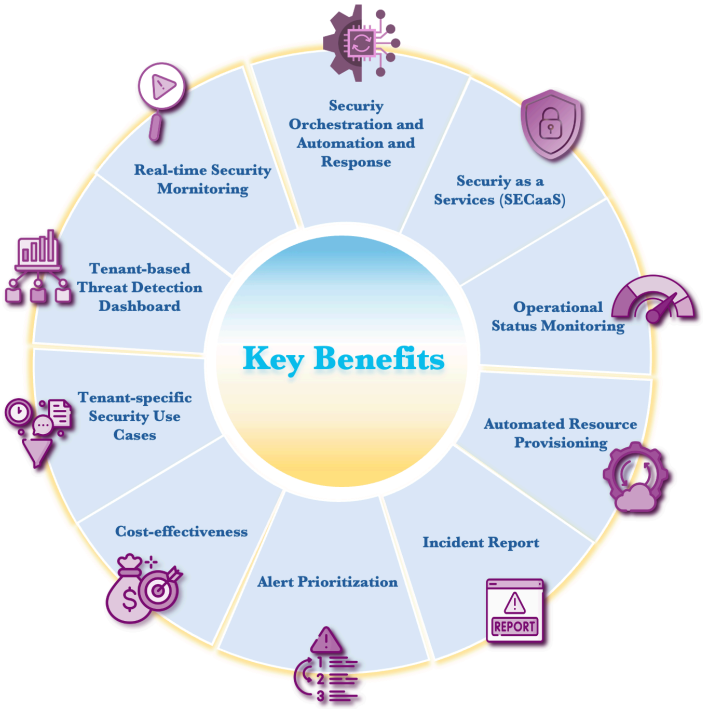



***รูปที่ 4.4.6:*** *กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาแฝงเฉลี่ย (end-to-end latency) ของแต่ละเน็ตเวิร์คเซอร์วิส (**หมายเหตุ: (1) จากผลการทดลองพบข้อสันนิฐานว่าสาเหตุที่ Suricata มีการส่งที่รวดเร็วอาจจะเป็นเพราะมีการรองรับ JSON log ไฟล์อยู่แล้ว ซึ่งแตกต่างจาก Zeek และ Snort ที่จะต้องมีการแปลงข้อมูลหรือเขียน log ให้อยู่ในรูป JSON) (2) สาเหตุที่ Zeek ส่งข้อมูลช้าอาจจะเป็นผลมาจากการที่ Zeek นั้นมีหลาย log ที่ต้องส่งข้อมูลจึงอาจจะทำให้เกิด bottle neck ที่ตัว fluent-bit ซึ่งทำหน้าที่เป็น log processor จึงทำให้เกิดความล่าช้าในส่วนนี้*

## 4.5 ประโยชน์ที่ผู้ใช้งานจะได้รับ

หากพิจารณาในมุมของผู้ใช้งานหรือหน่วยงานผู้รับบริการแล้ว เป้าหมายหลักของการพัฒนาแพลตฟอร์มนี้ คือ เพื่อช่วยแก้ปัญหาให้กับหน่วยงานต่างๆ ที่มีข้อจำกัด (1) ด้านการเงินต่อการลงทุน ติดตั้ง และค่าดำเนินการต่างๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัยในหน่วยงาน และ (2) ขาดแคลนบุคลากรผู้มีความเชี่ยวชาญด้าน cybersecurity ด้วยการช่วยอำนวยความสะดวกและลดภาระงานด้านความปลอดภัยให้กับหน่วยงานที่มาขอรับบริการ โดยสิ่งที่โครงการฯ นำเสนอ คือ การจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ เพื่อให้หน่วยงานดังกล่าวฯ สามารถเรียกใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ทางแพลตฟอร์มได้จัดเตรียมไว้ให้ได้ตามความต้องการ (on-demand services) เพื่อประโยชน์ต่อการช่วยวิเคราะห์ตรวจสอบ และเฝ้าระวังภัยคุกคามที่อาจจะเกิดขึ้นภายในหน่วยงานได้ โดยลักษณะเด่นของแพลตฟอร์มฯ มีการประยุกต์ใช้เทคโนโลยี Network Functions Virtualization (NFV) เข้ามาช่วยยกระดับการบริหารจัดการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบรวมศูนย์ ซึ่งจะช่วยเพิ่มประสิทธิภาพการทำงานได้ดียิ่งขึ้น ทั้งในแง่ความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส ความยืดหยุ่นในการจัดสรรทรัพยากร และประสบการณ์ที่ผู้ใช้งานจะได้รับ โดยประโยชน์ที่ผู้ใช้งาน (หน่วยงานที่รับบริการ) จะได้รับ สามารถสรุปได้เป็นข้อๆ ดังต่อไปนี้ (ดังรูปที่ 4.5.1)

1. **Security Orchestration, Automation and Response (SOAR):** ความรวดเร็วในการจำลองเน็ตเวิร์คเซอร์วิส การตั้งค่านโยบายด้านความปลอดภัย การออกแบบพิมพ์เขียวที่ขับเคลื่อนด้วยโมเดล




*รูปที่ 4.5.1: คุณลักษณะสำคัญของแพลตฟอร์มฯ*

(model-driven based network service templates) และการบริหารจัดการแบบอัตโนมัติ ที่สามารถตอบสนองต่อภัยคุกคามไซเบอร์ได้อย่างทันถ่วงที

2. **Security as a Services (SECaaS):** มีการเชื่อมต่อเข้ากับเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ เข้าด้วยกัน ดังเช่น
   - Deep Packet Inspection (e.g., Suricata) ใช้สำหรับป้องกันความปลอดภัยบนระบบเครือข่ายโดยการกรองเนื้อหาที่เป็นอันตรายผ่านการตรวจสอบเชิงลึกในรูปแบบ packet filtering ที่สามารถตรวจสอบเนื้อหาทั้งหมดภายในแพ็กเก็ตข้อมูลอย่างละเอียด โดยจะทำการตรวจสอบทั้งส่วนหัว (packet header) และเนื้อหาภายใน (content packet)
   - Intrusion Detection System (e.g., Snort) ใช้ในการตรวจสอบและวิเคราะห์ข้อมูลเครือข่าย เพื่อตรวจจับสัญญาณหรือกิจกรรมที่อาจก่อให้เกิดการบุกรุก ซึ่งเป็นระบบที่ใช้สำหรับการเฝ้าระวัง และแจ้งเตือนภัยหากมีการบุกรุกหรือมีสิ่งที่ผิดปกติเกิดขึ้น โดยวิธีการตรวจสอบนั้นจะอ้างอิงข้อมูลลายเซ็นหรือลักษณะเฉพาะของการบุกรุก (signature-based IDS) และนโยบายความปลอดภัย (security policy) ที่ผู้ใช้งานได้ทำการกำหนดไว้ล่วงหน้ามาช่วยในการตรวจสอบ
   - Traffic analyzer (e.g., Zeek) เป็นเครื่องมือที่ใช้ในการติดตามเหตุการณ์ด้านความปลอดภัยเป็นหลัก โดยระบบจะทำการสร้าง logs เอาไว้เพื่อบันทึกข้อมูล (network traffic) หรือกิจกรรมต่างๆ ที่เกิดขึ้นในเครือข่าย สำหรับใช้ประกอบการวิเคราะห์และตรวจสอบหาเหตุการณ์ที่น่าสงสัยต่อไป
3. **Real-time security monitoring:** ช่วยวิเคราะห์ ตรวจสอบ และเฝ้าระวังภัยคุกคามไซเบอร์ที่อาจจะเกิดขึ้นกับหน่วยงานแบบเรียลไทม์

4. **Tenant-based threat detection dashboard:** ขีดความสามารถในการแสดงผลข้อมูลภัยคุกคามโดยแบ่งตามลักษณะของผู้ใช้งาน กล่าวคือ หน่วยงานผู้ขอรับบริการสามารถเห็นผลลัพธ์ที่ได้จากการวิเคราะห์ภัยคุกคามเฉพาะข้อมูลที่เกี่ยวข้องกับหน่วยงานของตนเองได้เท่านั้น
5. **Tenant-specific security use cases:** สามารถรองรับการใช้งานที่หลากหลาย โดยการจัดทำเงื่อนไขการตรวจสอบ ปรับแต่งการตั้งค่า กำหนดนโยบายการตรวจสอบและวิธีปฏิบัติ เพื่อให้มีเฉพาะเจาะจงกับความต้องการใช้งานของแต่ละหน่วยงานได้
6. **Operationa status monitoring:** ผู้ดูและระบบสามารถติดตามและตรวจสอบสถานะทำงานของแต่ละ instances ต่างๆ ที่เกิดขึ้นในระบบหลังบ้านได้ ประกอบด้วย
    - CPU utilization: สถานะการใช้งานทรัพยากรประมวลผลในแต่ละ instance
    - RAM utilization: สถานะการใช้งานทรัพยากรหน่วยความจำในแต่ละ instance
    - Resident memory usage: สถานะการใช้งานหน่วยความจำจริงในแต่ละ instance
    - Process virtual memory: สถานะใช้งานหน่วยความจำเสมือนในแต่ละ instance
    - VM status: สถานะการทำงานของ Virtual Machines (VMs) ที่ได้ถูกสร้างขึ้น
    - VNF status: สถานะการทำงานของเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (VNFs) ที่ได้ถูกจำลองขึ้น
7. **Automated resource provisioning:** อีกหนึ่งคุณลักษณะเด่นของแพลตฟอร์ม คือ ความสามารถในการจัดสรรหรือขยายทรัพยากรให้กับเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้นได้อย่างยืดหยุ่น โดยการปรับลดหรือขยายขนาดของทรัพยากร (scale in/out) ตามประสิทธิภาพของการใช้งาน หรือเมื่อประสิทธิภาพของทรัพยากรที่กำลังใช้งานอยู่เกินกว่าค่า threshold ที่กำหนดไว้ ซึ่งทางผู้ออกแบบได้กำหนดเงื่อนไขและสร้าง metric สำหรับติดตามสถานะการใช้งานทรัพยากรเครือข่ายเอาไว้ในแบบพิมพ์เขียวของแต่ละเน็ตเวิร์คเซอร์วิส (network service templates) โดย NFV orchestrator จะดำเนินการปรับลดหรือขยายทรัพยากรให้อัตโนมัติ หากตรวจสอบพบว่าเน็ตเวิร์คเซอร์วิสดังกล่าวฯ มีการใช้งานทรัพยากรถึงค่า threshold ที่กำหนด
8. **Cost-effectiveness:** การช่วยลดค่าใช้จ่ายด้านการลงทุนสำหรับการจัดซื้อและติดตั้งอุปกรณ์เครือข่ายต่อการรักษาความปลอดภัยไซเบอร์ ค่าใช้จ่ายที่เกี่ยวข้องกับการบำรุงรักษาซ่อมแซม และค่าดำเนินการต่างๆ ที่เกิดขึ้นในองค์กร ซึ่งเป็นที่ทราบกันดีว่า โดยส่วนใหญ่แล้วค่าใช้จ่ายมักจะแปรผันตามขนาดของระบบเครือข่ายของแต่ละหน่วยงาน กล่าวคือ ยิ่งจำนวนของอุปกรณ์เครือข่ายยิ่งมาก ค่าใช้จ่ายในการจัดซื้อ ซ่อมแซม และบำรุงรักษาก็ยิ่งสูงขึ้นตาม ซึ่งแนวคิดการทำงานของแพลตฟอร์มฯ คือ การจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ (ซึ่งหากเป็นระบบเครือข่ายแบบดั้งเดิมจำเป็นต้องอาศัยการติดตั้งอุปกรณ์ฮาร์ดแวร์ที่มีหน้าที่การทำงานแบบจำเพาะในการดำเนินการ) โดยนำเน็ตเวิร์คเซอร์วิสต่าง ๆ ดังกล่าวฯ มาออกแบบพัฒนาในรูปแบบซอฟต์แวร์แอพพลิเคชั่นที่สามารถทำงานได้บนเครื่องเซิร์ฟเวอร์ทั่วไปได้ ลดการพึ่งอุปกรณ์เน็ตเวิร์คฮาร์ดแวร์จำเพาะและค่าใช้จ่ายในการบำรุงรักษาอุปกรณ์ระยะยาว

9. **Alert prioritization:** ดำเนินการวิเคราะห์ จำแนก และจัดลำดับความสำคัญของภัยคุกคาม ได้แก่ วิกฤต สูง ปานกลาง ต่ำ ซึ่งการจัดลำดับความสำคัญนี้จะช่วยให้หน่วยงานที่รับบริการสามารถเข้าใจผลกระทบที่อาจเกิดขึ้นโดยรวมได้ดีขึ้น
10. **Incident report:** รวบรวมข้อมูลภัยคุกคามที่ตรวจพบ และจัดทำรายงานการแจ้งเตือนตามช่วงเวลาหรือเงื่อนไขที่กำหนด

**4.6 บทสรุป**

สำหรับแนวทางการทดสอบประสิทธิภาพของระบบจะแบ่งเกณฑ์การวัดตามคุณสมบัติ 3 ด้านด้วยกัน ดังนี้ (1) **การทำงานเชิงหน้าที่** (functionality) มีวัตถุประสงค์เพื่อทำความเข้าใจถึงขั้นตอนการทำงานต่างๆ ตั้งแต่ระยะเริ่มต้น (pre-configuration and deployment time) ว่าด้วยการสร้างและออกแบบพิมพ์เขียวให้กับแต่ละเน็ตเวิร์คเซอร์วิส การกำหนดค่าเบื้องต้น การกำหนดค่า threashold สำหรับการทำ auto-scaling และการสร้าง metric สำหรับการทำ monitoring จนถึงระยะที่ใช้งานจริง (runtime) ว่ามีการดำเนินการอย่างไร (2) **ด้านการใช้งาน** (feasibility) มีเป้าหมายที่จะแสดงให้เห็นภาพส่วนแสดงผลของระบบ ซึ่งข้อมูลที่นำมาแสดงสามารถแบ่งข้อมูลออกเป็น 3 กลุ่มข้อมูลหลักๆ ได้แก่ (2.1) ข้อมูลที่แสดงสถานะทำงานของระบบหลังบ้าน ประกอบด้วย สถานะการใช้งานทรัพยากรประมวลผล (CPU utilization) หน่วยความจำ (RAM utilization) สถานะทำงานของเครื่อง VMs และเน็ตเวิร์คเซอร์วิส (VNF instances) เหล่านี้เป็นต้น (2.2) ข้อมูลแสดงสถานะการแจ้งเตือนของระบบ โดยข้อมูลทั้งสองกลุ่มนี้ถูกสร้างขึ้นมาเพื่ออำนวยความสะดวกให้แก่ผู้ดูแลระบบต่อการตรวจสอบและติดตามประสิทธิภาพการใช้งานทรัพยากรโครงสร้างพื้นฐานต่างๆ ว่ามีการใช้งานไปมากน้อยเพียงใด ตลอดจนคอยตรวจสอบการทำงานของ instances ต่างๆ และแจ้งเตือน (alerts) แก่ผู้ดูแลระบบเมื่อพบความผิดปกติเกิดขึ้น และ (2.3) ข้อมูลแสดงผลการวิเคราะห์ภัยคุกคาม (threat analysis) ตามรูปแบบเหตุการณ์ที่ได้กำหนดไว้ (signature-based detection) ซึ่งเป็นผลลัพธ์ที่ได้จากการประมวลผลในแต่ละเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ที่ได้ถูกจำลองขึ้นในที่นี้ได้แก่ Suricata, Snort และ Zeek โดยระบบได้ถูกออกแบบให้สามารถแสดงผลลัพธ์การตรวจสอบภัยคุกคามตามสิทธิ์ของผู้ใช้งาน ซึ่งผู้ดูแลระบบสามารถเห็นข้อมูลทั้งหมดที่ได้จากการวิเคราะห์ภัยคุกคามของทุกหน่วยงานที่เข้ารับบริการ ในขณะที่หน่วยงานผู้ขอรับบริการสามารถเห็นผลลัพธ์ของการวิเคราะห์ภัยคุกคามเฉพาะข้อมูลที่เกี่ยวข้องกับองค์กรของตนเองเท่านั้น (tenant-based dashboard) และ (3) **ด้านประสิทธิภาพ** (performance) มีจุดมุ่งหมายเพื่อตรวจวัดประสิทธิภาพการทำงานโดยรวมของระบบ โดยเกณฑ์ในการตรวจสอบจะพิจารณาจากหลักเกณฑ์ชี้วัดด้วยกัน 3 ด้าน ดังนี้ (3.1) เวลาที่ใช้ในการจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (deployment time) (3.2) เวลาที่ใช้ในการตั้งค่าข้อมูลเบื้องต้น (configuration time) และ (3.3) เวลาแฝงโดยรวมที่ใช้จากต้นทางถึงปลายทาง (end-to-end latency) โดยจะนับจากเวลาที่ใช้ในการส่งข้อมูล logs จากต้นทางเพื่อมาประมวลผลที่เน็ตเวิร์คเซอร์วิส และจากเน็ตเวิร์คเซอร์วิสส่ง alert logs ที่ได้จากการวิเคราะห์ผลเพื่อไปไปแสดงผลในรูปแบบกราฟรายงานต่างๆ ผ่าน dashboard โดยจากผลการทดสอบพบว่าระยะเวลาเฉลี่ยที่ใช้ในการวิเคราะห์และประมวลผลภัยคุกคามในแต่ละเน็ตเวิร์คเซอร์วิสอยู่ในเกณฑ์ที่น่าพอใจ กล่าวคือ 0.02, 3.33 และ 9.91 วินาทีสำหรับการประมวลผลของ Suricata, Snort และ Zeek ตามลำดับ

# บทที่ 5
# กิจกรรมที่ 4: ถ่ายทอดความรู้เชิงวิจัยและนวัตกรรม

## 5.1 บทนำ

ในยุคดิจิทัลที่เทคโนโลยีมีการพัฒนาไปอย่างรวดเร็ว อุตสาหกรรมต่างๆ รวมถึงธุรกิจขนาดกลางและขนาดเล็กต่างได้รับประโยชน์อย่างมากจากการนำเทคโนโลยีมาประยุกต์ใช้เพื่อปรับปรุงกระบวนการทำงานของธุรกิจให้มีประสิทธิภาพมากยิ่งขึ้น ไม่ว่าจะเป็นการเพิ่มผลผลิต ช่วยลดต้นทุน ช่วยเพิ่มประสิทธิภาพในการตัดสินใจ ตลอดจนการให้บริการและตอบสนองลูกค้าได้อย่างสะดวกรวดเร็ว จะเห็นได้ว่าการประยุกต์ใช้เทคโนโลยีดิจิทัลถือเป็นกลไกขับเคลื่อนสำคัญที่จะช่วยให้อุตสาหกรรมต่างๆ สามารถสร้างข้อได้เปรียบทางการแข่งขันในตลาดได้เป็นอย่างดี ในอีกมุมหนึ่ง การเปลี่ยนผ่านสู่ยุคดิจิทัล นำมาซึ่งการเพิ่มความเสี่ยงด้านความมั่นคงปลอดภัยไซเบอร์ อีกทั้งเป็นการสร้างพื้นผิดการโจมตี (attack surface) ให้ขยายวงกว้างมากยิ่งขึ้น จากสถิติในปีพ.ศ. 2564 ถึงพ.ศ. 2566 [61] พบว่าภัยคุกคาม ดังเช่น มัลแวร์เรียกค่าไถ่ (Ransomware) การขโมยข้อมูล และการโจมตีแบบปฏิเสธการให้บริการ (DDoS attack) มีอัตราเพิ่มสูงขึ้นถึง 76% ผลกระทบดังกล่าวฯ ก่อให้เกิดความเสียหายทางการเงินและชื่อเสียงขององค์กรที่ไม่สามารถประเมินค่าได้ แม้ว่าองค์กรหลายแห่งต่างตระหนักถึงปัญหาภัยคุกคามทางไซเบอร์ แต่ด้วยข้อจำกัดด้านงบประมาณและความเชี่ยวชาญของบุคลากรที่มีอยู่อย่างจำกัด ทำให้องค์กรเหล่านั้นต้องตกอยู่ภายใต้สถานการณ์ความเสี่ยงที่อาจจะถูกโจมตีได้ตลอดเวลา ดังนั้นเพื่อลดความเสี่ยงจากภัยคุกคามทางไซเบอร์ แพลตฟอร์มสำหรับบริหารและให้บริการเน็ตเวิร์คแบบเสมือนด้านความปลอดภัยต่างๆ (ดังเช่น IDS/IPS, IDP, Traffic analyzer) จึงมีความสำคัญอย่างยิ่งที่จะช่วยอำนวยความสะดวกและลดภาระงานด้านความปลอดภัยให้กับองค์กร โดยเป้าหมายในการพัฒนาแพลตฟอร์มนี้มุ่งผลสัมฤทธิ์ 3 ด้านหลัก ดังนี้ (1) *ด้านต้นทุน:* โดยการช่วยลดค่าใช้จ่ายด้านการลงทุนและค่าดำเนินการต่างๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัยในองค์กร; (2) *ด้านการรักษาความปลอดภัย:* โดยการจัดเตรียมระบบนิเวศเสมือนจริง (management and orchestration framework) สำหรับบริหารและจัดสรรทรัพยากรโครงสร้างพื้นฐานให้พร้อมสำหรับใช้งาน และการให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยอัตโนมัติ รองรับรูปแบบการให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs as a Service: VNFaaS) ไว้บนคลาวด์ เพื่อให้หน่วยงานที่เกี่ยวข้องสามารถใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ ที่จัดเตรียมไว้ให้ได้อย่างเหมาะสม สนับสนุนการรักษาความปลอดภัยบนระบบเครือข่ายขององค์กร; และ (3) *ด้านประสิทธิภาพ:* การขับเคลื่อนด้วยเทคโนโลยีการทำงานเสมือนจริงเครือข่าย (Network Functions Virtualization: NFV) ที่จะช่วยยกระดับการบริหารจัดการเน็ตเวิร์คเซอร์วิสแบบรวมศูนย์ (centralized model) และมีความเป็นอัตโนมัติ (automated workflows) ซึ่งจะช่วยเพิ่มประสิทธิภาพการทำงานได้ดียิ่งขึ้น ทั้งในแง่ความรวดเร็วในการสร้างและจำลองเน็ตเวิร์กเซอร์วิสด้านความปลอดภัย ความยืดหยุ่นในการจัดสรรทรัพยากร และประสบการณ์ที่ผู้ใช้งานจะได้รับ

ดังนั้น ในหัวข้อนี้จะพูดถึงแนวทางในการเผยแพร่ประชาสัมพันธ์ผลงาน การถ่ายทอดองค์ความรู้ การเผยแพร่และขยายผลนวัตกรรมที่พัฒนาขึ้น เพื่อให้เกิดการรับรู้และนำไปสู่การใช้งานจริงอย่างแพร่หลาย โดย

แนวทางประกอบด้วย การจัดสัมมนาเชิงวิชาการ การออกบูธจัดแสดงนิทรรศการ การจัดทำเอกสารเผยแพร่ และการสร้างความร่วมมือกับหน่วยงานที่เกี่ยวข้อง

**5.2 การจัดสัมมนาเชิงวิชาการ**

สำหรับการจัดสัมมนาเชิงวิชาการนี้ มีวัตถุประสงค์หลักเพื่อเปิดเวทีแลกเปลี่ยนองค์ความรู้ระหว่างนักวิจัย นักพัฒนา ผู้เชี่ยวชาญจากภาคอุตสาหกรรม และหน่วยงานภาครัฐที่เกี่ยวข้อง โดยมุ่งเน้นการนำเสนอผลงานวิจัย เทคโนโลยีต้นแบบ แนวทางการประยุกต์ใช้งานจริง และการหารือร่วมกันถึงทิศทางการต่อยอดและพัฒนาในอนาคต โดยที่ผ่านมาทีมวิจัยได้มีโอกาสเข้าร่วมกิจกรรมสัมมนาจำนวน 2 ครั้งดังนี้

**5.2.1 กิจกรรมสัมมนา “เพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์”** จัดโดยสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) ครั้งที่ 1 ในวันที่ 17 กันยายน พ.ศ. 2567 ณ โรงแรมเซ็นทาราแกรนด์ เซ็นทรัลลาดพร้าว กรุงเทพมหานคร (รายละเอียดดังเอกสารแนบท้ายภาคผนวก ก.)

โดยกิจกรรมสัมมนานี้จัดขึ้นเพื่อส่งเสริม สนับสนุนงานวิจัย และพัฒนาเทคโนโลยีและองค์ความรู้ที่เกี่ยวกับการรักษาความมั่นคงปลอดภัยไซเบอร์ ตลอดจนมุ่งเน้นการสร้างเครือข่าย แบ่งปันข้อมูลระหว่างนักวิจัย ส่งเสริมความร่วมมือในการบริหารจัดการความมั่นคงปลอดภัยไซเบอร์ที่มีคุณค่าเพื่อเป็นประโยชน์ต่อการต่อยอดและพัฒนานักวิจัยและองค์กรต่างๆ ด้านความมั่นคงปลอดภัยไซเบอร์อย่างมีประสิทธิภาพและประสิทธิภาพ โดยดร. มลธิดา ภัทรนันทกุล (หัวหน้าโครงการฯ) ทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ กลุ่มวิจัยการสื่อสารและเครือข่ายเนคเทค สวทช. ได้รับเชิญมาเป็นวิทยากรเพื่อนำเสนอผลงานวิจัยที่เกี่ยวข้องต่อการรักษาความมั่นคงปลอดภัยไซเบอร์ในครั้งนี้ด้วย

***รูปที่ 5.2.1:*** *กล่าวรายงานการจัดกิจกรรมโดยผู้ช่วยศาสตราจารย์ ดร. วราภรณ์ พรหมวิกร ผู้ช่วยเลขาธิการคณะกรรการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)*

***รูปที่ 5.2.2:*** *กล่าวเปิดงานและบรรยายพิเศษโดยพลอากาศตรีอมร ชมเชย ผู้ช่วยเลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)*

***รูปที่ 5.2.3:*** *ภาพบรรยากาศกิจกรรมสัมมนา*

***รูปที่ 5.2.4:*** *ดร มลธิดา ภัทรนันทกุล นักวิจัยจากสวทช. สังกัดเนคเทค ได้เสนองานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ ในหัวข้อเรื่อง ซีโร่ทัชเซอร์วิส ระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยอัตโนมัติสำหรับตรวจสอบและเฝ้าระวังภัยคุกคามไซเบอร์*

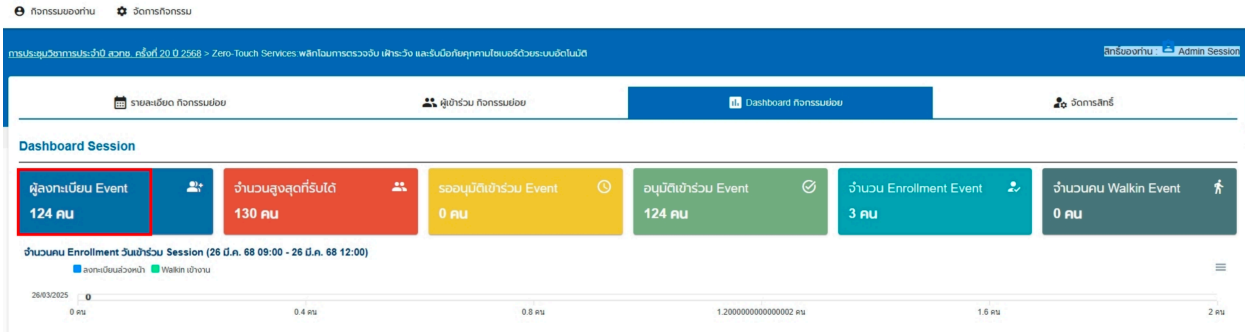
**5.2.2 กิจกรรมเสวนาในหัวข้อ “Zero-Touch Services: ผลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ” ในงานการประชุมวิชาการประจำปีสวทช. (NSTDA Annual Conference: NAC 2025) ครั้งที่ 20**” จัดโดยสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) ในวันที่ 26 มีนาคม พ.ศ. 2568 ณ ศูนย์ประชุมอุทยานวิทยาศาสตร์ประเทศไทย อาคาร 14 ถนนพหลโยธิน ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี (รายละเอียดดังเอกสารแนบท้าย ภาคผนวก ข.)

ด้วย สวทช. มีกำหนดการจัดงานประชุมวิชาการประจำปี 2568 โดยมุ่งเน้นในการนำเสนอผลงานวิจัยภายใต้ยุทธศาสตร์การขับเคลื่อนประเทศไทยด้วยวิทยาศาสตร์และเทคโนโลยีเพื่อพัฒนาประเทศไทยให้เติบโตอย่างมีคุณภาพและยั่งยืน ในการนี้ทางทีมวิจัยจึงได้มีส่วนร่วมในการจัดกิจกรรมเสวนาหัวข้อ “Zero-Touch Services: ผลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ”ขึ้นมา ซึ่งนวัตกรรมซีโร่ทัชเซอร์วิสถือเป็นส่วนหนึ่งที่สนับสนุนการทำงานของแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) โดยวัตถุประสงค์การจัดกิจกรรมเสวนาในครั้งนี้มีวัตถุประสงค์หลักที่สำคัญดังนี้

- เพื่อเผยแพร่แนวคิดและองค์ความรู้ในการพัฒนาแพลตฟอร์มฯ ซึ่งเป็นกระบวนทัศน์ใหม่ในการบริหารจัดการความมั่นคงปลอดภัยไซเบอร์ด้วยระบบอัตโนมัติ
- เพื่อแลกเปลี่ยนความคิดเห็นและประสบการณ์จากผู้เชี่ยวชาญ นักวิจัย และผู้ปฏิบัติงานจริง เกี่ยวกับการนำเทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) มาประยุกต์ใช้ให้เกิดประสิทธิภาพสำหรับการรักษาความปลอดภัยไซเบอร์ให้กับองค์กร
- เพื่อกระตุ้นให้หน่วยงานต่างๆ ตระหนักถึงความสำคัญของการปรับตัวด้านความมั่นคงปลอดภัยไซเบอร์ในยุคดิจิทัล อีกทั้งเพื่อเป็นการเผยแพร่ ขยายผล และส่งเสริมประโยชน์จากนวัตกรรมนี้ไปสู่การใช้งานจริง

- เพื่อเป็นเวทีในการประชาสัมพันธ์ผลงานวิจัย ตลอดจนขยายผลและพัฒนานวัตกรรมที่เกี่ยวข้องกับการรักษาความมั่นคงปลอดภัยไซเบอร์สู่ภาคส่วนต่างๆ ทั้งภาคอุตสาหกรรม ภาคอุดมศึกษา ภาครัฐ และภาคเอกชน
- เพื่อส่งเสริมการสร้างเครือข่ายความร่วมมือระหว่างทีมนักวิจัย หน่วยงานที่สนใจ และผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้อง อันนำไปสู่การพัฒนาอย่างยั่งยืนในอนาคต

*รูปที่ 5.2.5: โปสเตอร์ประชาสัมพันธ์กิจกรรมเสวนา Zero-Touch Services ที่จัดขึ้น*




*รูปที่ 5.2.6: สถิติผู้สนใจลงทะเบียนเข้าร่วมกิจกรรมเสวนา*

*รูปที่ 5.2.7: บรรยากาศจุดรับลงทะเบียนหน้าห้องจัดสัมมนา*

*รูปที่ 5.2.8: แนะนำวิทยากร*

*รูปที่ 5.2.9: คุณเอกฉันท์ รัตนเลิศนุสรณ์ หัวหน้าทีมวิจัยความมั่นคงปลอดภัยไซเบอร์ กลุ่มวิจัยการสื่อสารและเครือข่าย เนคเทค กล่าวต้อนรับและเปิดงาน*

*รูปที่ 5.2.10: ดร. มลธิดา ภัทรนันทกุล (หัวหน้าโครงการฯ) ทีมวิจัยความมั่นคงปลอดภัยไซเบอร์ กลุ่มวิจัยการสื่อสารและเครือข่าย เนคเทค ได้กล่าวถึงภูมิทัศน์ภัยคุกคามทางไซเบอร์ในประเทศไทย*

***รูปที่ 5.2.11:*** *มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคอุดมศึกษา โดยรองศาสตราจารย์ปานใจ ธารทัศนวงศ์ ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร*

***รูปที่ 5.2.12:*** *มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาครัฐ โดยนายแพทย์ภาณุพงศ์ ตันติรัตน์ รองผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงสาธารณสุข*

***รูปที่ 5.2.13:*** *มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคเอกชน โดยคุณนรินทร์ รุณภัย ที่ปรึกษาอิสระ ด้านความมั่นคงปลอดภัยไซเบอร์ ภาคเอกชน*

***รูปที่ 5.2.14:*** *มุมมองความคิดเห็นด้านภัยคุกคามทางไซเบอร์ในภาคเอกชน โดยคุณอังคาร รองกรรมการผู้จัดการ บริษัททีเ-เน็ต ไอที โซลูชัน จำกัด*

***รูปที่ 5.2.15:*** *แนะนำแพลตฟอร์มโดยดร. มลธิดา ภัทรนันทกุล*

*รูปที่ 5.2.16: บรรยากาศกิจกรรมเสวนา*

*รูปที่ 5.2.17: ช่วงถามตอบ*

**5.3 การออกบูธจัดแสดงนิทรรศการ**

การถ่ายทอดองค์ความรู้ผ่านการออกบูธจัดแสดงนิทรรศการ นับเป็นอีกหนึ่งกิจกรรมสำคัญที่สามารถสร้างการรับรู้และความเข้าใจเชิงภาพรวมเกี่ยวกับแนวคิด เทคโนโลยี และผลงานวิจัยที่เกี่ยวข้องให้แก่กลุ่มเป้าหมายที่หลากหลาย ไม่ว่าจะเป็นหน่วยงานภาครัฐ ภาคอุตสาหกรรม ภาคเอกชน นักวิจัย นักศึกษา และประชาชนทั่วไป โดยเฉพาะอย่างยิ่งวัตถุประสงค์ของกิจกรรมนี้ คือ (1) มุ่งเน้นนำเสนอผลงานวิจัยด้าน Zero-Touch Service: ระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยอัตโนมัติสำหรับตรวจสอบและเฝ้าระวังภัยคุกคามทางไซเบอร์ ซึ่งเป็นนวัตกรรมที่สนับสนุนการทำงานของแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) (โครงการวิจัยฯ ที่ได้รับทุน) (2) แสดงตัวอย่างการทำงานของต้นแบบเพื่อให้ผู้ใช้

เห็นภาพการใช้งานจริง (3) ถ่ายทอดองค์ความรู้และนวัตกรรมที่เกิดขึ้น (4) สร้างพื้นที่สำหรับการแลกเปลี่ยนองค์ความรู้ และเปิดโอกาสให้ผู้ที่สนใจได้รู้จักกับผลงานวิจัยที่สามารถต่อยอดไปสู่การใช้งานจริง ตลอดจน (5) สร้างเครือข่ายความร่วมมือกับหน่วยงานพันธมิตร เพื่อต่อยอดและขยายผลงานวิจัยไปสู่การใช้งานจริงในวงกว้าง ซึ่งที่ผ่านมาทีมวิจัยได้มีโอกาสนำผลงานวิจัยไปเผยแพร่และประชาสัมพันธ์ผ่านการออกบูธจัดแสดงนิทรรศการ 2 งาน ด้วยกันดังนี้

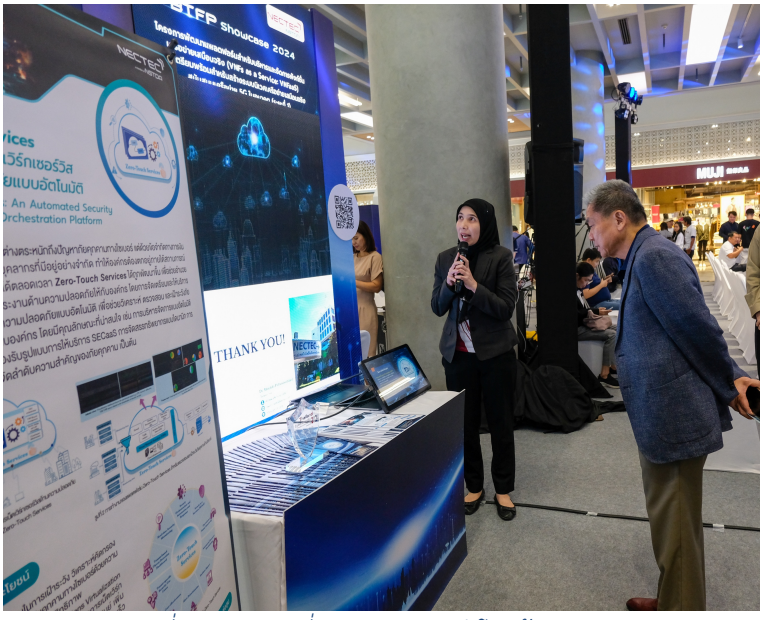
**5.3.1 งานประชุมวิชาการและนิทรรศการเนคเทค ประจำปี 2567 (NECTEC-ACE 2024)** จัดขึ้นโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค) ในวันที่ 10 กันยายน พ.ศ. 2567 ณ อิมแพคฟอรั่ม เมืองทองธานี

*รูปที่ 5.3.1: ภาพบรรยากาศการออกบูธจัดแสดงผลงานวิจัยในงาน NECTEC-ACE 2024*

*รูปที่ 5.3.2: แนะนำผลงานและแลกเปลี่ยนความคิดเห็นจากผู้ที่สนใจ*

**5.3.2 งานจัดแสดงนิทรรศการ BTFP Showchase 2024 ภายใต้สโลแกน “ส่งต่อพลังหนุน ส่งทุนสร้างสิ่งดีๆ”** จัดขึ้นโดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ (กทปส.) ในวันที่ 22 พฤศจิกายน พ.ศ. 2567 ณ ลานกิจกรรม ชั้น G ศูนย์การค้าสามย่านมิตรทาวน์ (รายละเอียดดังเอกสารแนบท้าย ภาคผนวก ค.)

*รูปที่ 5.3.3: ภาพบรรยากาศการออกบูธจัดแสดงผลงานวิจัยในงาน BTFP Showcase 2024*



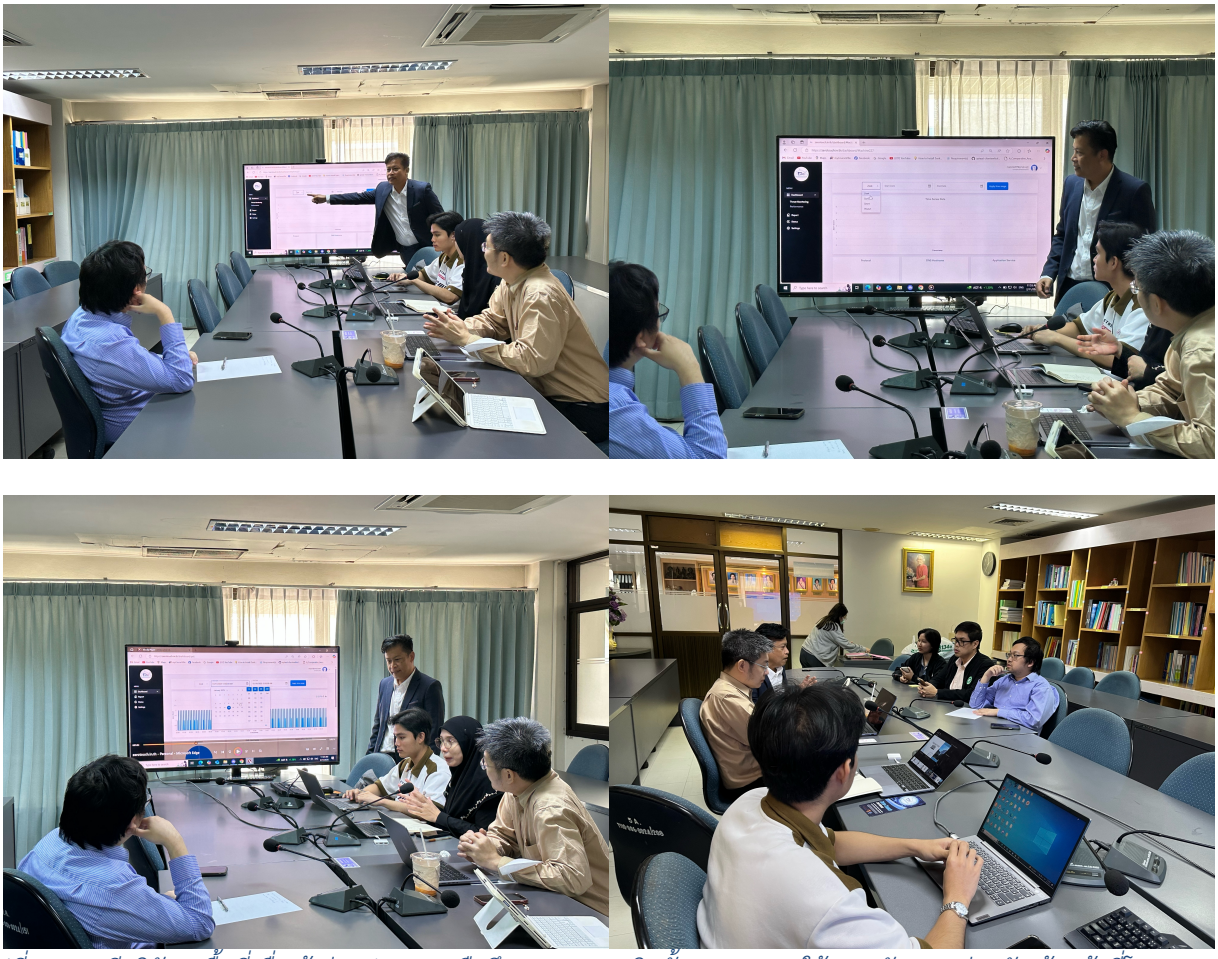

*รูปที่ 5.3.4: การเยี่ยมชมผลงานวิจัยโดยผู้ทรงคุณวุฒิ*

*รูปที่ 5.3.5: บรรยากาศการแลกเปลี่ยนความคิดเห็นจากผู้ที่สนใจ*

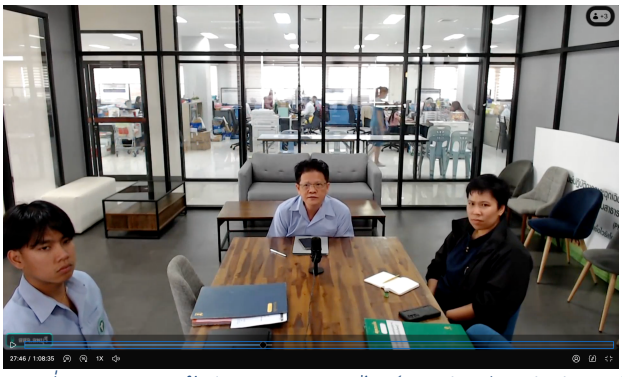
*รูปที่ 5.3.6: กิจกรรมเสวนาบนเวที*

**5.4 การสร้างความร่วมมือกับหน่วยงานที่เกี่ยวข้อง**

จากการเผยแพร่ประชาสัมพันธ์ผลงานวิจัยผ่านเวทีต่างๆ เช่น การจัดกิจกรรมเสวนา งานสัมมนาวิชาการ และการออกบูธจัดแสดงนิทรรศการ พบว่ามีหลายหน่วยงานโดยเฉพาะองค์กรภาครัฐ อาทิ หน่วยงานด้านความมั่นคง หน่วยงานในสังกัดอุดมศึกษา รวมถึงโรงพยาบาลรัฐบาลที่อยู่ภายใต้การกำกับดูแลของกระทรวงสาธารณสุข เขตสุขภาพที่ 4 ได้แสดงความสนใจในแนวคิดและศักยภาพของนวัตกรรมที่พัฒนา และได้ขอความอนุเคราะห์จากทีมวิจัยเพื่อนำระบบไปทดสอบติดตั้งใช้งานในรูปแบบของโครงการนำร่อง โดยมีเป้าหมายเพื่อประเมินความเหมาะสมของระบบกับบริบทการใช้งาน และศึกษาศักยภาพของระบบในการช่วยเสริมสร้างความมั่นคงปลอดภัยไซเบอร์ให้กับองค์กร (รายละเอียดดังเอกสารแนบท้าย ภาคผนวก ง. และจ.) ซึ่งในเบื้องต้น ทางทีมวิจัยได้ดำเนินการจัดประชุมออนไลน์กับหน่วยงานที่เข้าร่วมโครงการ (รูปที่ 5.4.1 และ 5.4.2) เพื่อนำเสนอภาพรวมของนวัตกรรมทั้งในด้านกรอบแนวคิดการออกแบบ ขอบเขตการใช้งาน และประโยชน์ที่คาดว่าหน่วยงานที่เข้าร่วมทดสอบจะได้รับ ตลอดจนเปิดโอกาสให้ผู้ที่เกี่ยวข้องจากแต่ละหน่วยงานได้ซักถาม แลกเปลี่ยนความคิดเห็น และเสนอข้อแนะนำ เพื่อให้ประกอบการวางแผนการจัดเตรียมระบบ ติดตั้งและทดสอบ รวมถึงการปรับปรุงระบบให้มีความเหมาะสมและสามารถตอบโจทย์การใช้งานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น



***รูปที่ 5.4.1:*** *ทีมวิจัยลงพื้นที่เพื่อเข้าร่วมประชุมหารือถึงแนวทางการติดตั้งและทดสอบใช้งานนวัตกรรมร่วมกับเจ้าหน้าที่โรงพยาบาลสระบุรี ต. ปากเพรียว อ. สระบุรี จ. สระบุรี เมื่อวันที่ 11 กุมภาพันธ์ พ.ศ. 2568*



***รูปที่ 5.4.2:*** *การเข้าร่วมประชุมออนไลน์ระหว่างทีมวิจัยกับสำนักงานสาธารณสุขจังหวัดลพบุรี เพื่อหารือถึงแนวทางการติดตั้งและทดสอบใช้งานนวัตกรรม เมื่อวันที่ 27 กุมภาพันธ์ พ.ศ. 2568*

## 5.5 ความสำเร็จที่ผ่านมา

ตามที่สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) ได้จัดให้มีกิจกรรมส่งเสริมการพัฒนาผลิตภัณฑ์และบริการด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ในภาคอุตสาหกรรม โดยการจัดให้มีการประกวดผลิตภัณฑ์และบริการในภาคอุตสาหกรรมด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ ประจำปี 2567 ซึ่งมีการกำหนดรับสมัครและคัดเลือกผลงานถึงวันที่ 21 ตุลาคม 2567 และจะมีการประกาศผลผู้ที่ได้รับรางวัลประเภทต่างๆ ในวันที่ 28 ตุลาคม 2567 ณ โรงแรม ทีเค พาเลซ แอนด์ คอนเวนชั่น กรุงเทพมหานคร นั้น (รายละเอียดแสดงดังรูปที่ 5.5.1) ในการนี้ทีมวิจัยมีความสนใจที่จะเข้าร่วมกิจกรรมดังกล่าวฯ ด้วยการส่งผลงานวิจัย "ซีโร่ทัชเซอร์วิส: ระบบให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ สำหรับตรวจสอบและเฝ้าระวังภัยคุกคามไซเบอร์ (Zero-Touch Services: An Automated Security Management and Orchestration Platform for Cyber Investigation and Mitigation)" เข้าร่วมการประกวดในครั้งนี้ด้วย ภายใต้โครงการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชันเครือข่ายเสมือนจริง (VNFs as a Services: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) สัญญารับทุนเลขที่ B 64-3-(2)-004 โดยมีวัตถุประสงค์เพื่อเผยแพร่ประชาสัมพันธ์ผลงานวิจัย สนับสนุนให้องค์กรในภาคอุตสาหกรรม ภาครัฐ และหน่วยงานที่สนใจสามารถใช้บริการนวัตกรรมซีโร่ทัชเพื่อตรวจสอบและเฝ้าระวังภัย

คุกคามไซเบอร์ให้องค์กรได้อย่างมีประสิทธิภาพ ด้วยกรอบแนวคิดในการให้บริการเน็ตเวิร์คเซอร์วิสแบบอัตโนมัติ ลดการพึ่งพาแรงงานมนุษย์ เพื่อตอบสนองต่อภัยคุกคามได้อย่างทันท่วงที

จากการประกาศผลเมื่อวันที่ 28 ตุลาคม 2567 ที่ผ่านมาผลงานวิจัยดังกล่าวฯ ได้รับการคัดเลือกให้ได้รับรางวัล “ความเป็นเลิศด้านความคิดสร้างสรรค์” ประเภทต้นแบบผลิตภัณฑ์หรือบริการ (Prototype) ด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ (รายละเอียดแสดงดังรูปที่ 5.5.2) ซึ่งนับว่าเป็นอีกหนึ่งก้าวสำคัญต่อการผลักดันผลงานวิจัยสู่การใช้งานจริงระดับประเทศ เวทีการประกวดนี้เปรียบเสมือนเวทีสำคัญระดับประเทศที่ได้รับการยอมรับในวงกว้าง โดยมีบทบาทสำคัญในการมุ่งส่งเสริม สนับสนุน และสร้างโอกาสการต่อยอดผลงานวิชาการและผลงานวิจัยเชิงลึกด้านความมั่นคงปลอดภัยไซเบอร์ สู่การพัฒนาเป็นนวัตกรรมที่นำไปใช้ได้จริงทั้งในภาครัฐและภาคเอกชน นอกเหนือจากนี้ การได้รับรางวัลนี้ยังสะท้อนให้เห็นถึงศักยภาพของนักวิจัยไทยในการพัฒนาเทคโนโลยีด้านความมั่นคงปลอดภัยไซเบอร์ที่มีความคิดสร้างสรรค์ ทันสมัย และสอดคล้องกับความต้องการของผู้ใช้ในบริบทประเทศไทยได้เป็นอย่างดี ตลอดจนเป็นแรงผลักดันที่สำคัญในการต่อยอดและขยายผลการใช้งานนวัตกรรมดังกล่าวฯ ไปยังหน่วยงานเป้าหมาย อีกทั้งยังปัจจัยสำคัญของการเพิ่มการพึ่งพาผลิตภัณฑ์และบริการในประเทศอย่างยั่งยืนในอนาคต

***รูปที่ 5.5.1:*** *สกมช.เชิญชวนส่งผลงานเพื่อเข้าประกวดรางวัลผลิตภัณฑ์และบริการด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ “Thailand Cybersecurity Product and Service 2024”*

***รูปที่ 5.5.2:*** *ประกาศผลรางวัลผลิตภัณฑ์และบริการด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ "Thailand Cybersecurity Product and Service 2024"*

***รูปที่ 5.5.3:*** *การกล่าวเปิดงานโดยพล.อ.ต.อมร ชมเชย เลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)*

*รูปที่ 5.5.4: บรรยากาศการประกาศผลรางวัล*

*รูปที่ 5.5.5: บรรยากาศการเยี่ยมชมบูธจัดแสดงผลงานโดยผู้ทรงคุณวุฒิ*

*รูปที่ 5.5.6: บรรยากาศเวทีเสวนาแลกเปลี่ยนแนวคิดผลงานที่ได้รับรางวัล*

*รูปที่ 5.5.7: นายประเสริฐ จันทรวงทอง รองนายกรัฐมนตรี ร่วมแสดงความยินดีกับผู้ที่ได้รับรางวัล ณ อาคารสำนักงานใหญ่ บริษัท โทรคมนาคม*

## 5.6 บทสรุป

ที่ผ่านมาทีมวิจัยได้ดำเนินการเผยแพร่ประชาสัมพันธ์ผลงานวิจัยดังกล่าวฯ อย่างต่อเนื่องผ่านเวทีและกิจกรรมต่างๆ เพื่อสร้างการรับรู้ ความเข้าใจ และกระตุ้นให้เกิดการนำไปใช้งานจริงในหน่วยงานเป้าหมาย โดยมีการเข้าร่วมกิจกรรมสำคัญต่างๆ ดังนี้ (1) กิจกรรมสัมมนา “เพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์” จัดโดยสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) (2) กิจกรรมเสวนา ในหัวข้อ “Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซ

เบอร์ด้วยระบบอัตโนมัติ” ภายใต้งานประชุมวิชาการประจำปีของ สวทช. (NAC 2025) ครั้งที่ 20 (3) งานประชุมวิชาการและนิทรรศการประจำปีของเนคเทค (NECTEC-ACE 2024) ซึ่งเป็นเวทีสำคัญในการถ่ายทอดองค์ความรู้และนวัตกรรมโดยคนไทยสู่หน่วยงานผู้ใช้ และ (4) งานจัดแสดงนิทรรศการ BTFP Showcase 2024 ซึ่งเน้นการนำเสนอผลงานวิจัยที่มีศักยภาพต่อการใช้งานเชิงพาณิชย์และขยายผลสู่ระดับประเทศ นอกเหนือจากกิจกรรมข้างต้น ทีมวิจัยยังได้ดำเนินการสร้างความร่วมมือกับหน่วยงานที่เกี่ยวข้องอย่างหลากหลายทั้งจากภาคอุดมศึกษา หน่วยงานภาครัฐ และภาคเอกชน เพื่อขยายผลและนำร่องการใช้งานนวัตกรรมที่พัฒนาขึ้นในหลายบริบท อาทิเช่น หน่วยงานในภาคอุดมศึกษา และโรงพยาบาลต่างๆ ในสังกัดกระทรวงสาธารณสุข ซึ่งนับเป็นก้าวสำคัญต่อการวางรากฐานการพัฒนานวัตกรรมด้านการรักษาความมั่นคงปลอดภัยไซเบอร์ของประเทศไทยให้มีความยั่งยืน นอกเหนือจากนี้ การผลักดันนวัตกรรมจากห้องปฏิบัติการสู่การใช้งานจริง ไม่เพียงแต่ช่วยลดการพึ่งพาเทคโนโลยีจากต่างประเทศ แต่ยังเปิดโอกาสให้ประเทศไทยสามารถพึ่งพาตนเองได้ในระยะยาว ซึ่งเป็นการเสริมสร้างศักยภาพด้านการแข่งขันระดับภูมิภาคและระดับโลกได้อย่างมั่นคง

# บรรณานุกรม


[1] Gartner research, “Hype Cycle for Enterprise Networking”, 2020. Available URL: https://www.gartner.com/en/documents/3987266/hype-cycle-for-enterprise-networking-2020

[2] R. Jain and S. Paul, “Network virtualization and software defined networking for cloud computing: A survey,” *IEEE Commun. Mag.,* Vol. 51, No. 11, pp. 24–31, Nov. 2013.

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: Challenges and opportunities for innovations,” *IEEE Commun. Mag.,* Vol. 53, No. 2, pp. 90–97, Feb. 2015.

[4] R. Mijumbi et al., “Network function virtualization: State-of-the-art and research challenges,” *IEEE Commun. Surveys Tuts.,* Vol. 18, No. 1, pp. 236–262, 1st Quart., 2016.

[5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on software-defined networking,” *IEEE Communications Surveys Tutorials*, Vol. 17, No. 1, pp. 27–51, Firstquarter 2015.

[6] D. Kreutz et al., “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE,* Vol. 103, No. 1, pp. 14–76, Jan 2015

[7] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, and A. Meddahi, “NFV Security Survey: From Use Case Driven Threat Analysis to State-of-the-Art Countermeasures,” *IEEE Communications Surveys Tutorials*, Vol. 20, No. 4, pp. 3330–3368, Jul 2018.

[8] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, “Making Middleboxes Someone else’s Problem: Network Processing As a Cloud Service,” in *SIGCOMM’ 12*, 2012, pp. 13–24.

[9] P. Miller, and S. Bhatia, “NFV, Cloud and SDN: Moving beyond simple virtualization”, 2016. Available: https://networkbuilders.intel.com/docs/Genband_nfv_whitepaper.pdf

[10] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, “Trends and challenges in cloud datacenters”, IEEE Cloud Computing, Vol. 1, No. 1, pp. 10-20, Jul. 2014.

[11] G. Ramachandra, M. Iftikhar, and F. A. Khan, “A comprehensive survey on security in cloud computing”, *Procedia Computer Science*, Vol. 110, pp. 465-472, 2017.

[12] ETSI, “Network Functions Virtualization (NFV); Architectural Framework”, Dec 2014. Available URL: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[13] Docker, “Difference between VMs vs. Docker Containers”, Dec 2017. Available URL: https://dockerlabs.collabnix.com/beginners/difference-docker-vm.html

[14] Mahdi Daghmehchi Firoozjaei, Jaehoon (Paul) Jeong, Hoon Ko, Hyoungshick Kim, “Security Challenges with Network Function Virtualization*”, Future Generation Computer Systems*, Vol. 67, pp. 315-324, Feb 2017.

[15] Francois. Reynaud, Francois-Xavier Aguessy, Olivier Bettan, Mathieu Bouet, and Vania Conan, “Attacks against network functions virtualization and software-defined networking: State-of-the-art,” *in Proc. NetSoft*, Seoul, South Korea, Jun. 2016, pp. 471–476.

[16] Shankar Lal, Tarik Taleb, and Ashutosh Dutta, “NFV: Security Threats and Best Practices,” *IEEE Communications Magazine*, Vol. 35, No. 8, pp. 211–217, Aug 2017.

[17] Francois. Reynaud, Francois-Xavier Aguessy, Olivier Bettan, Mathieu Bouet, and Vania Conan, “Attacks against network functions virtualization and software-defined networking: State-of-the-art,” *in Proc. NetSoft*, Seoul, South Korea, Jun. 2016, pp. 471–476.

[18] T-NOVA project, “T-NOVA in a nutshell”. Dec 2016. Available URL: http://www.t-nova.eu/overview/

[19] T-NOVA: A Marketplace for Virtualized Network Functions paper

[20] T-NOVA project. (2016). “Deliverable D2.22 Overall System Architecture and Interfaces” Available URL: http://www.t-nova.eu/wp content/uploads/2016/03/TNOVA_D2.22_Overall_System_Architecture_and_Interfaces_v1.0.pdf

[21] T-NOVA project. (2016). “Infrastructure Resource Repository” Available URL: http://www.t-nova.eu/wp-content/uploads/2016/03/TNOVA_D3.2_Infrastructure_Resource_Repository_v1.0.pdf

[22] ANASTACIA project, “Advanced Network Agents for Security and Trust Assessment in CPS/IOT Architectures”, Feb 2020. Available URL: http://www.anastacia-h2020.eu/

[23] Ziegler, S., et al. "ANASTACIA: Advanced Networked Agents for Security and Trust Assessment in CPS IoT Architectures." Presented at the 2017 Global Internet of Things Summit (GIoTS), June 6-9, 2017, Geneva, Switzerland. DOI: 10.1109/GIOTS.2017.8016285

[24] ANASTACIA project, “D2.1 Policy-based Definition and Policy for Orchestration Initial Report”,

[25] ANASTACIA project, “D1.3 Initial Architectural Design”

[26] ANASTACIA project , “D4.6-FinalAgentsDevelopmentReport-v1.0”

[27] Open source MANO project, Sep 2021. Available URL: https://osm.etsi.org/

[28] An ETSI OSM Community White Paper. (2017). Available URL: https://osm.etsi.org/images/OSM-Whitepaper-TechContentReleaseTHREE-FINAL.PDF

[29] ETSI OSM Community. (2017). "OSM RELEASE THREE A TECHNICAL OVERVIEW." An ETSI OSM Community White Paper. October 2017. Accessed on October 5, 2023 Available at: https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseTHREE-FINAL.PDF

[30] OSM. (n.d.). "OSM Scope and Functionality." Accessed on October 6, 2023. Available at: https://osm.etsi.org/wikipub/index.php/OSM_Scope_and_Functionality#SDN_Controllers/WIMs_as_managers_of_Software-Defined_Network_Platforms

[31] Linux Foundation Project, “Open Network Automation Platform”, Nov 2017. Available URL: https://www.onap.org/

[32] ONAP. (2018). "ONAP Architecture Overview." Accessed on October 6, 2023. Available at: https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf

[33] ONAP. (2018). "ONAP CaseSolution Architecture" Accessed on October 6, 2023. Available at: URL: https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_ Architecture_0618FNL.pdf.

[34] Palantir Project. (2019). PALANTIR Conceptual Solution. Available at: https://www.palantir-project.eu/architecture/

[35] Palantir Project. (2019). PALANTIR Technologies. Available at: https://www.palantir-project.eu/technologies/

[36] Compastié, M, S., et al. (2023). PALANTIR: An NFV-Based Security-as-a-Service Approach for Automating Threat Mitigation. Sensors, 23(3), 1658. https://doi.org/10.3390/s23031658

[37] NTT and KDDI, “NTT and KDDI challenges for sustainable infrastructure transformation”, Apr 2021. Available URL: https://superuser.openstack.org/articles/ntt-and-kddi-challenges-for-sustainable-infrastructure-transformation/

[38] OpenStack Foundation. (2023). OpenStack Deployment and Operations Guide. Release 2020.2. Accessed on October 13. Available URL: https://pure-storage-openstack-docs.readthedocs.io/_/downloads/en/victoria/pdf/

[39] OpenStack Overview. (2023). OpenStack - A Modular Collection of Cloud Services. Accessed on October 13. Available URL: https://pure-storage-openstack-docs.readthedocs.io/en/zed_a/openstack-overview/section_modular-collection.html

[40] Google Cloud products. (2023). Accessed on October 20. Available URL: https://cloud.google.com/products#section-1

[41] Google Cloud Architecture Framework. (2023). Accessed on October 20. Available URL: https://cloud.google.com/architecture/framework

[42] AWS Implementation Guide: “Workload Discovery on AWS Implementation Guide”. Accessed on October 16. Available URL: https://docs.aws.amazon.com/pdfs/solutions/latest/workload-discovery-on-aws/workload-discovery-on-aws.pdf#architecture-overview

[43] AWS Whitepaper 2023: "ETSI NFVO Compliant Orchestration in the Kubernetes/Cloud Native World." Accessed on October 16. Available URL: https://docs.aws.amazon.com/pdfs/whitepapers/latest/ETSI-NFVO-compliant-orchestration-in-kubernetes/ETSI-NFVO-compliant-orchestration-in-kubernetes.pdf

[44] Microsoft Azure. Accessed on October 21. Available URL: https://www.isaca.org/resources/isaca-journal/issues/2020/volume-5/auditing-the-cloud-microsoft-azure

[45] Vmware-Cloud. Accessed on October 22. Available URL: https://www.vmware.com/cloud-solutions/vmware-cloud.html#resources

[46] The Snort Project. (2020, April 8). Snort Users Manual (Version 2.9.16). [Online]. Available: https://snortorgite.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf. [Accessed: Mar 29, 2023].

[47] Gavrilović, N., Ćirić, V., & Lozo, N. (2022). Snort IDS System Visualization Interface for Alert Analysis. Serbian Journal of Electrical Engineering, 19(1), 67-78. DOI: https://doi.org/10.2298/SJEE2201067G

[48] Wandhare, G. M., Gujar, P. S. N., & Thakare, V. M. (2015). Research article: Network intrusion detection technique for regular expression detection using DPI in ad-hoc.

[49] OISF, "Suricata User Guide Release 5.0.3." [Online]. Available: https://suricata.readthedocs.io/_/downloads/en/suricata-5.0.3/pdf/. [Accessed: April 7, 2023].

[50] Fadhilah, D., & Marzuki, M. I. (2020). Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against DoS/DDoS Attacks. In Authorized licensed use limited to: National

Science & Technology Development Agency. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 78-1-7281-7450).

[51] Provos, N. (2023). Honeypot Background. [Online]. Available: https://www.honeyd.org/background/. [Accessed: November 22, 2023].

[52] Provos, N. (2023). Honeypot concepts. [Online]. Available: https://www.honeyd.org/concepts/. [Accessed: December 22, 2023].

[53] González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. Sensors, 21(14), 4759. https://doi.org/10.3390/s21144759

[54] Wazuh Inc. (2023). Wazuh platform overview. [Online]. Available: https://wazuh.com/platform/overview/ [Accessed: December 16, 2023].

[55] Zeek, 2022. “The zeek network security monitor”. [Online]. Available: https://zeek.org/. [Accessed: April 25, 2023].

[56] The Cyber Center for Security and Analytics. (2020). Zeek Intrusion Detection Series: Lab 1 - Introduction to the Capabilities of Zeek. Document Version: 03-13-2020.

[57] OpenStack Tacker, 2019. [Online]. Available: https://docs.openstack.org/tacker/latest/. [Accessed: Dec 5, 2023].

[58] OpenStack Foundation. "Accelerating NFV Delivery with OpenStack Global Telecoms Align Around Open Source Networking Future" [Online]. Available at: https://object-storage-ca-ymq-1.vexxhost.net/swift/v1/6e4619c416ff4bd19e1c087f27a43eea/www-assets-prod/marketing/OpenStack-NFV-Print.pdf. [Accessed: Jan 5, 2024].

[59] Villacis, F., & Mejia, D. (2021). Cloud-based Network Function Virtualization Using OPNFV Components. 19th LACCEI International Multi-Conference for Engineering, Education, and Technology, July 21-23, 2021. [Online]. Available at: https://laccei.org/LACCEI2021-VirtualEdition/full_papers/FP533.pdf. [Accessed: Jan 5, 2024].

[60] https://docs.suricata.io/en/latest/public-data-sets.html

[61] CrowdStrike: Global Threat Report, 2024. [Online]. Available: https://www.crowdstrike.com/global-threat-report/

[62] Montida Pattaranantakul, Chalee Vorakulpipat, and Takeshi Takahashi, “Service Function Chaining Security Survey: Addressing Security Challenges and Threats”, *Comput. Networks*, Vol. 221, pp. 1-27, 2023, DOI: 10.1016/J.COMNET.2022.109484.

## ภาคผนวก ก.

## หนังสือเชิญเข้าร่วมและเป็นวิทยากรในกิจกรรมสัมมนา เพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายด้านความมั่นคงปลอดภัยไซเบอร์

**สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ**
๑๒๐ อาคารรัฐประศาสนภักดี ชั้น ๗ ศูนย์ราชการเฉลิมพระเกียรติ ๘๐ พรรษา ๕ ธันวาคม ๒๕๕๐
ถนนแจ้งวัฒนะ แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพมหานคร ๑๐๒๑๐ อีเมล saraban@ncsa.or.th

ที่ สกมช ๐๕๐๐/๔๔๓๕

๖ กันยายน ๒๕๖๗

เรื่อง ขอเรียนเชิญเป็นวิทยากรในกิจกรรมสัมมนา เพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์

เรียน ผู้อำนวยการศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

สิ่งที่ส่งมาด้วย (ร่าง) กำหนดการสัมมนาเพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัย ด้านความมั่นคงปลอดภัยไซเบอร์

ด้วยสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) มีพันธกิจในการส่งเสริมและสนับสนุนงานวิจัยและการพัฒนาเทคโนโลยีและองค์ความรู้ที่เกี่ยวกับการรักษาความมั่นคงปลอดภัยไซเบอร์ จึงได้มีการจัดกิจกรรมสัมมนาเพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ เป็นการสร้างเครือข่าย แบ่งปันข้อมูลระหว่างนักวิจัย ส่งเสริมความร่วมมือ ในการบริหารจัดการความมั่นคงปลอดภัยไซเบอร์ที่มีคุณค่าและเป็นประโยชน์ในการพัฒนาต่อยอดของนักวิจัย และองค์กรต่าง ๆ ด้านความมั่นคงปลอยภัยไซเบอร์อย่างมีประสิทธิภาพและประสิทธิผล

ในการนี้ สกมช. ขอเรียนเชิญ ดร.มลธิดา ภัทรนันทกุล เป็นวิทยากรนำเสนองานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ เพื่อส่งเสริมการแลกเปลี่ยนความรู้และแนวทางการพัฒนางานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ ในวันที่ ๑๗ กันยายน ๒๕๖๗ เวลา ๑๓.๑๕ - ๑๔.๑๕ น. ณ โรงแรมเซ็นทาราแกรนด์ เซ็นทรัลลาดพร้าว กรุงเทพมหานคร โดยมี (ร่าง) กำหนดการสัมมนาฯ ตามสิ่งที่ส่งมาด้วย และขอความกรุณาตอบรับตาม QR Code ท้ายหนังสือนี้ และได้มอบหมายให้ นายพสิษฐ์ ปานเพชร โทรศัพท์ ๐๖ ๓๐๘๗ ๗๓๘๕ เป็นผู้ประสานงาน

จึงเรียนมาเพื่อโปรดพิจารณา และขอขอบคุณมา ณ โอกาสนี้

ขอแสดงความนับถือ

พลอากาศตรี
(อมร ชมเชย)
เลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ

แบบตอบรับ

ศูนย์วิจัยความมั่นคงปลอดภัยไซเบอร์
โทรศัพท์ ๐ ๒๕๐๒ ๗๘๒๓
ไปรษณีย์อิเล็กทรอนิกส์ research@ncsa.or.th

https://dg.th/f5ste8rxbn

**(ร่าง)**

**กำหนดการสัมมนาเพื่อส่งเสริมงานวิจัยและพัฒนาเครือข่ายวิจัยด้านความมั่นคงปลอดภัยไซเบอร์**

**วันที่ 17 กันยายน 2567 เวลา 08.30 – 16.00 น.**

**ณ โรงแรมเซ็นทาราแกรนด์ เซ็นทรัลลาดพร้าว กรุงเทพมหานคร**

---

| | |
|---|---|
| 08.30 – 09.00 น. | ลงทะเบียน |
| 09.00 – 09.45 น. | พิธีเปิด<br>- ผู้ช่วยศาสตราจารย์ ดร.วราภรณ์ พรหมวิกร ผู้ช่วยเลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ กล่าวรายงานการจัดกิจกรรม<br>- พลอากาศตรีอมร ชมเชย เลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (ประธาน) กล่าวเปิดงานและบรรยายพิเศษ |
| 09.45 – 10.45 น. | เสวนา “นโยบายและยุทธศาสตร์ การสนับสนุนงานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์” โดย<br>- เลขาธิการคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ<br>- ผู้แทนสำนักงานการวิจัยแห่งชาติ (วช.)<br>- ผู้แทนสำนักงานสภานโยบายการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรมแห่งชาติ (สอวช.)<br>- ผู้แทนสำนักงานคณะกรรมการส่งเสริมวิทยาศาสตร์ วิจัยและนวัตกรรม (สกสว.)<br>ดำเนินรายการโดย ดร. รุ่งรัตน์ สร้อยเงิน |
| 10.45 – 11.00 น. | พักรับประทานอาหารว่างและเครื่องดื่ม |
| 11.00 – 11.30 น. | พิธีลงนามบันทึกความเข้าใจ (MOU) ว่าด้วยความร่วมมือด้านการส่งเสริมการพัฒนาทักษะการรักษาความมั่นคงปลอดภัยทางไซเบอร์ ระหว่างสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ กับ มหาวิทยาลัยราชภัฏร้อยเอ็ด มหาวิทยาลัยกรุงเทพ และมหาวิทยาลัยหอการค้าไทย |
| 11.30 – 12.15 น. | บรรยาย “งานวิจัยและเทคโนโลยีด้านความมั่นคงปลอดภัยไซเบอร์ที่น่าจับตามอง”<br>โดย ดร.ชาลี วรกุลพิพัฒน์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ |
| 12.15 – 13.15 น. | พักรับประทานอาหารกลางวัน |
| 13.15 – 14.15 น. | นำเสนองานวิจัยด้านความมั่นคงปลอดภัยไซเบอร์ |
| 14.15 – 14.30 น. | พักรับประทานอาหารว่างและเครื่องดื่ม |
| 14.30 – 15.00 น. | นำเสนอระบบส่งเสริมงานวิจัย พัฒนาเครือข่ายวิจัย และอุตสาหกรรมด้านความมั่นคงปลอดภัยไซเบอร์<br>โดย สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ |
| 15.00 – 15.30 น. | แลกเปลี่ยนความคิดเห็นและข้อเสนอแนะจากผู้เข้าร่วมกิจกรรมสัมมนา |
| 15.30 – 16.00 น. | กิจกรรมสร้างเครือข่ายและปิดงาน |

<u>หมายเหตุ</u>: รายละเอียดกิจกรรมอาจมีการเปลี่ยนแปลงตามความเหมาะสม

## ภาคผนวก ข.

## รายละเอียดกิจกรรมเสวนาหัวข้อเรื่อง Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ

NAC2025 20th NSTDA Annual Conference การประชุมวิชาการประจำปี สวทช. ครั้งที่ ๒๐

**หัวข้อ (ไทย)** Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ
**หัวข้อ (อังกฤษ)** Zero-Touch Services: The Future of Automated Cyber Threat Detection and Response
**วันที่และเวลา** วันพุธที่ ๒๖ มีนาคม ๒๕๖๘ เวลา ๐๙.๐๐ – ๑๑.๓๐ น.
**สถานที่** ห้องประชุม ๔๐๔ อาคารศูนย์ประชุมอุทยานวิทยาศาสตร์ประเทศไทย (อาคาร ๑๔ หรือ อาคาร CC) ถนนพหลโยธิน ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี

**เนื้อหาโดยย่อ**

แม้ว่าองค์กรหลายแห่งต่างตระหนักถึงปัญหาภัยคุกคามทางไซเบอร์ แต่ด้วยข้อจำกัดทางการเงินและความเชี่ยวชาญของบุคลากรที่มีอยู่อย่างจำกัด ทำให้องค์กรต้องตกอยู่ภายใต้สถานการณ์ความเสี่ยงที่จะถูกโจมตีได้ตลอดเวลา

“Zero-Touch Services” ได้ถูกพัฒนาขึ้น เพื่อช่วยอำนวยความสะดวกและลดภาระงานด้านความปลอดภัยให้กับองค์กร โดยการจัดเตรียมและให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ เพื่อช่วยวิเคราะห์ ตรวจสอบ และเฝ้าระวังภัยคุกคามที่อาจเกิดขึ้นกับองค์กร โดยเป้าหมายหลักในการวิจัยและพัฒนาแพลตฟอร์มฯ นี้ มีความมุ่งหวังที่จะสัมฤทธิ์ผล ๓ ด้าน ประกอบด้วย

(๑) ด้านต้นทุน โดยการช่วยลดค่าใช้จ่ายด้านการลงทุนและค่าดำเนินการต่างๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัยในองค์กร

(๒) ด้านการรักษาความปลอดภัย โดยการจัดเตรียมและให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยอัตโนมัติรองรับการให้บริการในรูปแบบ Security as a Services (SECaaS) สำหรับช่วยวิเคราะห์ ตรวจสอบ และเฝ้าระวังภัยภัยคุกคามทางไซเบอร์

(๓) ด้านประสิทธิภาพ การขับเคลื่อนด้วยเทคโนโลยี Network Functions Virtualization (NFV) มาช่วยยกระดับการบริหารจัดการเน็ตเวิร์กเซอร์วิสแบบรวมศูนย์และเป็นอัตโนมัติ เพื่อเพิ่มประสิทธิภาพการทำงานให้ดียิ่งขึ้น

ในการเสวนาจะเป็นการพูดคุย แลกเปลี่ยนเรียนรู้ประสบการณ์จากผู้ทรงคุณวุฒิ อัพเดทเทรนด์เทคโนโลยีใหม่ๆ และสร้างเครือข่าย ร่วมกันเตรียมพร้อมรับมือกับภัยคุกคาม และสร้างความปลอดภัยทางไซเบอร์ที่ยั่งยืนต่อไป

ประเด็นการเสวนาสำคัญ :-

- ภัยคุกคามไซเบอร์ในยุคดิจิทัลมีความท้าทายอย่างไรในยุคปัจจุบัน
- การตรวจจับอัตโนมัติสามารถช่วยให้องค์กรตอบสนองต่อภัยคุกคามได้เร็วขึ้นอย่างไร?
- บริการ Zero-Touch คืออะไร และทำไมมันถึงกลายเป็นเครื่องมือที่สำคัญในการป้องกันภัยคุกคามไซเบอร์

**ผู้ร่วมเสวนา**

| | |
|---|---|
| นางสาวมลธิตา ภัทรนันทกุล | นักวิจัย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ |
| นายแพทย์ภาณุพงศ์ ตันติรัตน์ | รองผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงสาธารณสุข |
| นายนรินทร์ รุณภัย | ที่ปรึกษาอิสระ ด้านความมั่นคงปลอดภัยทางไซเบอร์ ภาคเอกชน |
| รองศาสตราจารย์ปานใจ ธารทัศนวงศ์ | ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร |
| นายอังคาร ภุมรินทร์ | รองกรรมการผู้จัดการ บริษัท ที-เน็ต ไอที โซลูชัน จำกัด |

**ผู้ดำเนินรายการ**

นายเอกฉันท์ รัตนเลิศนุสรณ์
หัวหน้าทีมวิจัยความมั่นคงปลอดภัยสารสนเทศ
กลุ่มวิจัยการสื่อสารและเครือข่าย
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

กำหนดการ

| | |
|---|---|
| เวลา ๐๘.๓๐ – ๐๙.๐๐ น. | ลงทะเบียน |
| เวลา ๐๙.๐๐ – ๑๐.๐๐ น. | เสวนา Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ |
| เวลา ๑๐.๐๐ – ๑๐.๓๐ น. | พักรับประทานอาหารว่าง |
| เวลา ๑๐.๓๐ – ๑๑.๑๕ น. | (ต่อ) เสวนา Zero-Touch Services: พลิกโฉมการตรวจจับ เฝ้าระวัง และรับมือภัยคุกคามไซเบอร์ด้วยระบบอัตโนมัติ |
| เวลา ๑๑.๑๕ – ๑๑.๓๐ น. | ถาม/ตอบ |

## ภาคผนวก ค.

## หนังสือเชิญเข้าร่วมงาน BTFP Showcase 2024

ที่ สทช ๒๐๐๖/๓๘๗๕๖

สำนักงาน กสทช.
๘๗ ถนนพหลโยธิน ซอย ๘
แขวงสามเสนใน เขตพญาไท
กรุงเทพฯ ๑๐๔๐๐

๒ พฤศจิกายน ๒๕๖๖

เรื่อง ขอเรียนเชิญร่วมงาน "กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี"

เรียน ผู้รับทุนโครงการที่ได้รับการสนับสนุนจากกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ (กทปส.)

สิ่งที่ส่งมาด้วย กำหนดการงาน กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี

ตามที่ สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม แห่งชาติ (สำนักงาน กสทช.) โดยกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ จะได้จัดประชาสัมพันธ์ภารกิจของกองทุนวิจัยและพัฒนาฯ และกิจกรรม กทปส. ส่งต่อพลังหนุน ส่งทุนสร้างสิ่งดีดี นั้น

ในการนี้ กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ ขอเรียนเชิญท่านร่วมงาน "กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี" ในวันพฤหัสบดีที่ ๑๖ พฤศจิกายน ๒๕๖๖ เวลา ๙.๐๐ – ๑๙.๐๐ น. ณ อาคารเพิร์ล แบงก์ค็อก ถนนพหลโยธิน รายละเอียดปรากฏตามสิ่งที่ส่งมาด้วย ทั้งนี้ ท่านสามารถลงทะเบียนออนไลน์ตาม QR Code ที่อยู่ด้านล่าง เพื่อตอบรับเข้าร่วมงาน

จึงเรียนมาเพื่อทราบ และตอบรับเข้าร่วมงานตามวัน เวลา และสถานที่ดังกล่าว

ขอแสดงความนับถือ

พันทิภา สุทธินนท์

(นางพันทิภา สุทธินนท์)
ผู้อำนวยการส่วน รักษาการแทน ผู้อำนวยการสำนักกองทุนวิจัยและพัฒนา
ปฏิบัติการแทนเลขาธิการ กสทช.

สำนักกองทุนวิจัยและพัฒนา
โทรศัพท์ ๐๒ ๕๕๔ ๘๑๒๔
โทรสาร ๐๒ ๙๓๐ ๑๕๘๑
Email: btfp@nbtc.go.th

SCAN ME

กำหนดการ

งาน "กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี"

โดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ

วันพฤหัสบดีที่ ๑๖ พฤศจิกายน ๒๕๖๖ เวลา ๐๙.๐๐ – ๑๙.๐๐ น. ณ อาคารเพิร์ล แบงก์ค็อก

---

**ช่วงเช้า**

| | |
|---|---|
| ๐๙.๐๐ - ๐๙.๓๐ น. | ลงทะเบียนผู้ร่วมงาน และสื่อมวลชน |
| ๐๙.๓๐ - ๐๙.๔๐ น. | กล่าวต้อนรับ ความเป็นมาและวัตถุประสงค์ในการจัดงาน พร้อมรับชมวีดิทัศน์ Corporate VDO |
| ๐๙.๔๐ - ๐๙.๕๐ น. | พิธีเปิดงานอย่างเป็นทางการ โดย **ศ.คลินิก นพ.สรณ บุญใบชัยพฤกษ์** (ประธานกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ) |
| ๐๙.๕๐ – ๑๐.๐๐ น. | ถ่ายภาพร่วมกันบนเวที<br>ถ่ายภาพครั้งที่ ๑ ประธาน กสทช./กสทช./กรรมการบริหารกองทุนฯ/รองเลขาธิการ กสทช. รักษาการแทนเลขาธิการ กสทช.<br>ถ่ายภาพครั้งที่ ๒ ประธาน กสทช./กสทช./กรรมการบริหารกองทุนฯ/รองเลขาธิการ กสทช. รักษาการแทนเลขาธิการ/กสทช.ผู้บริหารสำนักงาน กสทช./ผู้อำนวยการสถาบันวิทยาการ/รก.ผอ.สำนักกองทุนวิจัยและพัฒนา/ผอ.ส่วน สำนักกองทุนวิจัยและพัฒนา |
| ๑๐.๐๐ – ๑๐.๔๐ น. | เยี่ยมชมนิทรรศการ กทปส. และ ๑๑ โครงการ เชิญท่านประธาน กสทช./กสทช./กรรมการบริหารกองทุนฯ/รองเลขาธิการ กสทช. รักษาการแทนเลขาธิการ กสทช./ผู้บริหารสำนักงาน กสทช./แขกผู้มีเกียรติ และสื่อมวลชน |
| ๑๐.๔๐ – ๑๑.๐๐ น. | สื่อมวลชนสัมภาษณ์ **ศ.คลินิก นพ.สรณ บุญใบชัยพฤกษ์** (ประธานกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ) |
| ๑๑.๐๐ – ๑๒.๐๐ น. | สัมมนาวิชาการ กสทช.<br>บรรยายหัวข้อ การออกแบบสำหรับทุกคนด้วยแนวคิด "ไม่ทิ้งใครไว้ข้างหลัง" เพื่อสร้างสังคมที่เข้าถึงได้ทุกคน โดย **นายต่อพงศ์ เสลานนท์** (กรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ ด้านการส่งเสริมสิทธิเสรีภาพของประชาชน) |
| ๑๒.๐๐ – ๑๓.๐๐ น. | บรรยายหัวข้อ "ความต้องการของผู้พิการทางสายตาที่มีต่อระบบเสียงบรรยายภาพประกอบรายการโทรทัศน์ โดย **ผศ.ดร. ภัทธิรา กลิ่นเลขา** (คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี สาขาวิชาเทคโนโลยีสื่อสารมวลชน มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย) |

กำหนดการ

งาน "กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี"

โดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ

วันพฤหัสบดีที่ ๑๖ พฤศจิกายน ๒๕๖๖ เวลา ๐๙.๐๐ – ๑๙.๐๐ น. ณ อาคารเพิร์ล แบงก์ค็อก

---

ช่วงบ่าย

| | |
|---|---|
| ๑๓.๐๐ – ๑๓.๐๕ น. | แนะนำกิจกรรม กทปส. ส่งต่อพลังหนุน ส่งทุนสร้างสิ่งดีดี |
| ๑๓.๐๕ – ๑๔.๐๕ น. | บรรยายหัวข้อ "กทปส. เส้นทางสู่จักรวาลนฤมิตร (Metaverse) โดย **ศ.คลินิก นพ.สรณ บุญใบชัยพฤกษ์** (ประธานกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ) |
| ๑๔.๐๕ – ๑๔.๑๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการสารคดี ชุด "SOME ONE หนึ่งในหลาย |
| ๑๔.๑๕ – ๑๕.๑๕ น. | เสวนาหัวข้อ "การจัดการปัญหาเมือง ด้วยแพลตฟอร์ม Traffy Fondue" โดย **รองศาสตราจารย์ ชัชชาติ สิทธิพันธุ์** (ผู้ว่าราชการกรุงเทพมหานคร) และ **ดร.ชัย วุฒิวิวัฒน์ชัย** (ผู้อำนวยการศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ / คณะกรรมการบริหารกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ) |
| ๑๕.๑๕ – ๑๕.๒๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการขยายผลศูนย์บริการอินเตอร์เน็ตเพื่อสังคมด้อยโอกาสในกลุ่มบ้านสาขาบนพื้นที่สูง |
| ๑๕.๒๕ – ๑๕.๔๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการ กระต่ายตื่นรู้ |
| ๑๕.๔๕ – ๑๕.๕๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการการศึกษาและพัฒนาหอฟอกอากาศอัจฉริยะโดยใช้พลังงานแสงอาทิตย์และระบบควบคุมดูแลผ่านเทคโนโลยีไอโอที เพื่อการลดฝุ่นละอองขนาดเล็ก (PM ๒.๕) ในพื้นที่สาธารณะ |
| ๑๕.๕๕ – ๑๖.๐๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการพัฒนาระบบแจ้งเตือนล่วงหน้า และระบุตำแหน่งการเผาในที่โล่ง โดยใช้โครงข่ายสื่อสาร อุปกรณ์เซ็นเซอร์ และอากาศยานซึ่งไม่มีนักบิน (Drones) |

กำหนดการ

งาน "กสทช. เสริมสร้างความรู้ ส่งทุนสร้างสิ่งดีดี"

โดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ

วันพฤหัสบดีที่ ๑๖ พฤศจิกายน ๒๕๖๖ เวลา ๐๙.๐๐ – ๑๙.๐๐ น. ณ อาคารเพิร์ล แบงก์ค็อก

---

| | |
|---|---|
| ๑๖.๐๕ – ๑๖.๑๕ น. | กิจกรรมแจกของรางวัล |
| ๑๖.๑๕ – ๑๖.๒๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำ โครงการการยกระดับการตรวจพิสูจน์ทางนิติวิทยาศาสตร์คดีความผิดทางเพศ ด้วยระบบปัญญาประดิษฐ์ |
| ๑๖.๒๕ – ๑๖.๓๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการระบบบริหารข้อมูลนำชมพิพิธภัณฑ์แบบพกพาเพื่อรองรับการใช้งานในพื้นที่จัดแสดงซึ่งมีข้อจำกัดด้านสัญญาณอินเตอร์เนต |
| ๑๖.๓๕ – ๑๖.๔๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการพัฒนาแพลตฟอร์มไอโอทีสำหรับการเฝ้าระวังไฟป่าและมลพิษทางอากาศด้วยเทคโนโลยีโลล่า |
| ๑๖.๔๕ – ๑๖.๕๕ น. | กิจกรรมแจกของรางวัล |
| ๑๖.๕๕ – ๑๗.๐๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการฝึกอบรม The Metaverse โลกเสมือนจริงที่กลายเป็นโลกสมจริง (Merge Reality Become Reality) |
| ๑๗.๐๕ – ๑๗.๑๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการยกระดับและปรับปรุงห้องปฏิบัติการเพื่อพัฒนาบุคลากรให้พร้อมเข้าสู่ยุคเศรษฐกิจและสังคมดิจิทัล |
| ๑๗.๑๕ – ๑๗.๒๕ น. | เชิญผู้รับทุนขึ้นเวที แนะนำโครงการขยายผลระบบติดตามและอุปกรณ์ช่วยเหลือผู้สูงอายุหรือผู้ป่วยในการทำกายภาพบำบัดผ่านเครือข่ายสื่อสาร สำหรับประโยชน์สาธารณะ |
| ๑๗.๒๕ – ๑๗.๓๕ น. | พิธีกรกล่าวสรุปงาน กล่าวขอบคุณ (จบงาน) |
| ๑๗.๓๕ – ๑๙.๐๐ น. | ประชาชนทั่วไปเดินชมนิทรรศการ |

## ภาคผนวก ง.

## จดหมายขอความอนุเคราะห์ร่วมทดสอบให้ระบบจากสถาบันดลตรีกัลยานิวัฒนา

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
เลขรับ ๔๒๐
วันที่ ๒ เม.ย. ๒๕๖๘
เวลา ๒๐.๒๐

ที่ อว๘๕๐๑/๐๓๙๗

สถาบันดนตรีกัลยาณิวัฒนา
๒๐๑๐ ซอยอรุณอมรินทร์ ๓๖
ถนนอรุณอมรินทร์ แขวงบางยี่ขัน
เขตบางพลัด กรุงเทพฯ ๑๐๗๐๐

๒๗ มีนาคม ๒๕๖๘

เรื่อง ขอใช้ระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ (Zero-Touch Services)

เรียน ผู้อำนวยการศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ตามที่ สถาบันดนตรีกัลยาณิวัฒนา ได้ให้ความสำคัญกับการรักษาความมั่นคงปลอดภัยของระบบเทคโนโลยีสารสนเทศ ทราบว่าศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติมีการพัฒนาระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ (Zero-Touch Services) ในการบริหารจัดการด้านความมั่นคงปลอดภัยทางไซเบอร์ตามความที่ทราบแล้วนั้น

ทั้งนี้ทางสถาบันฯ มีความประสงค์ขอใช้งานในระบบให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ (Zero-Touch Services) เพื่อใช้ในการรักษาความมั่นคงปลอดภัยของระบบเทคโนโลยีสารสนเทศของทางสถาบันดนตรีกัลยาณิวัฒนา ในการนี้ทางสถาบัน ฯ จึงใคร่ขอความอนุเคราะห์มายังศูนย์ ฯ เพื่อขอใช้ระบบดังกล่าว ทั้งนี้มอบหมายให้นายพงศธร ศรีวิเศษ เบอร์โทร ๐๘๙ ๑๑๕ ๔๗๙๒ email: pongsathon@pgvim.ac.th เป็นผู้ประสานงาน

จึงเรียนมาเพื่อโปรดพิจารณาให้ความอนุเคราะห์

ขอแสดงความนับถือ

(ผู้ช่วยศาสตราจารย์ ดร.อโณทัย นิติพน)
ผู้รักษาการแทนอธิการบดีสถาบันดนตรีกัลยาณิวัฒนา

☐ ผอ.
☑ ฝ่าย/หน่วยงาน STC
☐ เพื่อทราบ ☑ เพื่อพิจารณา
☐ เพื่อดำเนินการ ☐ เห็นแห่งทราย/ถือปฏิบัติ
☐ อื่นๆ.....
* หากหนังสือนี้ไม่ใช่งานที่อยู่ในความรับผิดชอบ ของท่านโปรดส่งคืนงานสารบรรณ "ด่วน"
๓ เม.ย. ๒๕๖๘

## ภาคผนวก จ.

## จดหมายขอความอนุเคราะห์ร่วมทดสอบให้ระบบจากสำนักงานสาธารณสุขลพบุรี

ที่ ลบ ๐๐๓๓.๐๑๔/๘๘๕

สำนักงานสาธารณสุขจังหวัดลพบุรี
ถนนพหลโยธิน ลบ ๑๕๐๐๐

๕ มีนาคม ๒๕๖๘

เรื่อง ขอร่วมทดลองใช้งานระบบให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ สำหรับตรวจสอบและเฝ้าระวังภัยคุกคามด้านไซเบอร์

เรียน ผู้อำนวยการศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ตามที่ สำนักงานสาธารณสุขจังหวัดลพบุรี ได้เข้าร่วมประชุมการนำเสนอระบบให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ สำหรับตรวจสอบและเฝ้าระวังภัยคุกคามด้านไซเบอร์ จัดขึ้นโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ เมื่อวันพฤหัสบดีที่ ๒๗ กุมภาพันธ์ ๒๕๖๘ สำนักงานสาธารณสุขจังหวัดลพบุรี เห็นว่าระบบดังกล่าว ช่วยเพิ่มขีดความสามารถในด้านการตระหนักรู้ ช่วยสร้างความปลอดภัย และหยุดการกระทำของภัยคุกคามทางไซเบอร์ ในการโจมตีระบบสารสนเทศ อีกทั้งระบบดังกล่าว สอดคล้องกับแนวทางการยกระดับความมั่นคงปลอดภัยทางไซเบอร์ ของกระทรวงสาธารณสุข นั้น

ในการนี้ สำนักงานสาธารณสุขจังหวัดลพบุรี มีความประสงค์ขอร่วมทดลองใช้ระบบให้บริการเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแบบอัตโนมัติ สำหรับตรวจสอบและเฝ้าระวังภัยคุกคามด้านไซเบอร์ จึงขอความอนุเคราะห์จากหน่วยงานของท่าน เพื่อขอทดลองใช้ระบบฯ เป็นระยะเวลา ๔ เดือน และหากมีผลการพิจารณาเป็นอย่างไร หรือมีข้อขัดข้องประการใด โปรดกรุณาแจ้งผลการพิจารณาได้ที่นางสาวภาวิณี ชูกลีบ ตำแหน่งนักวิชาการคอมพิวเตอร์ชำนาญการ หมายเลขโทรศัพท์ ๐๘๕-๙๐๙๓๒๒๒

จึงเรียนมาเพื่อโปรดทราบ และพิจารณาให้ความอนุเคราะห์ต่อไป

ขอแสดงความนับถือ

(นายณัฐ หาวารี)
นายแพทย์เชี่ยวชาญ(ด้านเวชกรรมป้องกัน)ปฏิบัติราชการแทน
นายแพทย์สาธารณสุขจังหวัดลพบุรี

☐ ผู้ค้อ. ☐ ร.ม.ต.อ. .....
☑ ฝ่าย/หน่วยงาน SEC
☐ เพื่อทราบ ☑ เพื่อพิจารณา
☐ เพื่อดำเนินการ ☐ เรียนเพื่อทราบ/ถือปฏิบัติ
☐ อื่นๆ .....
* หากหน่วยงานยังไม่ได้ดำเนินการ ขอความกรุณาส่งคืนงานภายในวันที่ 6 เม.ย. 2568

กลุ่มงานสุขภาพดิจิทัล
โทร. ๐-๓๖๖๘-๙๖๘๙ ถึง ๒๒๐
โทรสาร ๐-๓๖๖๘-๙๖๘๖ ต่อ ๑๑๒

Computer Networks 221 (2023) 109484

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Review article

# Service Function Chaining security survey: Addressing security challenges and threats

Montida Pattaranantakul [a], Chalee Vorakulpipat [a,*], Takeshi Takahashi [b]

[a] National Electronics and Computer Technology Center, Pathumthani, Thailand
[b] National Institute of Information and Communications Technology, Tokyo, Japan

ARTICLE INFO

*Keywords:*
Service Function Chaining (SFC)
Network Functions Virtualization (NFV)
Software-Defined Networking (SDN)
Traffic steering
Security threats
Vulnerabilities
Attack detection
Threat mitigation

ABSTRACT

Service function chaining (SFC) is a trending paradigm and it has attracted considerable attention from both the industry and academia because of its potential to improve dynamicity and flexibility in service chain provisioning significantly. SFC makes it easier and more convenient to compose on-demand service chains customized for application-specific requirements. In addition to SFC, network functions virtualization (NFV) and software-defined networking (SDN) are two other technology enablers that drive software-based service chain solutions. SFC leverages NFV for flexible deployment and for the placement of virtual resources and virtual network functions (VNFs); further, it employs SDN to provide traffic steering and network connectivity between the deployed VNF instances to form an application-specific service chain. Although SFC introduces many promising advantages, security is a major concern and a potential barrier for the widespread adoption of SFC technology. The integration of these technologies introduces a wide variety of security risks in the different levels of SFC stacks because SFC relies on NFV and SDN, and this results in a greater attack surface. Therefore, this survey aims to conduct a comprehensive analysis of SFC from a security perspective. To this end, we examine the SFC architecture in detail, including the design principles and relationships between other functional components, to obtain a clear understanding of SFC. The significant enhancements achieved by adopting SFC are highlighted. Further, we exemplify its deployment in several realistic use cases. Based on the SFC layering model, we analyze security threats to identify all possible risk exposures and establish a layer-specific threat taxonomy. We then systematically analyze the existing defensive solutions and propose a set of security recommendations to secure an SFC-enabled domain. Our goal is to help network operators deploy cost-effective security hardening based on their specific requirements. Finally, several open research challenges and future directions of SFC are also discussed.

## 1. Introduction

Recent years have witnessed an exponential growth in user demands and the diversity of services delivery because of the increasing popularity of mobile devices and the emergence of new networking technologies such as the Internet of Things (IoTs) and big data networking. Therefore, an unprecedented increase in the number of Internet users and network traffic usage has resulted in excessive capital expenditure and operational expenses.

One key problems is that network functions are tied to proprietary hardware devices, which are network devices that are static, time consuming, inflexible, and costly. Therefore, many enterprises and service providers have been investigating more effective solutions to speed up service deployments, improve operational efficiency, and reduce overall costs.

### 1.1. Service function chaining SFC related technologies

The emergence of network functions virtualization (NFV) [1,2], along with software-defined networking (SDN) [3,4] has led to massive changes in the networking paradigm, which has significantly changed the manner in which network services are deployed, managed, and operated. NFV concept was introduced by the European Telecommunications Standard Institute (ETSI) [5] to deal with traditional challenges. NFV has transformed the manner in which network operators build networks by utilizing standard IT virtualization technologies for decoupling network functions (such as, routers, firewalls, and load balancers) from dedicated hardware devices and abstract them into software elements known as virtual network functions (VNFs) that run on a standardized virtualized network infrastructure. The ultimate goals include

* Corresponding author.
*E-mail addresses:* montida.pattaranantakul@nectec.or.th (M. Pattaranantakul), chalee.vorakulpipat@nectec.or.th (C. Vorakulpipat), takeshi_takahashi@ieee.org (T. Takahashi).

https://doi.org/10.1016/j.comnet.2022.109484
Received 12 May 2022; Received in revised form 6 October 2022; Accepted 19 November 2022
Available online 24 November 2022
1389-1286/© 2023 Elsevier B.V. All rights reserved.

minimizing the hardware investment cost, enhancing the capacity of resource utilization, and accelerating service deployment for network operators. Further, the VNF can be created, configured, provisioned, scaled, and quickly terminated, which allows it to satisfy the diversity of service requirements and business models. Meanwhile, the SDN concept proposed by the open networking foundation (ONF) [6] aims to decouple the control plane from the data plane by removing the control plane from network hardware and implementing it in software. Therefore, moving the control plane to the software provides more flexibility in network control, configuration, and operation. A network administrator can dynamically configure network traffic from a centralized control console without physical access to network switches. In other words, SDN [7,8] is recognized as a complementary technology to NFV that allows full network control, such as forwarding capabilities and routing decisions, to support the dynamic reconstruction of service chains while providing traffic steering in a flexible and automated manner.

Laying down a foundation of service function chaining (SFC) allows leveraging NFV to build dynamic and virtualized network functions (VNFs) and uses SDN to steer traffic flows across different types of instantiated VNFs in an ordered manner [9–12]. Further, it is key to maintain the sequence and interconnection of two or more service functions for providing a complex service chain with specific functionalities. Advancements in SFC have made it possible for network operators and service providers to set up arbitrary service chains using different types of instantiated VNFs to satisfy their specific requirements. In [13–16], the authors exemplified several SFC use cases developed and widely deployed under different scenarios. It is gaining popularity because of the advancements in SDN/NFV technology, which allows for dynamic and flexible service chains, and thus enables network operators to add or remove VNF instances on demand to meet their specific customer requirements.

Owing to all the reasons mentioned above, SFC has found its interesting use case and has becomes an active research direction. Therefore, several ongoing studies from both industry and academia are attempting to address the unique and unprecedented challenges related to SFC, including (1) Interoperability challenge when combining dedicated hardware-based physical network functions to coexist with virtualized network function appliances [17]; (2) Implementation challenges in terms of defining the SFC model that needs to be adapted and fitted to the carrier-grade telecommunication networks [18,19]; (3) Path selection challenges to find the shortest feasible service function path for steering traffic flows across the intended service chain [20–22]; (4) Optimization challenges related to service chain composition and the placement of service functions in an SFC-enabled domain with the goal of minimizing service response time and maximizing resource usage and performance [23–28]; and (5) Security and dependability challenges to ensure that the predefined SFC policy and the intended security properties of concerns are translated correctly into the network flow classification with accurate packet forwarding rules, to ensure that they are enforced securely in the underlying data plane [29–33]. However, there are only a few studies in the literature that describe the comprehensive assessments of SFC. For example, Hantouti et al. [34] presented a high-level overview of different research problems related to SFC and investigated several key challenges that need to be addressed for realizing more reliable SFC operations. Further, in [12,35], the authors presented a survey on SFC architectures, the taxonomy of prior research related to the SFC concept and its implementations, and conceptual approaches based on SDN and NFV. In addition, some open research challenges associated with SFC environments were highlighted. Hantouti et al. [36] discussed the current SDN architectures and the various traffic steering techniques used to create a service chain. Bhamare et al. [37] conducted a survey on SFC implementation in contemporary networks, examined various approaches for optimizing network parameters (such as total deployment cost, end-to-end latency, network bandwidth, and resource allocation), and further outlined some SFC challenges and open research issues. Similarly, the authors of [38,39] conducted literature reviews to solve optimization problems in the SFC domain. This involved the study of a formal formulation and several optimization strategies, and then summarizing different approaches for solving these problems.

### 1.2. SFC security concerns

The abovementioned investigations focused on building a service chain, the description of the SFC architecture, the mechanisms for steering traffic flow across the desired set of service functions, the optimization approaches for optimizing resource usage and network performance, and concluding the remarks with some open research issues; none of the aforementioned studies examined SFC from a security perspective. To the best of the authors' knowledge, a security survey dedicated to a comprehensive threat analysis for the SFC-enabled domain is yet to be conducted. From the perspective of the client, several aspects remain unclear, *e.g.*, whether there is a gap between security risks and its adoption in the SFC-enabled domain, whether there are SFC security flaws that can affect the landscape of cyber defense, what types of security threats and vulnerabilities can be introduced, or how large is the attack surface it may present. We envision that the attack surface of the SFC can be increased significantly and multiplied for two main reasons: (1) SFC is considered to be in its nascent stage of development, and therefore, it always comes with a broad range of security challenges that range from high-level SFC policy specifications and enforcement to service chain orchestration and management, service function deployments and outsourcing, protocol stacks used to create a service chain, traffic steering, and packet forwarding, and so on. (2) The implementation of SFC fundamentally relies on two networking technologies: leveraging NFV to deploy a set of VNF instances on-demand and employing SDN to manage network connectivity among the instantiated VNFs and to steer traffic flows through them. Therefore, any type of security threat that appears in the NFV/SDN environment is inherited implicitly to the SFC-enabled domain.

### 1.3. Contributions

In this paper, we provide readers with a comprehensive SFC security survey. The main contributions of this survey are listed below:

- We provide a detailed background of the SFC to establish a comprehensive understanding of its design principles, architectural framework, and relationships between other functional components. Further, the advantages of SFC are discussed for highlighting the enhancements of its adoption towards overcoming the long-standing problems of the traditional networking model.
- Based on a thorough overview of the SFC framework and use cases, we conducted a comprehensive security analysis for identifying all potential attack surfaces and threat vectors introduced in the SFC-enabled domain. The outcomes of this analysis allowed us to further establish a layer-specific threat taxonomy, and it served as the basis of a holistic understanding on the SFC attack surface.
- A series of comparative studies were conducted considering security best practices. We aim to develop a set of recommendations on security countermeasures that should be followed to achieve higher levels of security protection against the aforementioned threats in the SFC-enabled domain.
- The open and emerging challenges in SFC security are discussed to highlight the directions for future research efforts and proposals.

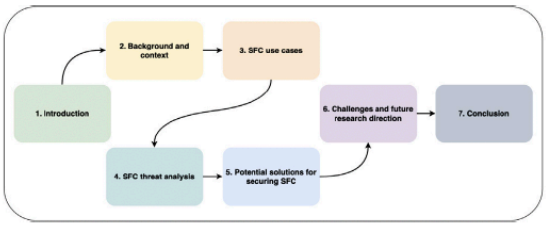



**Fig. 1.** High-level paper structure.

### 1.4. Organization

The remainder of this paper is organized as shown in Fig. 1. Section 2 outlines the problem statements faced by the traditional networking model and highlights the significant enhancements, *i.e.*, the concepts of SFC, its architectural framework, and the relevant components for SFC deployment. Further, the major terminology defined in the SFC standards and its relevance to NFV and SDN is discussed accordingly. In Section 3, practical implementations and use cases related to SFC are investigated to showcase the different possible solutions for deploying service chains in a realistic environment. In Section 4, we consolidate our understanding of the SFC attack surface based on a thorough comprehensive threat analysis using its architecture and deployment use cases; we aim to establish a layer-specific threat taxonomy. Section 5 investigates state-of-the-art security countermeasures in SFC. We discuss open research topics in the field of SFC and future research directions in Section 6. Finally, Section 7 summarizes the main conclusions of the study.

## 2. Background and context

In this section, we first address the existing issues of the traditional networking model, and then, we highlight the significant advancements related to adopting SFC for overcoming these issues. To this end, a set of properties is presented for comparing the traditional networking model and the SFC-enabled domain. The key findings are highlighted in Fig. 2. A brief description about the SFC standardized architecture and its key functional components are discussed to achieve a comprehensive understanding of the core concept of SFC. Finally, we provide detailed discussions on the related technologies like NFV and SDN, which act as key enablers for driving SFC solutions.

### 2.1. Problems with traditional networks

In principle, the concept of *"service function chain"* has been proposed a long time ago [40], which defines how physical networking devices are chained together to create a processing path. However, the hardwired service chains are very difficult to configure and deploy, and they are characterized by hand-crafted complexity [41]. As such, the traditional approach for implementing networks and services in hardware makes the network static, rigid, manual, and unable to respond the dynamic changes attributed to application requirements [42]. Further, this constraint limits the ability of the operator to deal with new use cases that have complex business models when considering competitive markets with rapid innovation at the application layer. In the following, we outline the different types of inefficiencies introduced by the traditional networking model.

#### 2.1.1. Topology dependencies

Traditional service deployments are tightly coupled to network topology. Several physical networking elements are fixed in place by topology and linked together using a complex and rigid set of configurations for creating a service path [42]. Network operators need to reconfigure the routing scheme and update the packet forwarding path to comply with this modification considering the new network topology information when there is a change in the service chain, *e.g.*, the addition/removal of service functions or the reordering the positions of service functions along the service path. Unfortunately, this results in nontrivial operational complexity and resource intensive and time-consuming efforts.

#### 2.1.2. Fixed service chains

The rapidly growing demand for cloud services and the many business-specific requirements have led service providers to develop capabilities for customizing their own network services to meet the specific requirements of users/tenants. In the traditional networking model, it was difficult for network operators to dynamically create a service chain that comprises different types of service functions. Further, it is difficult to steer traffic flows to go through a set of specified service functions at a granular level. For example, traffic flows that originate from different subscribers may be enforced by specific service chains (*e.g.*, video streaming, content delivery, security control, or monitoring purposes), which depend on their behaviors and characteristics. In [42], the authors reported that this can be attributed to the complexity in routing configuration and modification. The deployment model for service functions is tightly coupled with network topology and physical resources, which results in rigid and static deployments that make it difficult to achieve elastic scaling and dynamic SFC.

#### 2.1.3. Rigid and static nature in service delivery

A hardwired service chain comprising different types of service functions can be designed to meet diversified user requirements. However, it often is a complex task with nontrivial administrative burden; this task is also error prone and time consuming. Each networking component has to be individually configured with its own command syntax because all networking components are developed based on specialized hardware devices; they are linked together by a complex and rigid set of configurations to create a service chain [41]. Thus, the static chain configuration cannot flexibly deal with dynamic changes in user requirements, rapid service migration, and provisioning in the clouds, *e.g.*, the dynamic insertion of new service functions and scaling out the number of services for load balancing.

#### 2.1.4. Proprietary constraints

The traditional network suffers from several drawbacks including complex configurations, topology dependencies, physical hardware constraints, and vendor specific properties. These characteristics allow limited network operators with the necessary management capabilities to achieve cost-efficient deployment and provisioning of service chains [43]. A good example is where a vendor $X$ is better at serving application-layer firewall (firewall-L7), but vendor $Y$ provides a better performance for the network layer firewall (firewall-L3/L4). According to the traditional network limitations, network operators may be forced to choose only one hardware-based appliance from one specific vendor to process their workloads, which can lead to poor performances in the delivery of services and inefficient utilization of network resources.

### 2.2. Major benefits of SFC

The virtualization principles of NFV and SDN have made the implementation of SFC more practical; this has made it easy to address the aforementioned problems introduced by hardwired service chains. The improvements made by NFV/SDN enabled SFC domain are discussed in the following subsection.

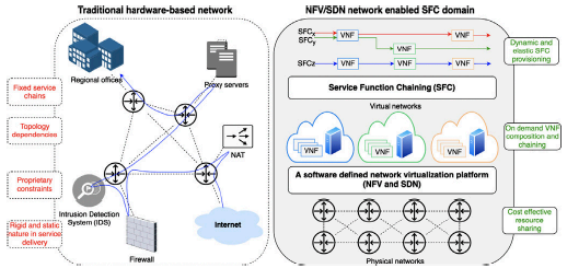



**Fig. 2.** The key differences between traditional and NFV/SDN enabled SFC domain.

#### 2.2.1. On-demand VNF composition and chaining

The approach employed for transforming hardware-based appliances to service-based software can significantly shorten the development lifecycle of network services. The combined strengths of an NFV/SDN-enabled SFC domain allow network operators to deploy various types of VNF instances (*e.g.*, virtual firewalls, load balancers, virus scanners, NATs) more quickly and flexibly with lower operational costs [2,35]. These deployed VNF instances can be scaled-in/out as required to support the growing demands and dynamic changes in customer requirements. The SDN [7] can help turn the rigid network configuration into a more flexible, programmable, agile, and robust network design, wherein traffic flow can be dynamically steered and flexibly managed to gain maximum performance benefits. Thus, different types of service chains can be built efficiently because of NFV and SDN, and this allows network operators to arbitrarily set up different service chains composed of several types of VNFs on demand to suit their business-specific requirements. For example, network operators can customize a service chain with a sequence of the following security functions to secure an enterprise or home network: firewall, intrusion/malware detection, and parental control. Similarly, the service chain can be sequentially customized as a serving gateway (S-GW), packet gateway (P-GW), and IP multimedia subsystem (IMS) to improve the performance of voice and video calls. In addition, a content delivery network can be added to the service chain for improving the quality and reliability of video streaming.

#### 2.2.2. Dynamic and elastic SFC provisioning

In the past, building a service chain comprising different types of service functions was a complex, nontrivial, and time-consuming process. Network operators required a couple of weeks to months for deploying service functions and configuring their networks to form/build a complete end-to-end service chain [43,44]. Interestingly, the inherently potential and recent advances in NFV and SDN have been recognized as key technological enablers for driving SFC solutions in terms of addressing flexibility, agility, scalability, and cost-efficiency requirements. Network operators now use NFV/SDN to build network infrastructure in a timely manner, dynamically deploy or scale the VNF instances on-demand, and customize their own software-based service chain as required without the need for specialized hardware and manual intervention to configure the network [12,37]. Thus, these advantages introduce significant improvements to the dynamic deployment and elastic provisioning of SFCs.

#### 2.2.3. Cost-effective resource sharing

Vendor locked-in constraints and inefficient use of resources can be addressed by leveraging the NFV/SDN enabled SFC domain. We can avoid the undesirable scenario of selecting proprietary monolithic service functions or appliances from a single-vendor stack if it is discovered later that these deployed services are monetary provisioning software difficult to amortize across applications even though their workloads offer natural opportunities to do so. In other words, NFV/SDN allows network operators to consolidate service functions from different vendors that run on a consolidated hardware platform and to manage them in a centralized manner [45,46]. Such network operators can quickly instantiate various VNFs, dynamically insert them into a specific service chain, and flexibly configure their network forwarding paths to meet application-specific requirements because these service functions are virtualized (namely VNFs); for example, security protection, load balancing, QoS, or flow control purposes. Another key advantage is that a common and widely used VNFs such as firewall and DPI can be shared and leveraged by multiple service chains. In addition, network operators can dynamically grow or shrink their capacity by scaling up/down these VNF instances based on traffic load. This significantly affects the higher utilization of network resources.

### 2.3. SFC standardized architecture

SFC has become an attractive research direction and has attracted considerable attention from the community of researchers, network operators, and network equipment vendors. The promising benefits of NFV and SDN offer a new way to design, deploy, and manage the VNFs and their network connectivity, and therefore, the implementation of SFC has become more flexible, programmable, dynamic, and cost efficient. Nowadays, SFC has been recognized as a key enabler for driving innovative changes in delivering end-to-end network services, especially when the deployment of service chains involves multidomains (*i.e.*, core data centers, edge computing nodes) and multitenant context (*i.e.*, tenant-specific application requirements). The initial concept of SFC was first introduced by the IETF SFC working group [11], which aimed to design an SFC architecture by separating the logical SFC overlay from the data plane and defining how traffic flow can be sequentially routed between the deployed VNFs in a service chain based on a predefined SFC policy. Thus, the key principles of these plane separations (*i.e.*, logical service function paths (SFPs) separated from the packet handling operations in the data plane, *e.g.*, packet forwarding) make the SFC architecture independent of the underlying network topology because no topological changes affect the SFC operations. In addition to the SFC architecture, we categorize its deployment into three standardization bodies as listed below.

- The Internet Engineering Task Force (IETF) — RFC 7665 [11] has contributed in defining the SFC problem, architecture, operation administration and maintenance, traffic steering techniques, and new protocols supported by SFC.
- Based on the IETF SFC specification, the ONF [47] proposed SFC deployment using SDN and OpenFlow. It defined the extension modules, *i.e.*, the northbound and southbound interfaces of the SDN controllers, which need to be implemented to support the controller's interaction, and it also specified an extended OpenFlow switch version (acting as service function forwarders (SFF)) to support the network service header (NSH) protocol for SFC encapsulation.
- The ETSI working group [7] has been motivated to integrate the SDN controller and combine SFC components as defined by the IETF SFC specification into the ETSI NFV architectural framework, and they defined a new terminology called VNF forwarding graph (VNFFG).[1] The objective is to provide SFC solutions that can dynamically adapt to the service chain conditions on demands. For example, rerouting the forwarding path, reordering or recomposing the VNFs in a service chain.

[1] VNFFG and SFC provide the same definitions: they describe the sequence of VNFs in a service chain. A single VNFFG may contain one or more network forwarding paths (NFPs). In SFC terminology, a VNFFG is considered as the SFC, VNFs are equivalent to service functions, NFPs are similar to SFPs, policies defined by network service providers are the rules used for the SFC classifiers, and virtual links are implemented by one or more SFFs.

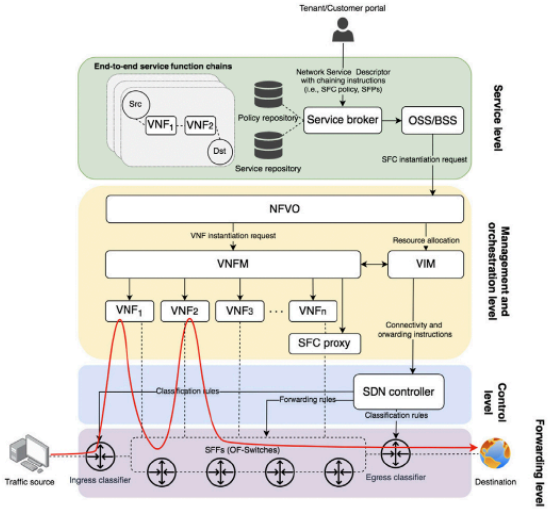



**Fig. 3.** The combined architecture of NFV and SDN towards supporting SFC deployment.

Fig. 3 illustrates the SFC deployment in the context of NFV and SDN. It involves several steps, ranging from (1) high-level SFC policy specification; (2) network resource allocation to create virtual machine instances to run the required VNFs; (3) VNF instantiation and placement to find the best locations for the optimal placement of VNF such that system performance (*e.g.*, resource utilization, power consumption, cost) can be optimized; (4) VNF selection and SFC composition, which select appropriate VNFs to be included or chained in a specific order considering the service chain request; (5) traffic steering and forwarding that interconnect the required VNFs and steer traffic flows through them based on the specified SFC policy; and (6) monitoring resources on-demand to examine whether it is necessary to scale out resources when utilization reaches a high threshold, or to scale in resources when the utilization is below the threshold. For the sake of simplicity, we classify the SFC architecture into four different levels: service (*i.e.*, end-to-end service function chains) level, management and orchestration level, control level, and forwarding level. We discuss the relationship of each level with SDN/NFV and examine in detail the relevant functional components used to support the SFC deployment.

**Service level.** The service level represents a logical connection of VNFs ordered in the form of service chains to fulfill the promised SFC deployment. Once network operators specify the SFC policy via the tenant/customer portal, this request is forwarded to the service broker and operations/business support system (OSS/BSS). The service broker receives and parses the SFC requests from users and identifies the policies and specific services (*i.e.*, end-to-end service function chains) that need to be delivered for achieving the requests. The OSS/BSS refers to a system that comprises specialized software and hardware for monitoring, analyzing, configuring, and managing operational activities within a network domain. It acts as a software system that provides additional management capabilities and reference points for the ETSI NFV MANO specification [48].

**Management and orchestration level.** This level aims to provide the centralized orchestration of network services, coordinate the underlying infrastructure resources, maintain the lifecycle management of both physical and virtual resources, and manage and orchestrate VNF instances (*e.g.*, create, update, scale, or terminate) to support the creation of an end-to-end network service chain. The NFV Orchestrator (NFVO) acts as a key component of the management and orchestration plane for providing a unified interface for managing virtual resources concerned with instantiation, provision, and termination of SFCs. Fig. 3 shows that it coordinates with the VNF Manager (VNFM) to instantiate the VNF instances such as virtual firewalls and virtual proxies, as specified in the service chain description. Further, it coordinates with the virtualized infrastructure manager (VIM) to allocate the resources required for creating the service chain.

**Control level.** Within the control level, there is an SDN controller that acts as a strategic control point in the SDN. It is responsible for managing network connectivity (*e.g.*, routing decisions based on the specified SFC policy), providing SFC forwarding path programmability, and communicating with the data-plane network elements (*e.g.*, classifiers, SFFs, and SFC proxies) to manage connectivity in the underlying virtual infrastructure. After receiving the SFC instruction from the VIM, the SDN controller consequently translates the related SFC policies into a specific service function path, adjusts the path according to the VNF status and overlay links, pushes the SFC classification rules into the traffic classifiers, and installs a set of forwarding rules for the data-plane elements. The advent of the SDN enables the network to be programmatically controlled, and thus, this results in a flexible and dynamic solution when steering traffic flows (*e.g.*, reactive or proactive installation of forwarding rules) through an ordered sequence of VNFs in a service chain.

**Forwarding level.** This level comprises several network elements involved in the SFC operations, which include traffic classifiers, SFFs (*i.e.*, SDN-enabled switches), SFC proxy, and a set of VNF instances. (1) The classifier acts as an entry/exit point of SFC-enabled domain, and it is responsible for packet classification by filtering different traffic types based on the SFC policy profiles. Further, it performs SFC encapsulation and directs the matched packet to the appropriate SFPs. In addition, the classifier can be classified into an ingress classifier (SFC ingress node) and an egress classifier (SFC egress node). Both classifiers are denoted as SFC boundary nodes. The ingress node aims to handle the traffic that enters the SFC domain, which is responsible for performing the SFC encapsulation to the packet. The egress node handles the traffic that leaves the SFC domain, and it removes service path information and other metatdata specific to the SFC domain; therefore, this process is called SFC decapsulation. (2) SFFs takes charge of forwarding the packet to the corresponding SFC elements; either the adjacent SFFs or the associated VNFs. The forwarding operations are based on the service path information contained in the SFC header.[2] (3) VNF is responsible for the specific treatment of the received packets. It can be either NSH-aware or NSH-unaware VNFs. (4) An SFC proxy is added and interfaced with an NSH-unaware VNF. Its role is to enable SFC communication between the SFFs and unaware VNFs, which indicates that it performs SFC encapsulation by removing NSH (SFC header) before forwarding the packet to the NSH-unaware VNFs and reinserting the packet with NSH before sending it to the next SFF along the associated service function path.

## 3. SFC use cases

Different service providers tend to customize their own network services to satisfy the specific requirements because of the growing demand for cloud services and the diversity of tenant-specific requirements. 5G network slicing is a good example that well illustrates the diversity of tenant requirements, wherein a specific slice independently defines a logical service chain that can comprise a particular set of VNF instances with a specific traffic forwarding path, and it may be distinct

[2] It is also known as network service header (NSH) [49]. NSH is added by the ingress classifier to carry or exchange service path information between the SFC elements and support SFC creation. NSH helps achieve topology independence by sharing metadata passed between nodes and by facilitating the exchange of context information between SFC elements.

from the other tenant slices. Therefore, network operators can dynamically provide the VNFs, arbitrary adjust their sequencing, and flexibly configure their network forwarding path based on subscribers and traffic information as required, using SFC in 5G networks. The positive impact of SFC has led to its popularity in SFC deployments. Nowadays, several types of SFC use cases have emerged in the networks. Some SFC use cases have been widely deployed in cloud datacenters to provide value-added services to improve security postures [13], or to increase the network performance or end-to-end latencies [50]. Meanwhile, some of them have been created on the specific layers of telecommunication networks. For example, deploying SFCs at the mobile core [14], radio access network [15] or edge gateway [16], with the common objective of enhancing system performance. Therefore, in this section, we intend to conduct comprehensive studies on several SFC use cases that are currently deployed across different networks, which include cloud datacenters, mobile networks, broadband networks, and radio access networks (RANs). Our objective is to provide some exemplary SFC deployment use cases in various environments. For each use case, we aim to study about its architectural model and concept (including concern metrics, criteria, policy specification, *etc.*) used for building a service chain in detail. These findings will help all involved parties to gain a clear understanding about a set of concern metrics necessary to implement or deploy SFCs in their networks. Furthermore, studying from the exemplary use cases can also help us gain insight into the deployment of service chains, which can be further adopted or customized to suit specific needs.

### 3.1. Cloud datacenter

The authors of [50] exemplified the possible deployment scenarios considering the SFC deployment in cloud datacenters. These scenarios were divided into two primary types of traffic flow: *North–South* and *East–West* traffic flow based SFC creations. Fig. 4 illustrates the different SFC deployment use cases between the North–South and East–West traffic flows. The first type defines the traffic originating from outside the datacenter (as shown in Fig. 4(a)), and it is associated with users/subscribers and is destined to applications/services hosted in the datacenter. For examples, deploying SFCs for security control, web services, or social networking access. The second type focuses on improving end-to-end latencies by leveraging a load-balancing technique (as shown in Fig. 4(b)), wherein one specific VNF instance can be initiated and scaled out into multiple VNF components represented as a pool of VNFs, each of which can be worked as a tier to perform a subset of functions. The SFC use case related to the North–South traffic is focuses on traffic flow that originates from outside the datacenter. Based on the given example shown in Fig. 4(a), flow originating from the user needs to sequentially traverse to (1) *web optimization control (WOC)* for optimizing the use of the WAN link bandwidth to improve user experience; to the (2) *edge firewall (EdgeFW)* used to filter traffic based on the specified policies, primarily focusing on threats external to the datacenter; and (3) *application delivery controller (ADC)*, which distributes traffic across a pool of servers for efficient resource utilization before reaching the requested applications. The East–West traffic based SFC use case employs the load-balancing concept to provide consistent latencies from one point in the fabric to another. From the given example shown in Fig. 4(b), *ADCs* can be scaled-out into multiple ADC components in which $ADC_1$ and $ADC_2$ act as load balancers responsible for the application server pool and DB server pool, respectively.

The authors of [13] exemplified other SFC deployment use cases primarily dedicated to network security. Therefore, enterprises can deploy a variety of security VNFs in the cloud datacenter to secure their networks, hosts, and endpoints. Sample security VNF instances include firewalls, intrusion prevention/detection system, deep packet inspection (DPI), application visibility and control (AVC), network virus and malware scanning, data loss prevention, DDoS mitigation, and TLS proxy. According to [13], the authors classified SFC use cases

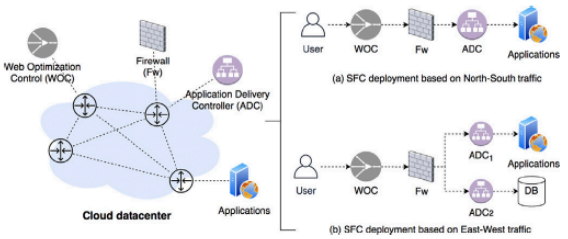



**Fig. 4.** An example of SFC deployment scenarios in cloud datacenter based on North–South and East–West traffics.

into four categories: (1) *Bi-directional traffic*, which defines the SFC to traverse forward and return the traffic forms to the same SFP with a similar set of VNFs. For example, a DDoS mitigator and firewall may need to receive the bi-directional traffic of a connection for examining the return traffic and maintaining the proper state of connection. (2) *Initiator and responder classification*, which classifies traffic flow based on the types of initiator and responders. For example, we can define the SFC and configure the DDoS mitigator to receive bidirectional traffic initiated from the Internet; however, we cannot use it for traffic initiated from the internal network. (3) *Network and application criteria*, wherein the SFC policies and network data (such as network interface port, source and destination IP addresses, and IP protocol) are considered to determine the service chain and its associated SFPs. For example, we can specify an SFC policy wherein all TLS traffic traverses to the TLS proxy. (4) *Inspection and scanning results*, in which an SFP of one specific SFC is determined based on the inspection results. For example, we define a particular SFC with two SFPs ($SFP_1$: AVC → firewall; and $SFP_2$: AVC → TLS proxy → firewall). In case a TLS session is detected by the AVC service function in $SFP_1$, the traffic flow is then switched to $SFP_2$ and passed through the TLS proxy to decrypt the TLS traffic for inspection before forwarding it to the firewall.

Another SFC use case for security is reported in [51]. This use case focuses on parental control. For example, parents may want to restrict their children from accessing specific sites and other Internet resources that are inappropriate. This control is implemented considering the children's device. By deploying the parental control function in a service chain, all incoming packets of the corresponding flow are examined to determine whether they contain the restricted information, *e.g.*, the URL or the source IP address subject to the parental control function. If a certain pre-configured pattern is matched, then the packet is blocked by this parental control function.

### 3.2. Mobile core networks

In mobile core networks, there are two segments between RAN and the Internet: (1) the evolved packet core (EPC), which is a packet-based mobile core that runs both voice and data on a 4G long time evolution (LTE) network, and (2) the Gi/SGi LAN, which is a network where service providers create several service chains comprised of various types of service functions to provide various homegrown and value-added services. Therefore, deploying a service chain in the mobile core network can involve many service functions such as the firewall, NAT, DPI, policy control, traffic and content optimization.

For SFC deployment scenario in the EPC, the authors in [52] exemplified an SFC use case that focuses on the data plane elements of a mobile network. As shown in Fig. 5, it includes the user equipment (UE) such as smart phones or tables, the LTE radio unit called enhanced NodeB (eNB), the S-GW with the mobility management entity (MME) to provide mobility, and P-GW, which provides connectivity from the UE to external data networks (*e.g.*, Internet, cloud datacenter, enterprise/corporate network, or SIP-based IMS networks) by being its point

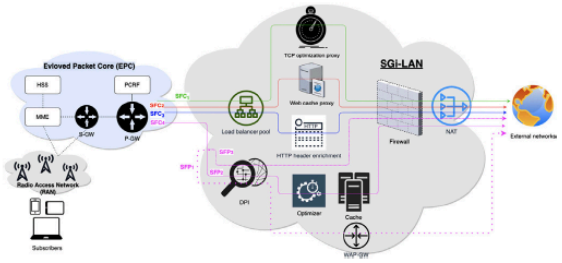



**Fig. 5.** Examples of SFC deployments in the mobile core network to improve user experience and value-added services such as TCP optimization, HTTP services, HTTP header enrichment, security control, and privacy protection, *etc.*.

of exit and entry of traffic. Other important components include the home subscriber system (HSS) and policy and charging rules function (PCRF). The HSS supplies the details of subscribers to other entities within the network; the IMS allows users to be granted or refused access to other services depending on their status; and the PCRF provides P-GW with the policy control and flow based charging control decisions to handle the UE's traffic.

In the Gi/SGi LAN network, the deployment of SFCs occurs in a LAN segment that links the mobile access network to the actual application platforms located in the carrier's datacenters or somewhere else on the Internet. The objective is to ensure a fair distribution of network resources based on agreed service policies, enhance the performances of service delivery, control security, and privacy protection. Considering the current deployments, many service functions are deployed in the mobile networks. Some of these service functions can be consolidated on the same commodity servers, whereas others are enabled in standalone boxes. In [14], the authors discussed general considerations and highlighted the SFC deployment use cases based on three functionalities:

First, deploying SFC to enhance the contracted quality of user experience, *e.g.*, *TCP optimization*. In mobile networks, TCP is considered the most widely used communication protocol for transmitting data between nodes. Over 90% of all traffic in mobile networks are based on TCP [53]. Thus, a set of network parameters such as latency, packet loss, available throughput, and bandwidth need to be addressed in advance to improve the performance of its transmission. An alternative solution for improving the performance is to deploy a TCP optimization proxy in the service chain for monitoring latency and throughput, and to dynamically optimize TCP parameters for each TCP connection with the goal of ensuring a better transmission behavior. A rudimentary service chain setup for TCP optimization is presented in Fig. 5 ($SFC_1$). The P-GW directs traffic flow to a load balancer pool, which then chooses an appropriate service function, *i.e.*, TCP optimization proxy to improve the TCP performance. The traffic is consequently traversed through a firewall for packet filtering, and then, it is passed to NAT to remap the private IP address space dedicated to the UE into a public IP address before sending the subscriber's traffic to external networks like the Internet.

Second, deploying SFC for *Internet HTTP services*. Mobile operators have started to introduce performance enhancement proxies (PEPs) to optimize network resource utilization owing to the increase in Internet traffic in mobile networks. These PEPs are integrated platforms that ensure the best possible quality of experience; their service functions include, but are not limited to, DPI, web and video optimizations, subscriber/service policy control, analytics and management support. A simple SFC model for mobile Internet traffic is illustrated in Fig. 5 ($SFC_2$). The service chain includes a load balancer that splits HTTP over TCP port 80 away from the rest of the Internet traffic; this HTTP traffic is then passed through the web cache proxy to help reduce round trip times and contribute to improve web page load times, firewall, and NAT before sending them to the Internet. Finally, SFC is deployed to support *HTTP header enrichment* for processing flows-based service-related policies. As indicated in Fig. 5 ($SFC_3$), operators can employ the HTTP header enrichment proxy to append information such as subscriber ID, international mobile equipment identity, and international mobile subscriber identity into the HTTP header used by downstream servers for identifying subscriber details.

In [54], the authors illustrated another example of SFC deployment in the Gi/SGi LAN network. In this scenario, it can involve many service functions, which include but are not limited to wireless application protocol gateway (WAP-GW), TCP and video optimizers, content caching, firewall, and NAT. As shown in the example in Fig. 5 ($SFC_4$), traffic flow originating from P-GW to the Internet can be categorized and divided into three corresponding SFPs ($SFP_1$: DPI → WAP-GW; $SFP_2$: DPI → optimizer → cache → firewall → NAT; and $SFP_3$: DPI → firewall → NAT). The DPI is deployed after P-GW to perform traffic classification. On the one hand, DPI continues to forward traffic to WAP-GW through $SFP_1$ once the WAP protocol is detected; on the another hand, it uses $SFP_2$ to process HTTP traffic by sequentially forwarding them to the optimizer to perform HTTP video optimization, HTTP content cache, firewall, and NAT. For other traffic types, it uses $SFP_3$ to enforce this traffic to pass through the firewall and NAT before it is sent to the external network, *e.g.*, the Internet.

The authors of [55] presented the SFC model, namely *CATENAE*'s architecture, which provides SFC in mobile networks for traffic steering and provides an overview on the possible deployment options in the LTE infrastructures. The goal is to provide SFC for minimizing the effect on current infrastructures, such as handling the actions of the opaque network function. Therefore, the proposed SFC model in LTE networks is defined as $SFC_m$: P-GW → PCRF → transparent/opaque functions → NAT. The P-GW is a UE for IP networks, and therefore, all IP traffic coming/going to the Internet or other operator's IP networks goes through the P-GW. The PCRF supports P-GW for data flow detection, policy enforcement and flow-based charging. The transparent function regulates traffic without modifying the packet header, and the opaque function operates in different way to modify the packets header values. NAT is deployed at the last node of the service chain to inspect and perform address translation. After the packets are processed by the specified service functions in the service chain, it is finally delivered to an Internet gateway to forward these packets to external networks.

### 3.3. Radio access network (RAN)

Virtualization appears to be invading all domains of the end-to-end networks, including cloud datacenter, mobile core networks, and mobile base stations. Thus, RAN is being used to achieve service agility and flexibility towards supporting the end-to-end 5G network slicing concept. This section intends to focus on the SFC deployment scenario in a RAN environment. Conceptually, RAN is considered a part of a mobile telecommunication system that connects individual UEs such as mobile phones, computers, or any remotely controlled machines and provides connectivity in its core network via radio connections. The RAN architecture includes the evolved NodeB (eNodeB), which acts as the main component of the RAN that facilitates the connectivity between UEs and other external networks including mobile core networks. The main functionalities of the eNodeB include radio resource management, routing of user plane packets towards the S-GW, packet compression and ciphering, security, and connectivity. Therefore, the SFC deployment use cases for the RAN environment are associated with eNodeB functionalities. One example of SFC deployment has been discussed in [15], wherein eNodeB functionalities were split into multiple specific elements such as radio resource control, radio link control, and packet data convergence protocol. These elements are then

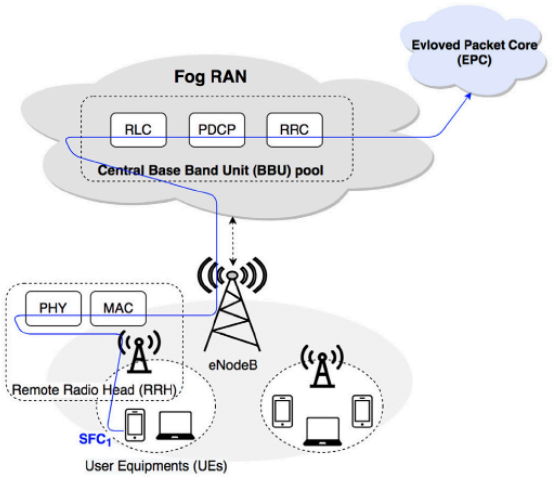



**Fig. 6.** An example of SFC deployment scenario in fog RAN.

virtually deployed in the fog network, which is called the *fog RAN*.[3] The remaining elements such as MAC and PHY are deployed at the remote radio head (RRH). Fig. 6 shows the SFC deployment in a fog RAN environment. The SFC mechanisms can be used to interconnect different functions in the correct order for meeting the specific requirements (*e.g.*, latency, compute and network capabilities).

### 3.4. Industrial edge networks

Not all applications are a good fit or best suited for deployment in the cloud datacenter. For example, some applications like the industrial IoTs have special communication requirements such as high reliability, low latency, flexibility, and security protection. One such example was highlighted in [16], wherein the control loop of a robotic arm was required to be less than or equal to 10 ms, which included the time for the system to acquire the sensor's signals, compute the input, and transmit this signal to the actuators. Thus, deploying the controller remotely on the cloud is not practical because the round trip time of signals is too long, and packet loss will lead to instability. Further, transmitting all raw data to the cloud is not economical, *i.e.*, VPNs or reserved bandwidth require considerably more expenditure. Further, industrial data are very sensitive, and thus, the data owners may not be willing to expose this data to public networks as sending such raw data to the cloud could lead to security risks. Therefore, the concept of deploying SFCs in industrial edge networks close to the data sources is a promising solution to address the aforementioned requirements.

SFCs related to time-sensitive applications can be deployed at the edge network for achieving a faster response. Further, the raw data can be processed with less end-to-end delay, and data privacy can be regulated under the control of users. An example of the SFC deployment scenario in industrial edge networks has been discussed in [16], wherein the authors consider the deployment of SFCs and other related service functions in a hierarchical manner. First, the manufacturing level groups a set of end devices such as sensors, actuators, and related equipment close to the edge gateway for reducing end-to-end latency. The examples of service functions involved in this level include aggregation, filtering, translation, consolidation, and analytics. These service functions focus on local data processing. Second, an edge cloud server with powerful resources is deployed at the factory level. The goal is to provide centralized operation management, task offloading control, big data desensitization, and logging. The service functions involved in this level include automation control, decision making, fault detection, predictive maintenance, machine learning, digital twins, *etc.*

[3] Fog RAN is the concept of exploiting cloud radio access network (C-RAN) and fog computing by splitting the RAN functionality such as eNodeB into several elements and virtualizing them on the fog virtualization infrastructure close to users with the objective to minimize transmission delay.

## 4. SFC threat surface

The potential to improve the flexibility of service provisioning and dynamic composition of a service chain using different types of service functions that can suit the current application-specific requirements has generated considerable interest in SFC from both industry and academia. Given these significant advancements, it is not surprising to see that SFC has been widely discussed and is being considered as a key enabler for driving an innovative change in delivering end-to-end services with unprecedented agility that allows network operators to deploy, manage, configure, or alter each service function in the chain specifically tailored to the demands of their individual subscribers or tenants. Based on thus point of view, the key findings indicate that SFC relies on two new networking paradigms: (1) leveraging NFV for flexible and on-the-fly creation and placement of virtual resources and VNF instances, and (2) employing SDN as a complementary technology for NFV, which can provide connectivity between the instantiated VNFs in a flexible and automate manner (NFV uses SDN as part of a SFC). In other words, the development of SFC involves multiple levels of abstraction (as illustrated in Fig. 3) spanning from the service (*i.e.*, end-to-end service function chains) level to the management and orchestration level, control level, and traffic steering and forwarding level when considering an implementation perspective. Indeed, the integration of these two key technologies has presented new opportunities for cyber criminals, and it has led to a variety of possible attacks in the different levels of SFC stacks. This includes a set of security threats and vulnerabilities inherited from NFV and SDN to a new set of specific threats and vulnerabilities related to the SFC-enabled domain. Although the open-layered structure of SFC offers promising solutions for flexible service chain deployment to meet specific business needs, it still unavoidably encounters a broad range of cyber security risks that could significantly influence the results with a larger attack surface.

A comprehensive threat analysis considering the SFC-enabled domain is required to obtain a holistic understanding on how SFC would fundamentally affect the landscape of cyber defense, how its implementation creates security concerns, and what kind of specific security risks it may introduce. To this end, we apply a level-based security analysis to identify all potential threats and vulnerabilities that can be incurred at each specific level of SFC stacks. This includes the service (*i.e.*, end-to-end service function chains) level, management and orchestration level, control level, and forwarding level. The major and most common security threats are shown in Table 1. The key findings can allow us to establish a layer-specific threat taxonomy, obtain a holistic view of the entire SFC attack surface, and provide appropriate security recommendations to protect against all identified security threats. Finally, we believe that our systematic approach in conducting this SFC threat analysis considering layer-specific architecture can help network operators and the involved parties to gain a more comprehensive view of the SFC attack surface. This would potentially empower them to find a solution for reducing the attack surface or eliminate the opportunities of an attacker, which will help increase the number of barriers when the attacker attempts to access the targeted assets.

**Table 1**
Threat analysis and potential attacks in SFC-enabled domain.

| Different levels of SFC stacks | Types of threats and vulnerabilities | Description |
|---|---|---|
| Service level | Untrustworthy in service-chain composition | Adversary disrupts VNF's availabilities and impersonate them for malicious intent |
| | Undependability in VNF chaining | Adversary manipulates the flow rules for deviating packet flows from their original service function path |
| | Ordering-property preservation | Adversary launches out-of-order traversal attack to disrupt the ordering properties (positions) of VNFs in a service chain |
| | Limited end-to-end service visibility | Multi-layer operations in SFC model makes it difficult to gain visibility into service chain deployment |
| | Inconsistent enforcement of security policies | Nature of multiple levels and inter-cooperation make it challenging to coordinate global security policies or incorporate trust in the SFC domain |
| Management and orchestration level | Attacks to centralized orchestration point | Successful attack on centralized orchestration point can result in harm or cause adverse effects to system performance |
| | Security risks from outsourced VNFs | Data breaches can be disclosed when outsourcing VNFs from third parties |
| | Vulnerabilities in VNF software | Adversary may exploit the VNF's software flaws to take control over the victims' system for malicious purposes |
| | Security issues posed by shared VNF package | Security risks such as backdoors and malware |
| | Security issues in VNF placement and chaining | Challenges in identifying the best location for VNF placement and chaining in the network, while ensuring security requirements are not violated |
| | DDoS attack | Service disruption and single point of failure |
| | Virtualized infrastructure threats | As the SFC implementation relies on two technologies (*i.e.*, NFV, SDN), threats that appear in the NFV/SDN environment can implicitly be inherited into the SFC domain |
| Control level | DDoS attack targets to controller | Centralized network intelligence and limited resources of the controller are the main reasons for attracting the DDoS attack |
| | Control channel congestion | Flow requests flooded by the victim SFFs can overwhelm the control channel bandwidth, which results in congestion |
| | Topology poisoning attack | Host location hijacking and link fabrication are two types of attacking models used by the adversary to poison the network topology |
| | Elevation of privileges attack | Adversary exploit API's vulnerabilities to gain privilege access to the controller |
| | Inconsistency of privacy | Dynamic configuration and decentralized environment with multi-controller and multi-tenancy supports are prone to configuration errors and policy conflicts |
| | Lack of data-plane acknowledgment | Without effective data-plane acknowledgment mechanisms, the controller and SFFs can have inconsistent views of network states and flow rule installation |
| Forwarding level | DDoS attack targets to SFFs | The SFFs have a limited TCAM space, which makes it an attractive attack surface for the adversary to launch a DDoS attack for saturating their resources |
| | Flow table overflow | Caused by the limited storage space of TCAM for storing flow tables, which has the potential risk of overflow |
| | Flow rule modification | SFF's software flaws can leave the door open for a flow rule modification attack |
| | Traffic interception | Adversary may get an opportunity to sniff or learn the contents of control messages when communications between the controller and SFFs are unencrypted |
| | Information disclosure | Transparency in SFC information and metadata sharing can result in information disclosure |
| | Packet injection | Vulnerability in OpenFlow protocol could allow adversary to maliciously inject packet-in messages with fake MAC or IP addresses |
| | Replay attack | Adversary records legitimate flows and replay them afterwards to pass validation |

### *4.1. Security issues in service level*

Although SFC facilitates an open architecture with many flexible solutions for supporting the development and deployment of service chains, a gap still exists between the high-level SFC policy specifications and their enforcement at the lower level. A good example is that network operators may have no idea whether the intended VNF instances are chained as expected, or if the packet flows of a particular service chain are forwarded sequentially to the intended and legitimate VNFs strictly compliant with the specified SFC policy. These intrinsic limitations lead to significant security threats, which are analyzed below.

- *Untrustworthy in service chain composition:* One of the most significant benefits of the SFC is dynamic and on-demand service function chain composition, which means that VNF instances such as firewalls, load balancers, and anti-virus scans provided by different vendors can be composed or scaled on demand to meet application-specific performance needs. From a security perspective, these VNF instances must be authenticated beforehand and the trust relationship between them should be established. In other words, the related service chain will be compromised if the VNFs are compromised. However, the authors of [29,56] pointed out that verifying the authenticity of these VNFs in a service chain is yet to be addressed in detail. In fact, network operators may not even notice and have the right to conduct an examination if a set of predefined VNFs involved in a service chain is compromised, or if there exists any malicious VNF instances. An attack scenario is discussed in [57,58], wherein an adversary launches a DDoS attack to make legitimate VNFs unavailable and manages to impersonate these victim nodes by receiving and responding to the packets on their behalf. This scenario misleads the network nodes including the controller and other VNF instances to communicate and synchronize with the adversary instead of the legitimate ones. The lack of authenticity can introduce major risks and ultimately jeopardize the entire service chain.

- *Undependability in VNF chaining:* One of the SFC features is to provide the network operators with the ability to build a complex service chain comprising several types of VNF instances and steering them together in a sequential manner as specified in the SFC policy. However, SFC can be vulnerable to new pitfalls that can be exploited by attackers. An example of an attack model is presented in [29,59,60], wherein the attackers target the underlying network devices (*e.g.*, switches and routers) using rule modification attacks to tamper flow rules with the intent to violate the specified SFC policies by deviating the packets from their original SFPs. If a rule has been defined, the traffic flow must traverse through security functions like firewall or IDS; however, the attackers can maliciously modify the flow rule by intending to bypass the packets that belong to this flow to evade from these security functions. Considering the gap between different abstraction levels of SFC stacks, it is challenging to examine the actual SFC deployment and ensure that the packet associated with this service chain is indeed traversed to all intended VNFs in the correct order, while respecting the predefined SFC policy.
- *Ordering property preservation:* The delivery of an end-to-end service chain often requires traversing through various types of VNFs in a sequential manner as specified in the SFC policy. These VNFs may be delivered within the tenant boundaries or shared amongst many tenants, groups of users, or applications; however, one key observations is that these VNFs involved in a service chain are deployed independently, and each of them is responsible for the specific treatment of the received packets and is completely unrelated to the others [42]. However, many VNFs in a service chain are clearly defined in a strict order, and it is impossible for network operators to verify their ordering properties (positions) in the actual SFC deployment. Out-of-order traversal attack [29,59,61,62] is a type of attack that can disrupt the ordering properties of VNFs in the service chain substantially. The results of our survey indicates that there are few research works that have attempted to provide a self-verification mechanism for examining the correct sequence of service functions (or correct packet traversal) in a service chain.
- *Limited end-to-end service visibility:* The implementation of SFC involves several steps: from high-level SFC policy specification, to network resource allocation, VNF instantiation and placement, VNF selection, SFC composition, traffic steering and forwarding. For each operational step, it is necessary to correlate and exchange information across heterogeneous entities within the SFC domain. However, this can potentially lead to increased complexity in network management and operations, especially when service function chains span across multiple cloud datacenters or administrative boundaries. Therefore, it is extremely difficult to infer the root causes of failures or to gain insight into networks and correlating service-impacting disruptions because of the interleaved dependency between these functional components [42].
- *Inconsistent enforcement of security policies:* In the context of SFC, the dynamic nature of service composition, provisioning in multiprovider environments, and topological independence from the underlying forwarding topology can greatly increase the complexity of the security policy enforcement. Researchers are aware about the importance of security policies and their enforcements in the underlying infrastructure, especially when the service chains are built over multiple cloud datacenters of third-party infrastructure providers or across administrative boundaries. Although a significant number of works associated with security policy enforcement have already been discussed in virtualized environments like NFV [1] and SDN [63], this issue needs to be reconsidered in the context of SFC as well. The authors of [37] indicated that a specific service chain can be formed with one or more SFPs that must be followed by the packets. Each path may be composed of an ordered set of service functions served by various cloud service providers. Each provider may have different terms related to security regulation and policy enforcement, and they are independently defined or implemented without sufficient collaboration with others. However, joining factors related to multicloud environments, multiple types of service functions, and multivendor integration make it extremely difficult to coordinate global security policies and hard to build trust relationships between the service components across multiple SFC-enabled domains.

### 4.2. Security issues in management and orchestration level

According to the SFC architecture, once network operators specify high-level descriptions about network services and their chaining policy through a portal, this information is parsed to the respective lower levels: (1) *service level*, which aims to identify an end-to-end service function chain and policies that need to be deployed for achieving the requests; (2) *management and orchestration level*, in which the NFV is used to provide resource allocation, VNF instantiation, and placement without considering network connectivity; (3) *control level*, wherein the SDN controller provides the full network control capabilities (*e.g.*, configuring, controlling network connectivity, and providing service function path programmability along the instantiated VNFs according to the specified SFC policy); and (4) *forwarding level*, in which network devices such as routers and switches are located to take charge of steering the traffic flow based on a quick lookup of the forwarding tables. Even though NFV and SDN play significant roles in the SFC domain which helps increase agility, flexibility, and scalability in delivering application-specific service function chains with cost efficiency, integrating NFV and SDN to drive SFC solutions is challenging because of the variety of intertwined network elements involved and the complexity of cross-boundary interactions. Specific threats considering the management and orchestration level are discussed below.

- *Attacks to centralized orchestration point:* Accounting for the core module in orchestration level, the NFV management and orchestration functional blocks are considered the brain of all operations that play a vital role in providing the full end-to-end lifecycle management of network resources and services. These include virtual resource allocation, VNF on-boarding, instantiation, scaling, and termination. Therefore, it can become a single point of failure or represents an attractive target for attackers. As pointed out in [1,64,65], any compromised operations can lead to a failure of the entire system or yield a wider network impact. For example, if attackers gain access to this centralized orchestration point, all network services will be under their control. Then, they can freely reconfigure network resources, disrupt the lifecycle management of network services or individual VNFs, monitor data transactions, and further acquire confidential and sensitive information of the victims.
- *Security risks from outsourced VNFs:* To form a service function chain, network operators can outsource VNF instances from third-party vendors or cloud service providers to reduce operational costs. Thus, instead of deploying all VNFs in their own infrastructure, they can instantiate VNFs on the infrastructure of third parties who are ready to interconnect in the chain. This helps avoid the cumbersome operations and maintenance tasks. Despite these advantages, numerous security risks can occur when outsourcing VNFs. First, because the outsourced VNFs and related policies are fully managed and controlled by third-party vendors or cloud providers, it can provide a good opportunity for malicious providers to take control of the services and compromise confidentiality, or peek into the traffic and maliciously modify/drop packets for unlawful gain [46,66]. Second, outsourced VNFs rely on contractual relationships and co-opetitive settings

in which most user data, operator policies, and VNF codes need to be disclosed to the third parties; such an event can ultimately lead to a potential compromise and data breaches in a service chain [67].

- *Vulnerabilities in VNF software:* An important contribution of SFC is to allow various VNFs to be chained together to form an end-to-end service that allows carriers to benefit from the virtualized software-defined infrastructure. These VNFs are network functions embedded in software removed from network peripherals and hosted on virtual machine to offer as services. From a security perspective, they could be vulnerable because of various types of software flaws [1,65,66]. For example, remote attackers can leverage software flaws to bypass the intended firewall restrictions via crafted packets (CVE-2012-2663), exploit incorrect security configurations in the access control levels of firewalls (CVE-2020-7278), or attempt to exploit stack-based buffer overflow to execute arbitrary code within the victim VNFs, such as Snort (CVE-2006-5276), DNS proxy (CVE-2017-8390), and antivirus scan (CVE-2010-3497) [68]. Despite these types of software flaw being patched, they provide an initial insight into the possible security issues that threaten network functions when they are deployed in a virtualized environment.
- *Security issues posed by a shared VNF package:* A VNF package is a ZIP file that contains all essential information for managing the lifecycle of a VNF. This package is used as reference during service chain deployment. A typical VNF package includes (1) a VNF descriptor that contains all information used by the centralized orchestrator for deploying the VNF; (2) a VNF software image used to launch VNF instances on demand; (3) a metadata file that provides additional information to the centralized orchestrator for interpreting the content of the package; and (4) a script that contains all instructions required for instantiating, configuring, and starting a VNF on the VMs or containers instantiated during the lifecycle. The network operator can either customize their own VNF package from scratch or download the available VNF package store in public cloud repositories from third-party vendors or service companies for support. For example, Amazon offers a public repository wherein network operators can download an image, *i.e.*, the Amazon Machine Image, to launch the instances; Cisco provides a VNF software image repository to allow network operators to store, update, or download the software images as required. Although a pre-packaged VNF solution facilitates the ease of VNF deployment, the exchange of resources (that contain all software components and configurations) poses three types of security risks [69]: backdoors and leftover credentials, unsolicited connections, and malware. In addition, the authors of [70] pointed out that it is easy to tamper with VNF images by inserting some malicious code or malware while it is being uploaded to the database or being transferred from the database to a remote compute node.
- *Security issues in VNF placement and chaining:* These problems are still challenging and have not been investigated sufficiently. On receiving an SFC request, the centralized orchestrator needs to determine the best approach to deploy the intended VNF instances and select the best optimal path for chaining them to make it consistent with the specified application/tenant requirements. Therefore, it is possible to migrate the VNF instance to a new location in the network, instantiate a new VNF dedicated to a specific service chain, or share the existing VNF with other service chains. However, considering the existing approaches for VNF placement and chaining from the perspective of resource optimization, this can lead to deployment solutions that do not comply with well-known security-related best practices and recommendations [71]. In other words, it is challenging to find the best location for VNF placement and chaining in the network while ensuring that the service level agreement (SLA) between the cloud service providers and customer requirements is not violated [72]; this includes security policies and security best practices. Thus, the ignorance of these concerned properties can cause significant damage (including boundary breaking, evading security control, and user privacy violation) and security breaches in a service chain.
- *DDoS attack:* This type of attack aims at targeting the centralized orchestrator by utilizing multiple compromised hosts to flood the network with heavy amounts of unreal traffic. The goal is to overwhelm the resources of the orchestrator (*e.g.*, memory, CPU, and network bandwidth) to make them indefinitely suspended or temporarily unreachable by other service chain requests. As pointed out in [73,74], the centralized orchestrator is considerably more vulnerable to a DDoS attack because it is built using a software-based approach. Any result from the DDoS attack potentially leads to service disruption and a single point of failure in the orchestrator.
- *Virtualized infrastructure threats:* The implementation of SFC relies on leveraging NFV to build the network forwarding and other networking functions in the virtualization environment and employing SDN for controlling and managing the connectivity between the instantiated nodes towards forming an application-specific service chain. Based on this networking paradigm, all underlying resources are shared among various network elements in the same NFV stacks; these resources can be scaled dynamically to meet fluctuating needs and demands to achieve highly efficient resource utilization and better service at a lower cost. However, sharing infrastructure resources creates new challenges and security issues. Indeed, some computing resources do not fully support the multitenancy requirements demanded by end users, which causes the components to act abnormally and finally cause data leakage or unavailability of service requests [75]. Another critical security issue is hypervisor vulnerabilities [1,66,69,76]. The hypervisor is responsible for communication and isolation of different entities that create the virtual resources for VMs and VNFs, and therefore, if it lacks sufficient security awareness, then attackers may be able to alter the hypervisor to gain malicious control. Hypervisor hijacking [77] is another type of attacking model that allows attackers to take control and gain access to all VMs and VNF instances running on top of the same hypervisor. Further, the attackers may exploit bad configurations or design flaws in the hypervisor to break out its isolation [70], exploit the hypervisor's misconfiguration, or even launch a DoS attack [78] in one VM/VNF to exhaust all available resources. This result in a scenario wherein the hypervisor cannot support more requests. Therefore, other VMs/VNFs that run on the same host machine are eventually starved for resources.

### 4.3. Security issues in control level

Once VNF instances are initiated, the SDN controller links these VNFs to form an end-to-end application-specific service chain. The SDN controller is designed to make the network more flexible and programmable, which allows network operators to gain greater control of the traffic flow based on the SFC requirements and ease the management of the entire network from a single centralized location. However, increased flexibility and centralized control can generate security risks from various types of attacks. We highlight a list of security threats and vulnerabilities arising in the control level.

- *DDoS attack targeting the controller:* The SDN controller acts as the brain of network operations, and it is responsible for performing the control decision tasks and configuring the underlying forwarding devices in a logically centralized manner. Therefore, it leverages network intelligence to manage the traffic flow, which makes it attractive for attackers to launch DDoS attack and overwhelm

the controller resources or render the controller unavailable to serve any further requests. This scenario can cause the controller to become a single point of failure or a performance bottleneck with an increase in the processing overhead. Examples of DDoS attack are discussed in [79–81], wherein attackers compromise victim SFFs like switches and force them to send a large number of anomalous packet-in messages to the controller. When SFFs receive the incoming packets that do match any flow rules installed in the flow tables of the switch, these packets trigger table misses. The ingress switch is then enforced to buffer the table-miss packets, while sending packet-in messages to the controller to ask for action to be performed. In response, the controller processes the incoming packet-in message, computes the flow rules, determines the corresponding instruction, and then transmits an appropriate flow-mod messages back to the SFFs. Assuming that the controller receives a large number of flooded-flow requests, this can dramatically affect the overall network performance and lead to congestion in the control channel, controller resource saturation, delay in the installation of new flow rules, overloading of the buffer memory of SFFs, and the generation of amplified traffic to occupy the data-to-control plane bandwidth.

- *Control channel congestion:* A centralized controller communicates with the underlying SFFs to exchange the control messages (*i.e.*, routing decision) via the control channel; however, the flow requests flooded by the victim SFFs can cause excessive congestion on the bottleneck link, which can overwhelm the shallow switch buffer and result in a large number of packet drops [79,82]. If the SFFs encounter massive table-miss packets, then they will send the packet-in message containing the entire packet information to the controller via the control channel to request appropriate flow actions. Therefore, an increasing number of flooded-flow requests can ultimately lead to heavy bandwidth consumption and congestion bottlenecks in the control channel. Another attack that disrupts the control channel is described in [83], wherein the authors discussed an idea of exploiting the control channel by generating well-crafted data traffic to implicitly interfere with the delivery of the control traffic. This can cause real-time control messages delivered between the controller and the switches to be significantly delayed or dropped.
- *Topology poisoning attack:* Another critical challenge that appears in the SDN controller is a topology poisoning attack, wherein attackers intend to spoof the controller view of the network topology and deceive it into accepting a spurious topology. This attack causes a false flow rule installation in the SFFs, which makes it possible to deviate the traffic flow from its original service function path, which is chained and processed by the legitimate intended VNF, into a new path that has malicious nodes under the control of attackers. Hong et al. [58] highlighted two ways to poison the network topology: host location hijacking and link fabrication. In the former, attackers launch a DDoS attack [57] to render the legitimate VNFs unavailable, and they then impersonate these victim nodes by receiving and responding to the packets on their behalf. A major causes of this attack is a lack of host authenticity checking. In the latter, attackers attempt to create fake links in the network by manipulating the propagation of the link layer discover protocol (LLDP) packets during the link discovery process [58]. To do this, they either generate fake LLDP packets and transmit them to the target SFFs for spurious link announcement, or they relay genuine LLDP packets received from one target switch to another. In both cases, the controller would assume that a new link has been established between those SFFs even though it is not true. This causes all traffic supposed to be routed via the original link to go through the malicious VNF, which would allow the attacker to eavesdrop on the traffic.
- *Elevation of privileges attack:* In [84,85], the authors suggested that it is possible for attackers to grant privileged access to the SDN controller by exploiting insecure APIs. In fact, the SDN controller provides an API for third-party applications to test, maintain, manipulate, debug, and monitor their applications; however, a careless design of this interface can lead to vulnerabilities in the controller. For example, attackers can abuse the API to increase their privileges, directly modify the configuration of network policies without authorization, set highest priorities for flow rules that are relevant to their malicious applications, decrease the priorities of all related flow rules being generated from the policies of security functions, provide access to their applications to facilitate communication with all network elements and other controllers, and capture raw data containing confidential information. Varadharajan et al. [86] exemplified the attack model by exploiting the vulnerable northbound API to install a malicious script such as cross-site scripting, SQL injection, command injection, and buffer overflow, which bypasses the authentication mechanism in the controller. This allows attackers to gain access to the controller and perform harmful operations. For example, the attackers can eavesdrop on the traffic, and delete or modify the forwarding rules in the flow entry tables. In addition, the attackers may exploit the vulnerabilities of the controller software to elevate their privileges. An example of this case is discussed in [87], in which attackers exploit software code vulnerability, *i.e.*, improper input validation (CVE-2019-1010245), to facilitate the execution of arbitrary commands on the controller.
- *Inconsistency of policy:* The controller determines the optimal topology-based routing paths after it receives the specified SFC policy from the upper level, and consequently, it translate the SFC policy into a set of low-level packet forwarding rules for deployment in the relevant networking elements. However, dynamic (fast changing) and decentralized environments with multicontroller and multitenancy supports remain a significant barrier that create a spacing issue for the accurate enforcement of policy, which results in configuration conflicts between flow rules from different points of presence. Pisharody et al. [88] pointed out that, in a decentralized environment with multiple controllers, the policy conflict issue is amplified and conflicts can arise because of incorrect synchronization between different controllers, although each of them may not have the same view of the environment. To further complicate matters, a dynamically changing network topology adds its own problems. For example, the authors of [89] pointed out that inconsistencies in controller configurations will result in potential inter-federated conflicts. If the state of network changes, then all controllers may not receive the network information simultaneously. This can cause some stateful applications (*e.g.*, firewall) to behave improperly because of the controller's nonvisibility into past events.
- *Lack of data-plane acknowledgment:* One of the main functions of the controller is to regulate the rules for the entire network based on the requests generated by network applications; based on these set of rules, it configures the SFFs in the data plane. In addition to the dynamic environment in which an end-to-end service function chain can be provisioned as desired, its service function path can be also changed dynamically. This causes the controller to update the flow rules frequently; however, the communications between the controller and data-plane elements such as SFFs are open loop in nature [90]. Thus, the controller may not notice or be aware that the network updates in SFFs are out-of-order, or it may even fail to take effect. For example, the controller may consider a rule to have taken effect even if the switch fails to install it [91]. Therefore, the result from inconsistent views between the controller and SFFs in terms of flow rule placement is attributed to the lack of data-plane acknowledgments.

### 4.4. Security issues in the forwarding level

At this level, SFFs such as routers and switches are distributed over the network topology, which represent a logical representation of the underlying network resources and is responsible for traffic steering and forwarding. Once routing decisions are made by the SDN controller at the control level, a set of corresponding forwarding rules are then transferred over a standard control channel, *i.e.*, OpenFlow [8] to the SFFs located at this level. The corresponding rules are installed accordingly in the flow tables of the SFFs. The SFFs use these forwarding rules to determine the forwarding path and decide which action needs to be performed to steer the traffic flows. However, traffic steering creates an open door for attackers to launch many forms of attacks aimed at disrupting the routing operations in the forwarding packet. We present a list of attacks below.

- *DDoS attack targets to SFFs:* Like in the previous discussion, DDoS attack can occur in SFFs. This type of security attack overwhelms the SFF's resources until the target can no longer offer services [81]. For example, attackers may launch a DDoS attack by flooding a large number of table-miss packets to the SFFs for saturating their ternary content addressable memory (TCAM). This forces the SFFs to buffer them while sending the packet-in messages to the controller and waiting for responses as packet-out (or flow-mod) messages. However, this scenario leads to the two types of saturation: (1) By exhausting the SFF's resources either with massive table-miss packets to be buffered [80] or a large number of flow rules to be inserted into their flow tables based on the routing instruction of the controller [64]. (2) By congesting the switch-controller communication channel [79] when the memory of the SFF is full with the buffered table-miss packets, it sends the entire packet (instead of a fraction of the packet headers under normal circumstances) without buffering it to the controller. This places stress on resource utilization overheads (CPU, memory, bandwidth, *etc.*) for routing computation and rule installation, which negatively affects the controller performance. The authors of [1,75] pointed out that, when several VNFs are deployed, they act as SFFs that run on one physical host. Therefore, it is possible for the attackers to launch DDoS attacks to exhaust all available resources shared between the SFFs. For example, attackers can send fake requests to a victim SFF to let it exhausts its computing resources; this leads to a scenario wherein other SFFs that run on the same host machine are starved for the resources.
- *Flow table overflow:* Once the SFF receives the forwarding rules generated by the controller, it uses TCAM to store its corresponding rules. However, the limited storage, high power consumption, and high cost of TCAM, leads to a number of forwarding rules (flow table entries) to be severely limited for storing, and thus, it has potential risk to be overflown [92,93]. Considering the fact that most vendors design the TCAM to support up to a few thousands of flow entries per SFF, the main reason is attributed to the inherent trade-off between table size, power consumption, and cost [94,95]. This memory limitation can be exploited by attackers to allow flow rule flooding from the malicious controller to the victim SFF to overflow its buffer space. This attack implicitly enforces the flow table of the SFF to fill up quickly with a large number of new flow rules; this leads to buffer overflow and causes zero space for storing incoming legitimate flow rules, which leads to them being dropped.
- *Flow rule modification:* The authors of [59,60,96] pointed out that flow rule modification can be targeted at the victim SFFs to deviate from the original service function path in which the packets are to forwarded. The source of this attack could be attributed to the result of flow table manipulation such as changing NAT, redirecting the paths, or rewriting the destination to forward the flow to an attacker's server. If this attack succeeds, the attackers can either manipulate the SFF's flow rules to redirect traffic based on their requirements, or masquerade the legitimacy of network elements by leveraging spoofing approaches. In [29,61], the authors classified a flow rule modification attack into three basic forms: (1) bypassing elements: at least one or more VNFs on the intended service chain are bypassed; (2) detouring paths: the packets are redirected back to the original path after being forwarded to anomalous VNFs; and (3) out-of-order traversal: to use brute force to compromise the ordering property of the service function chain as specified in the SFC policy. To achieve these goals, attackers can compromise the SFF by exploiting the vulnerabilities of its hosting OS, and by installing a backdoor program that allows them to arbitrarily perform malicious operations, *e.g.*, install, delete, or modify the flow tables. A more severe attack deals with taking control of SFFs (*i.e.*, hijacking) to compromise network behavior and infer confidential information within the system.
- *Traffic interception:* If control messages between the controller and SFFs are unencrypted, attackers can perform a man-in-the-middle attack to sniff and intercept the control messages. As identified in [89], attackers can place themselves in the middle of a communication channel to eavesdrop on the flow for determining which flows are in use and which traffic is permitted across the network. Then, they can guess what topology is constructed by passively sniffing control messages or intercepting the control messages, followed by changing some field values with malicious intent [97]. In the worst-case scenario, they can immediately seize full control of any downstream SFFs and execute fine-grained eavesdropping attacks that would be difficult to detect [98]. Thus, the risk posed by a successful man-in-the-middle attack is attributed to the constant connectivity and lack of authentication in the control messages.
- *Information disclosure:* Various overlay network technologies such as virtual extensible local area network (VXLAN), generic routing encapsulation, and NSH have been widely used to connect SFC data-plane elements to form an end-to-end application-specific service chain. For steering packet flow, it relies on SFC information (*i.e.*, SFC header, packet fields, or tags) and metadata sharing to identify the path of packets being forwarded along a service function path. The process of handling this information is often transparent to the SFC data-plane elements, and therefore, all risks associated with data exposure are presented. The authors of [34,36] pointed out that this transparency allows SFC information to be forged and the traffic flow to be manipulated. For example, attackers can obtain an opportunity to discover the content of packets during their transportation, or to intercept the SFC information to further obtain infrastructure, network topology, or the subscriber's information as part of a larger attack plot.
- *Packet injection:* Deng et al. [99] revealed that a new vulnerability can be discovered if the communication between the controller and SFFs is based on the OpenFlow protocol. In the existing design of communication processes, when the SFF receives a new packet with no matching flow rules in the flow table, this unmatched packet is encapsulated in a packet-in message and transmitted to the controller over the OpenFlow channel without any security regulation mechanisms. This scenario allows adversaries to exploit control messages and overwrite flow rules by launching a packet injection attack [58,100]. For example, they can enforce a malicious user to continuously inject manipulated packets with fake MAC or IP addresses, which consequently force the SFFs to send packet-in messages to the controller. The result from this attack deceives the controller and causes it to generate a fake topology with numerous non-existing devices, which results in misleading network forwarding decisions and flow rule installation in the SFFs.

- *Replay attack:* This is a network attack in which a valid data transmission is maliciously or fraudulently repeated. This attack attempts to subvert the security of legitimate communications and repeat them afterwards. The objective is to disrupt or create a negative impact for legitimate connections. For example, when a service chain is defined as the security requirement for data integrity, a passive attacker can record legitimate packet flows and replay them into the network to pass the validation. As stated in [59,101], the existing techniques for traffic steering are prone to replay attacks in which multiple SFFs can be compromised and colluded by the adversary to evade the verification.

## 5. Potential solutions for securing SFC

The key findings presented in the previous section with regard to threat analysis indicate that advancements of SFC come at the expense of an exponential growth in attack surfaces. Because SFC stacks involves multiple levels of abstraction from the service (*i.e.*, end-to-end service function chains) level, to the management and orchestration, control, and forwarding levels, this would yield a larger attack surface which would create new security threats that include cross-layer attacks. The established layer-specific threat taxonomy helps us gain a holistic view of the entire SFC attack surface. The results from a comprehensive threat analysis shed light on potential security risks in SFC domain from three different perspectives.

1. We observed that novel security threats and vulnerabilities are inevitably introduced (untrustworthiness in service chain composition, undependability issue in VNF chaining, inconsistency issue in the ordering properties of VNFs in a service chain, *etc.*), whereas others are inherited naturally from adopted technologies like NFV and SDN.
2. With multilevel cooperation and shared infrastructure resources, the security threats in the lower layer of the SFC architecture inevitably have an impact on the upper layers as well, and they incur cross-layer attacks.
3. Some threats such as DDoS attacks, data leakage, and policy conflicts are the most common threats that can appear on several levels of SFC stacks.

Therefore we intend to provide a set of recommendations for securing the SFC environment to defend against the aforementioned threats. We begin this section by systematically analyzing existing solutions associated with attack prevention and mitigation methodologies used for strengthening security in SFC towards securing end-to-end service chain delivery. A detailed survey of security solutions in the existing literature is discussed in the following sub-section with regard to the established threat taxonomy shown in Table 1. Our ultimate goal is to highlight all possible security measures and provide a holistic view of the SFC security framework, which would allow the involved parties including network operators to deploy adaptive, scalable, and cost-effective security hardening based on their specific requirements. To this end, a summary of security recommendations that explicitly strengthen the security of each SFC level is highlighted in each subsection accordingly.

### 5.1. Service level

We present security recommendations for defending and mitigating against the aforementioned threats in the service (*i.e.*, end-to-end service function chains) level to realize a higher level of security protection. A list of security recommendations is presented in Table 2.

- *Trust establishment:* One of the several methods used to secure service chain compositions requires all involved VNFs to be authenticated and the trust relationship between them to be established in advance. An alternative approach has been discussed in [29,61], wherein a verification model was proposed by leveraging an identity-based ordered multisignature scheme to ensure only legitimate and authorized VNFs possess the secret key, which is an essential input for producing a valid signature on a packet. Thus, any relationship in which the trust is broken either from unauthorized VNFs or forwarding anomalies can still be detected using this scheme.
- *Dependability guarantee:* End-to-end service chains can be deployed using a series of independent VNF instances stitched together in a logically ordered manner for creating a service function path. However, any path deviation to anomalous nodes, failure, or compromise of the involved VNF instances can lead to undependability in VNF chaining. Thus, it is important to ensure whether the dependability property of the service chain is well preserved, and if it is truly conformed to the specified SFC policy. To solve this problem, the authors of [30,31,102,103] proposed a verification mechanism for examining the dependability of packet traversal in the SFC domain. In [102], the authors proposed an SFC verification scheme (named *vSFC*) for evaluating the correctness of packet traversal. They utilized a hop-by-hop packet verification paradigm that leverages each VNF node to validate whether packets are correctly forwarded along the link from/to the VNF neighbors. The proposed method provided in [30] used SFC-related rules stored in the flow tables of virtual switches, and they modeled them as property-based graphs. Then, these graphs were used to examine whether the actual packet flow conforms to the required SFC specifications. In [31], the core concept is divided into three stages: (1) by taking a snapshot of the network state, *i.e.*, the topology, flow tables and the network function model; (2) creating a stateful forwarding graph to represent the entire networking state; and (3) developing graph-traversal algorithms on the created graph to check the dependability of packet traversal. Meanwhile, a verification tool proposed in [103] contains two sub-modules. The offline module takes a snapshot of the network configuration, and the online module leverages the analysis results obtained from the offline module for building a stateful forwarding graph and for further identifying the behaviors of packet forwarding during run time.
- *Traceability and monitoring:* Several types of VNFs can be created independently at the service endpoints and chained together in a specific order to form a service chain for achieving application-specific functionality defined in the service definition. However, (1) due to the dynamic nature of SFC in which VNF instances can be provisioned as desired while the service function path is changed dynamically, and (2) the configuration complexities in SFC deployment, it is challenging to verify if the service chains are rebuilt based on the agreed policy specifications. Therefore, any misconfiguration can cause the system to exhibit erroneous behavior with packets being forwarded to wrong VNFs or even being dropped along its path. To address this problem, a troubleshooting and monitoring tool was developed in[105]. Initially, the controller artificially injected a probe packet in the chain input to generate the trace. After the probe packet traversed the network elements in the target chain, it was mirrored by the trace tool to discover which forwarder handled the packet.
- *Ensuring the correct sequence of service functions:* There is a possibility that attackers may attempt to disrupt the ordering (position) properties of the involved VNFs in a service chain for security policy violation. An alternative solution to ensure the correct sequence of service functions in a service chain was proposed in [29,61]. The authors proposed a new security primitive that enforced all VNFs involved in a service chain to attach their signatures on the received packets, and then, the last hop of the chain (which has the role of verifier) validated the signature for checking the correctness. Any disruptions that impact the sequences of VNFs in a service chain can be detected by the verifier because of the verifiable properties of the proposed

**Table 2**
Security recommendations for service level.

| Types of threats and vulnerabilities | Relevant security recommendations | Existing works |
|---|---|---|
| Untrustworthy in service chain composition | Trust establishment | ChainSign [29,61] |
| Undependability in VNF chaining | Dependability guarantee | vSFC [102], ChainGuard [30], SFC-Checker [31], SLA-verifier [103] |
| Ordering-property preservation | Ensuring the correct sequence of service functions | ChainSign [29,61], POT [104] |
| Limited end-to-end service visibility | Traceability and monitoring | SFC Path Tracer [105] |
| Inconsistent enforcement of security policies | Verifying the correctness of policy enforcement | A verification model for policy enforcement [106], Automatic enforcement of security policies [107], VeriGraph [33], Graph-based policy change detection [108], SAM [107] |

scheme. Another approach called symmetric masking was proposed in [104]. The idea is to leverage an additional encryption mechanism for encoding the proof of transit's data with symmetric key pairs shared between the VNF nodes. For each service function path, the controller provisions all VNFs with two types of secret key pairs: (1) the upstream key is used to decipher the proof of transit's data received at an intermediate node; and (2) the downstream key is used in a reverse manner, which ciphers the proof of transit's data before transmitting the packet to the next downstream nodes. This reconstruction process allows the verifier to ascertain whether the packet followed the correct order while traversing the path. If the order is not respected, it generates an incorrect verification value and thus can be detected by the verifier.

- *Verifying the correctness of policy enforcement:* One of the most critical issues currently faced by SFC is the gap between the high-level SFC policy specifications and their enforcement in the data plane. It is uncertain how the specified SFC policy is securely enforced and correctly translated into network flow classification with accurate packet forwarding rules. The authors of [106] proposed a formal model to facilitate the verification of correct policy enforcement in SFCs. They aimed to clarify how the model can be used to describe real SFCs, and how the verification process is performed to cope with unwanted network behaviors (*e.g.*, security flaws) derived from the incorrect interactions of VNFs in the same SFC. Another method was proposed in [31], which aims to address the verification of policy consistency in the data plane by analyzing the forwarding behaviors of SFCs. A forwarding model was developed to verify the correct implementation of this policy. Using this developed model, network administrators can diagnose how the packets are forwarded, how the states of service function are changed, and how the state changes affect the forwarding path. In [33], the authors proposed a network modeling and a verification approach called *VeriGraph2.0*[4] to check reachability-based policies in a service chain. The designed method is structured into two sub-tasks: (1) chain extraction, which identifies all possible paths of service chains considering the policy specification; and (2) reachability verification, which examines whether the packet flows are traversed correctly while conforming to the specified policy. In [108], the authors leveraged multiattributed graphs to abstract the network topology and packet model, and then, they applied graph matching algorithm to detect the changes in policies. Further, the authors of [107] introduced a new component, called security awareness manager *(SAM)*, for taking charge of executing the policy refinement when the network is initialized or when a change is detected. It is developed based on the policy refinement model that translates high-level policies into low-level configurations, examines the service functions in which security policies is enforced, and reports users when nonenforceability is detected.

[4] It is extended version of [109] and integrated in both Extensible Service Chain Prototyping Environment (ESCAPE) and Open Platform for NFV (OPNFV).

### 5.2. Management and orchestration level

The management and orchestration level plays a vital role in managing and orchestrating virtual resources and network services, and its functionalities include (1) allocating network infrastructure resources; (2) validating and authorizing the requested resources; (3) managing software images; (4) instantiating, configuring, scaling, and terminating VNF instances; (5) upgrading VNF's software; (6) managing the integrity of VNF instances throughout their life cycles; (7) providing the topology management of network service instances (*i.e.*, VNFFG), and so on. Unfortunately, the broad set of features provided by this level results in a larger attack surface and new attack avenues. In this section, we intend to identify security recommendations that can help defend and mitigate against the potential security threats that can occur in this level. A focused set of security recommendations is highlighted in Table 3.

- *Security management and orchestration:* One of the best practices for enhancing security at the management and orchestration level is to extend its architectural framework with a security orchestrator that can manage security mechanisms and provide security as a service while incorporating security and trust requirements in an NFV/SDN enabled SFC domain. The preliminary concept of a security orchestrator was proposed in [110,111]. A few years later, the first prototype of the security orchestrator was developed [112] considering access control as the first use case to validate its feasibility and effectiveness. The NFVO could be enabled to enforce fine-grained access control to its assets because of the security orchestrator's capability, and this helped provide protection against attacks. Similar contributions have been proposed in [113,114], which aim to develop a security orchestrator that takes charge of providing security as a service. For example, selecting appropriate security enablers to be used for countermeasure when anomalous behaviors are detected.
- *Securing outsourced VNFs:* It has become increasingly common for enterprises to outsource VNF instances from a third-party cloud service provider. However, outsourcing VNFs poses considerable security risks. As the outsourced VNFs and related policies are fully managed and controlled by the untrusted third-party providers, this raises concerns about the disclosure of enterprise confidentiality, which can include organizational policies, sensitive details about an organization's network, IP addresses of hosts, or topology of the private network. Further, malicious providers can gain access to the organization's traffic or even modify and

**Table 3**
Security recommendations for management and orchestration level.

| Types of threats and vulnerabilities | Relevant security recommendations | Existing works |
|---|---|---|
| Attacks to centralized orchestration point | Security management and orchestration | Security orchestrator [110], SecMANO [111,112], Security orchestration plane [113,114] |
| Security risks from outsourced VNFs | Securing outsourced VNFs | SplitBox [115,116], Embark [117], SICS [118], SafeLib [46,119], Trusted Click [120], SafeBricks [121], S-NFV [122] |
| Vulnerabilities in VNF softwares | VNF image signing and software integrity | VNF image signing [70], VNF-Host sealing [123] |
| Security issues posed by shared VNF package | | |
| Security issues in VNF placement and chaining | Security-aware VNF placement | PESS [72,124], NSDP [71], Machine-learning-based VNF placement decisions [125,126] VNF placement using SPN models [127], VEREFOO [28], Security function graph configuration and deployment [128] |
| DDoS attack | Ensuring the availability of VNFs | A master-slave replication model [129,130], Replications and migrations based LP model [131], Weighting VNFs [132] |
| | Security monitoring | Security management and monitoring [133], Security monitoring module [114,134], SIM [135] |
| Virtualized infrastructure threats | Validation and authorization of network infrastructure resource requests | Security extension [56,112], SDAC [136] |
| | Remote attestation | Remote attestation based TPM [1,65,70,137,138] |
| | Regular VM updates and patches | Regular VM paching [1,70,78] |
| | Hypervisor and host protection | Hypervisor introspection [70,78,139] |

drop packets. To solve this problem, two categories of approaches have been proposed in recent years: cryptographic approaches (*e.g.*, [115–118]) and trusted-hardware approaches (*e.g.*, [46,119–122]). The former approach encrypts traffic that reaches the cloud and enables the cloud to process the encrypted traffic without decrypting it to protect privacy. The latter approach uses a hardware enclave such as Intel's software guard extensions [140] to provide an isolated execution environment that preserves the confidentiality and integrity of code and data within the enclave.

- *VNF image signing and software integrity sealing:* VM images are used to launch VM instances in an NFV/SDN-based SFC-enabled domain. Then, specific VNF software are deployed over a single VM or multiple VMs to deliver specific network appliances. In other words, these VNFs can be linked or chained together to form application-specific service function chains. From a security perspective, it is easy to tamper a VNF image by inserting malicious code while it is being uploaded to an image database or being transferred from an image database to a compute node [70,123]. An alternative countermeasure was proposed in [70], wherein the VNF images are cryptographically signed and verified during launch time. To achieve this, a signing authority needs to be set up and the hypervisor configuration needs to be modified to verify the signature of an image before execution. Like [123], the authors presented a verification mechanism by setting up a signing authority, which acts as a trusted third parity responsible for signing and verification. Each signed VNF image contains a separate signature file used in the verification process. The fresh hash digest of the VNF image is calculated to validate the signature. The authority uses this fresh hash digest along with the associated signature for integrity checks.
- *Security-aware VNF placement:* A major concern when solving the VNF placement problem is identifying how to efficiently utilize the resource and optimally provision application services using SFC deployment techniques and ensuring that the application requirements of the user are consistent from a security viewpoint. An alternative solution was proposed in [72], wherein a mathematical model was used to tackle the placement problem by considering both security policies and quality-of-service (QoS) requirements of the user applications. The authors presented an integer linear programming (ILP) formulation of the problem and validated their model by measuring the execution time based on four different constraints (*i.e.*, routing, resource, QoS, and security constraints), while comparing the model with other heuristics in terms of placement cost. The extended version of [72] was introduced in [124], and it was called the progressive embedding of security services (*PESS*). It aimed to (1) provide the ILP model for formulating the estimation of processing delay based on the computing resources of the physical nodes and for formalizing the impact on the end-to-end latency of operational services when allocating computing resources for new requests; and to (2) develop heuristic algorithm for obtaining near-optimal solutions of the embedding problem within an acceptable time frame. A similar idea involving the application of the ILP model was discussed in [71,125,126]. The author of [71] proposed network security defense pattern (*NSDP*) for expressing different deployment solutions for different tenants in the cloud with well-known security-related best practices and recommendations. An ILP formulation that captures various security constraints and then formalizes the optimal routing paths based on the proper VNF sequence for each traffic request has been presented; this formulation considers the capacity and security deployment constraints derived from security-related best practices. The authors of [125,126] applied machine learning to the ILP model for enhancing the prediction performance in the VNF placement decision. Further, the authors of [127] proposed an approach that uses a stochastic petri net to model the operational states of the physical nodes and the VNFs, and to calculate the overall SFC placement availability. In [28], authors proposed an automated

framework called verified refinement and optimized orchestration (*VEREFOO*) that aims to provide optimized allocation and orchestration of a service chain in a virtualized network. It was developed based on the maximum satisfiability modulo theories problem to automatically choose which service functions are required and where to allocate them. Furthermore, the authors of [128] proposed an approach for the configuration and deployment of virtual security functions. In this approach, the selection process of the security functions was performed after synthesizing the virtual network graph.

- *Ensuring the availability of VNFs:* Many users are affected when a VNF shared by multiple SFCs fails to deliver the service because all service chains that share this malfunctioning VNF are paused. Thus, it is important to guarantee high levels of VNF availability to improve the reliability of SFCs. A master—slave replication model was proposed in [129,130] to solve this problem; in this model, the master nodes serve as active VNF instances, while the slave nodes remain idle until the masters fail. The authors of [131] presented a linear programming model to optimally perform placement, replications, and migrations of VNFs, while maximizing the service reliability and minimizing the required network and server resources. In [132], the concepts of VNF's unavailability rate (*i.e.*, the probability that a user will be unable to use the services because of VNF failure) and the weighted index are applied. All VNFs involved in a service chain are weighted to determine duplications; the VNF with the largest weight is considered the most important VNF for duplication.
- *Security monitoring:* The status of infrastructure resources, VNF instances, network service chain, and other related events are hardly observable when considering the SFC solution deployed in the SDN/NFV environment. Therefore, effective security monitoring is recommended by the ETSI standard [133] to facilitate the visibility of management, control, network events, and activities of interest, while enabling advanced logging and accounting functions for additional security forensics and anomaly detection. Many proposals for security monitoring have been conducted to comply with the ETSI standard. For example, the ANASTACIA project [114,134] developed a security monitoring module as part of the security orchestration plane to collect security-focused real-time information related to the system behavior from physical/virtual appliances. The main objective was to detect anomalies and provide alerts for the reaction module when anomalous behavior was detected. Another security monitoring approach, called SFC integrity module (*SIM*) was proposed in [135] to serve as an anomaly detection solution for the orchestration plane. The design framework consisted of (1) a detector module for detecting anomalies with respect to SFC and VNF's operations; (2) a specifier module to identify the anomalous elements and select the most appropriate action to be taken; and (3) a library for storing the anomaly detection results and forwarding them to the NFV orchestration when queried.
- *Validation and authorization of network infrastructure resource requests:* One of the roles of the management and orchestration level is to provide network infrastructure resources in which VNF instances on the target service chain can run. However, it is very important for this level to have comprehensive security capabilities such as access control, *etc.*, to prevent adversaries from unauthorized or suspicious access to the resources and for securing and protecting network infrastructure resources against attacks. In [112], the authors developed a security extension for the NFVO to enable fine-grained access control to protect the underlying network infrastructure resources. An administrator can define specific security attributes for each virtual component (*e.g.*, VMs, VNFs), which are then leveraged to create access control policies that are eventually enforced in the cloud infrastructure. An extended version of [112] has been proposed in [56], which aims to extend the security capability to support multilayer and multidomain protection. The core module of the security extension employs an access control engine [136] for specifying high-level security policies; these policies are used to control access to the resources of specific cloud tenants. The central part of the access control engine serves as a tenant-specific access control paradigm that allows administrators to dynamically customize access control models (*e.g.*, DAC, MAC, RBAC) and policies for different tenant domains across multicloud data centers.
- *Remote Attestation:* The use of NFV and SDN introduces new network abstractions and high-level software-based primitives that are powerful and flexible for service provisioning and chaining. However, this creates a trust gap for administrators because they cannot easily assess the correct behavior of the software components run on top of these architectures. Remote attestation has been recommended as interesting solution for this problem [1, 65,70,137]. It is a technique that can be used to remotely verify the trust status of the NFV platform. The concept is based on boot integrity measurement using a specially designed hardware chip called the trusted platform module (TPM) that provides isolated storage, limited access capability, and a unique cryptographic identity. Coupled with the appropriate firmware, the TPM creates a platform wherein each software component is measured (*i.e.*, the digest is computed and reliably stored) before it is executed or accessed. In other words, remote attestation can be provided as a service, and it can be used by either the platform's owner or the consumer to verify whether the platform has booted in a trusted manner [138].
- *Regular VM updates and patches:* VNFs commonly refer to the software form of network appliances such as virtualized routers, firewalls, WAN optimization, and NAT services that run in virtual machines (VMs) on common virtualization infrastructure software like Linux KVM [141] or VMware vSphere hypervisor on the commercial off-the-shelf hardware [142]. Thus, security risks associated with VMs can affect the VNF instances deployed on top of them. Some examples of security risks include hyperjacking, VM escape, VM hopping, and VM DoS. Thus, to minimize the vulnerabilities and mitigate security risks associated with these VM attacks, the authors of [1,70,78] suggested maintaining regular VM updates and patches. One of the security best practices is to keep the hypervisor updated by regularly applying the released security patches; these patches should be applied not only to hypervisors and the hosted VMs, but also to all relevant software and workloads deployed on the top of virtual network infrastructure resources.
- *Hypervisor and host protection:* A central component of VMs is a software called a hypervisor, which is responsible for creating, running, and isolating the VMs in the virtualization environment. This component allows one host computer to support multiple guest VMs by virtually sharing its resources including memory and processing. Any attached VMs are also potentially compromised if the hypervisor is compromised. Therefore, it is necessary to maintain security at the hypervisor level. One of the security best practices is to regularly update the hypervisor by applying the latest security patches [70,78]. Further, it is important to disable all services that are not in use. For example, SSH and remote access service may not be needed all the time, and therefore, they should be enabled only when needed [139].

### 5.3. Control level

The control level uses the SDN controller to maintain a centralized view of the overall network, perform forwarding (routing) decisions, and manage flow control to the underlying systems, *i.e.*., SFFs, on how they should handle network traffic. The SDN controller is considered the brains of the network, and therefore, it is an attractive target for

**Table 4**
Security recommendations for control level.

| Types of threats and vulnerabilities | Relevant security recommendations | Existing works |
|---|---|---|
| DDoS attack targets to the controller | Ensuring the resilience and reliability of the SDN controller | Distributed controllers [143,144], Multiple controllers using Byzantine mechanism [145], Onix [146], HyperFlow [147], ONOS [148] |
| Control channel congestion | Congestion control | Floodlight [149], OF-Guard [150], FlowFence [151], FloodDefender [152], FloodGuard [153], Avant-Guard [80], FlowRanger [154], MLFG [79] |
| Topology poisoning attack | Establishing trust in network topology discovery | TopoGuard [58], SPHINX [100], DeepIDS [155], HMC [156], TopoGuard+ [157], TrustTopo [158], SVP [159] |
| | Data-to-control flow identification and verification | TopoGuard [58], PacketChecker [99] |
| Elevation of privileges attack | Access control enforcement | Controller DAC [160], AEGIS [161], PermOF [162], SDN-RBAC [163], SE-Floodlight access control model [164], AuthFlow [165], SE-Floodlight. [166], FRESCO [167], FortNOX [168] |
| | Secure northbound interfaces | Web-based northbound interface [169], Controller DAC [160] |
| Lack of data-plane acknowledgment | Consistent network updates | VeriDP [90], Monocle [91] |

attack. Thus, the following security recommendations must be carefully considered for the control level; the key recommendations are listed in Table 4

- *Ensuring the resilience and reliability of SDN controller:* The SDN controller acts as a strategic control point in a software-defined network, and therefore, it can become a single point of failure and an attractive attack target. A distributed control scheme was proposed in [143] to ensure the resilience and reliability of the controller; in this scheme, multiple controller instances were created and organized in a cluster. The key concept was to deploy each instance on dedicated virtual machines that favor the distribution of the network control workload across the cluster. In[145], the authors proposed the use of multiple controllers to confirm the update of flow tables in each switch, and they applied the Byzantine mechanism to guarantee that each switch correctly updated its flow tables, even if some compromised controllers issued false instructions. The objective of this approach was make the system robust against Byzantine attacks on the controllers and the communication links between the controllers and the switches. Similar ideas of leveraging multiple controllers for resilience and reliability have been proposed in [144,146–148].
- *Congestion control:* Flow requests flooded by malicious SFFs may cause excessive congestion in the control-data plan channel and overwhelm the controller's load owing to the limited processing capacity of the controller. To overcome this limitation, several approaches have been proposed.
  - *Solutions for data-control plane channel congestion: Floodlight* [149] uses a simple port throttling method to prevent switches from sending requests too fast. *OF-Guard* [150] presented a framework for preventing data-to-control plane saturation attacks by using (1) packet migration to monitor the rate of packet-in messages from the data plane, and (2) the data plane cache to store proactive flow rules and cache table-miss packets and to identify fake packets. *FlowFence* [151] was designed to handle network traffic congestion; it forces the switches to monitor their interfaces. The switch notifies the controller once a congestion state is detected; the controller then sends commands back to the switches to limit bandwidth usage on the congested interfaces. *FloodDefender* [152] employs three techniques for defending against a channel congestion attack: (1) table-miss engineering to prevent the communication bandwidth from being exhausted; (2) packet filtering to identify attack traffic and conserve computational resources of the control plane; and (3) flow rule management to eliminate useless flow entries in the flow table of the switch. *FloodGuard* [153] uses a proactive flow rule analyzer and packet migration for preventing a data-to-control plane saturation attack. The proactive flow rule analyzer module is used in the controller as an application that combines symbolic execution and dynamic application tracking for deriving proactive flow rules in the runtime. The packet migration module temporarily caches the flooding packets and submits them to the controller using the rate limit and round-robin scheduling.
  - *Solutions for controller resource saturation: Avant-Guard* [80] extends the existing OpenFlow data plane with two additional modules to overcome controller saturation. First, an *actuating trigger* module enables the data plane to asynchronously report the network status and payload information to the control plane. This module can be used to

activate a flow rule under predefined conditions for assisting the control plane in managing the network flow without delays. Second, a *connection migration* module delays the handshake by allowing the switch to act as a proxy server for the TCP handshakes. Only flow requests that complete the handshake are sent to the controller. *FlowRanger* [154] proposed a buffer prioritization solution for handling the controller saturation attack. The mechanism determines the source identity based on the ranking algorithm and serves the requests using multiple priority buffers. The packets in a higher priority buffer are processed with a higher priority than the packets in lower priority buffers. Multilayer fair queuing (*MLFG*) [79] enforces fair sharing of controller resources with multiple layers of queues; this is dynamically expanded and aggregated based on the controller load. When using MLFQ, the controller only needs to maintain a small number of queues. When an attacks occurs, the controller expands the corresponding queues into multiple lower-level queues to isolate the flooded requests.

- *Establishing trust in network topology discovery:* It is essential to protect the integrity of the network topologies. As indicated in the threat analysis (Section 4.3), malicious hosts and switches can poison the network view of the controller using host-location hijacking and link fabrication attacks. This leads to more severe security attacks, and many methods have been proposed to address this problem. For example, *TopoGuard* [58] checks the precondition and postcondition to verify the legitimacy of host migration and switch port property. *SPHINX* [100] uses an anomaly detection approach that relies on the inconsistencies in the network state at different sensors to detect the attacks. *DeepIDS* [155] and hierarchical multi-class (*HMC*) [156] apply a deep-learning approach for increasing the accuracy of anomaly detection. TopoGuard+ *TopoGuard+* [157] is an extension of TopoGuard [58], and it adds a control message monitor module and a link latency inspector module to detect characteristic anomalous interactions corresponding to in-band and out-of-band port amnesia attacks used as precursors to link fabrication and host location hijacking by attackers; these attacks are undetectable when using TopoGuard and SPHINX. *TrustTopo* [158] designed (1) a host location verification strategy based on the path tracking and asynchronous rollback technique for coping with host hijacking attacks, and (2) a link information verification strategy that uses a chaotic model and dynamic password generation technique for addressing link fabrication attacks. A stealthy probing-based verification approach (*SVP*) [159] sends probing packets to verify its legitimacy and identify how fake links are fabricated.
- *Data-to-control flow identification and verification:* Attackers can capture information such as host migration, switch status, and packet-level information from the data-to-control flow. Such information can be used for topology poisoning, faked device connection, malicious flow attack, and network application coexistence conflict [85]. *TopoGuard* [58] provides two approaches for ensuring the integrity of data-to-control flow packets: (1) authentication for LLDP packets; and (2) verification for the switch port property. It adds a controller-signed type length value into the LLDP packet and checks the signature when receiving the LLDP packets. In this case, the adversary can hardly manipulate the LLDP packets. *PacketChecker* [99] acts as a lightweight extension to the existing OpenFlow controller that identifies anomalies of the data-to-control flow packets, *i.e.*, packet-in messages in the network. to verify the legitimacy of these packet-in messages, PacketChecker filters out their header fields such as MAC addresses, switch data-path IDs, and in-port information, before these messages are processed by the controller core module. Once an anomaly is detected, it informs the switch to discard these packets using the flow-mod message.
- *Access control enforcement:* The SDN controller is a centralized decision point that facilitates a centralized view of the overall network and allows network administrators to dictate to underlying systems such as SFFs about how to handle the network traffic. Security attacks on the controller assets and applications have a broad network impact [170]. Therefore, access to the SDN controller and its applications should be tightly controlled and monitored. Currently, there are many access control and authentication mechanisms for protecting the SDN controllers from security threats; these include protecting against API abuse [160, 161], malicious applications [162–164], malicious hosts [165], and flow rule conflicts [166–168].
- *Secure northbound interfaces:* Applications communicate with the controller and can program underlying network devices, *e.g.*, SFFs, via SDN northbound interfaces with a REST API. However, without security control in place, these applications will have full access to the underlying network by default, which will enable the possibility of potentially malicious applications. To address this problem, an alternative solution has been proposed in [169], wherein a trust manager is developed and incorporated with the controller to examine the applications from a specific trust entity, *e.g.*, application name, vendor. Only authenticated and trusted applications are allowed to utilize the interface and communicate with the controller. In [160], a controller-independent dynamic access control system (*Controller DAC*) was developed to protect SDN controllers against API abuse. It comprises three major components: a security extension, controller-specific IDS, and high-level policy engine. The security extension authenticates and authorizes every request from the application, and it checks the request legality against the accounting records provided by controller-specific IDS.
- *Consistent network updates:* One of the main reasons that cause rule installation to be faulty at the data plane is attributed to the lack of acknowledgment from switches. This acknowledgment is important for the controller to enforce consistent network updates because it cannot update the upstream switch until the downstream switch completes updating its data plane [171]. *VeriDP* [90] is proposed to address this problem. VeriDP monitors the consistency between control plane policies and data plane forwarding behaviors, and it uses a path table as an abstract of control plane configurations, and it checks switches in the data plane to determine whether packet forwarding behaviors conform to the path table. *Monocle* [91] operated as a proxy between the controller and the switches by verifying whether the network view configured by the controller corresponds to actual switch behaviors. To ensure that a rule is functioning correctly, Monocle injects a monitoring packet (or a probe packet) into a switch, closely monitors the updated rules, and then informs the controller when the rule updates are completely installed in the switches.

### 5.4. Forwarding level

The threat analysis at the forwarding level suggests that the following security recommendations must be carefully considered. The available security best practices tailored to protect the data-plane traffic are also exemplified. Table 5 provides a summary of the threat analysis in the forwarding level with the corresponding security recommendations.

- *Threat monitoring:* A malicious attacker can launch a DDoS attack from compromised hosts and force them to generate a large number of packets with missing entries in the flow table of switches that are then sent to the central controller by OpenFlow [8]. This ultimately exhausts the controller and adversely affects the performance of the network; therefore, effective threat monitoring is required to maintain the visibility into network events and

**Table 5**
Security recommendations for the forwarding level.

| Types of threats and vulnerabilities | Relevant security recommendations | Existing works |
|---|---|---|
| DDoS attack targets to SFFs | Threat monitoring | Time and space-efficient solution [172], StateSec [173] |
| Flow table overflow | Protection against flow table overflow | Flow aggregation [82], FTGuard [174], STAR [175], FTS [176], A filter aggregation [92] |
| Flow rule modification | Flow rule verification | FlowChecker [177], VeriFlow [178], REV [59], 2MVeri [179], RueScope [180] |
| Traffic interception | Communication security | Secure NSH encapsulation [181], |
| Information disclosure | | Secure communication with Identity-Based Cryptography (IBC) [182], |
| Packet injection | | Network address translation method [183], OF-RHM [184], |
| Replay attack | | A distributed blockchain solution [185,186] |

enable the identification of infected hosts, which are the root cause of DDoS attack. In [172], the authors proposed a time and space efficient solution for identifying the compromised hosts that are induced to a DDoS attack. The defining characteristic of the compromised host is that most packets received by the serving switch have unique (nonrepetitive) headers. Therefore, the host can be identified as a compromised host if a majority of non-repetitive headers are high. *StateSec* [173] is an approach based on the use of in-switch processing capabilities that can detect and mitigate a DDoS attack; this approach monitors packets matching configurable traffic features (*e.g.*, source/destination IP and port), and feeds them to an entropy-based detection algorithm that exacts traffic information and differentiates between the different types of potential attacks.

- *Protection against flow table overflow:* According to the literature, several methods have been proposed to address flow table overflow attack. For example, the authors of [82] proposed two methods for preventing flow table overflow: (1) flow aggregation, in which one flow rule entry is used to match several flows, and (2) optimal timeout value, which is used as a threshold for removing the flow rule entries before the flow table of the switch overflows. *FTGuard* [174] distinguishes different flows with different priorities to achieve soft isolation for different users; this evaluation is formalized as a score. Based on the evaluation scores, flows from different users are assigned with the different priorities. The flows of benign users tend to have higher priorities, while those of suspected users are more likely to have lower priorities. Flow entries with the lowest priorities are evicted from the flow table to create a free space for new incoming flows when the flow table is full. *STAR* [175] utilizes limited flow-table resources for maximizing network performance. It detects the real-time flow-table utilization of each switch, evicts expired flow entries when there is a need to accommodate new flows, and selects routing paths for new flows based on the flow-table utilizations of switches across the network. The flow table sharing (*FTS*) mechanism is introduced in [176], wherein the table-miss packet are distributed to another lightly loaded switch in the network that has a spare flow table for processing. A filter aggregation algorithm is proposed in [92]; this algorithm merges content filters on individual switches to comply with its TCAM constraints, while ensuring a minimal increase in unnecessary network traffic. The algorithm uses the knowledge of advertisements, subscriptions, and a global view of the network state to perform bandwidth-efficient aggregation decisions on switches.
- *Flow rule verification:* A data plane must be secured from malicious applications that can install, change, or modify flow rules in the data path [89]. *FlowChecker* [177] uses binary decision diagrams to identify the intra-switch misconfiguration within a single flow table, and it can be used as an OpenFlow application or a master controller for validating, analyzing, and enforcing OpenFlow end-to-end configurations at runtime. *VeriFlow* [178] is a network debugging tool that identifies faulty rules issues and checks for network-wide invariant violations when each forwarding rule is inserted, modified, or deleted, and it prevents them from causing anomalous network behavior. Rule enforcement verification (*REV*) [59] and *2MVeri* [179] are tag-based solutions that exploit tags to measure the flow rule consistency. The REV allows a controller to verify whether rules have been enforced correctly and installed on the switches, which can help dismiss the rule modification attack. Further, it utilizes a compressive MAC that allows switches to compress tags before reporting them to the controller. Finally, the controller verifies the packet with its tag to determine whether the packet has traversed the intended path according to the rules. 2MVeri uses a Bloom filter and a two-dimensional vector as a tag, which is inserted in the packet header and updated in each switch. Finally, the tag is removed and reported to the controller for verification when the packet leaves the network. Moreover, RueScope [180] presents a probe-based solution for detecting and troubleshooting rule faults by injecting probe packets into the data plane and collecting probing results for forwarding inspection.
- *Communication security:* Control messages between the controller and authorized end points (*e.g.*, switches) should not be maliciously manipulated or intercepted. OpenFlow defines the use of the transport layer security (TLS) [187] and datagram transport layer security (DTLS) [188] to respectively secure the southbound communication and applications designed for UDP traffic. They are widely used to secure communication to prevent eavesdropping, tampering, or message forgery. Further, besides the use of TLS and DTLS, the following techniques can be used to provide information privacy. In [181,182], the authors presented an encryption technique to secure the communication. A network address translation method was proposed in [183]; this method does not notify the receiving host about the address of the transmitting host (sender) to prevent eavesdropping attacks. OpenFlow random host mutation (*OF-RHM* was proposed in [184]; it was developed based on a moving target defense approach for mutating the IP addresses of end-hosts to avoid scanning attacks. In [185,186], the authors exemplified the implementation of the blockchain to ensure security and privacy.

## 6. Challenges and future research directions

The key findings obtained from the previous studies reveal that SFC acts as a key enabler for network slicing in 5G systems. Further, it plays a critical role in delivering sophisticated services per slice and enables traffic traversal through a set of ordered service functions towards delivering an end-to-end network service. Although SFC has several advantages, security remains one of the vital concerns and a potential hurdle. Along with assessing different types of threats and vulnerabilities, we observed that several research issues and challenges still remain open although a significant amount of studies that have been conducted.

### 6.1. Security issues

Security issues are a primary focus and concern for all involved parties, including network infrastructure providers, network carriers, network operators, administrators, and enterprise customers. Deploying and operating dynamic SFC in a secure manner involves several abstraction levels in the SFC architectural model, and it covers the service (*i.e.*, end-to-end service function chain), management and orchestration, control, and forwarding levels. Thus, the key finding obtained from Section 4 reveals that different levels can potentially be exploited for security attacks. Some security attacks target network elements such as the orchestrator, controller, and VNFs to disrupt their availability. For example, failures of orchestrator/controller inhibit the ability of the network to respond to the new SFC requests or to react to the events originating from the underlying network nodes. Similarly, failures in the VNF nodes lead to a failure of the complete service chain because the traffic to be processed by the intended VNF is unreachable. This causes the traffic to be starved for service, and it does not reach the next VNFs in the service chain. Meanwhile, some attacks are launched via message fabrication to impede normal operations of the system, such as flow rule modification, topology poisoning, flow table overflow, and control channel congestion. Therefore, the results from these types of attacks can eventually lead to untrustworthiness or undependability in the delivery of the service chain. Moreover, more sophisticated attacks may involve the coordination of attackers at multiple levels, and this will yield a larger138 attack surface, which will open doors to new security threats such as cross-layer attacks. For example, if high-level SFC policy specifications do not either interpret cross layers or domains (in case of multicontroller support) accurately, or translate into low-level forwarding rules correctly, this can cause a new type of network configuration problem, which ultimately leads to policy conflicts. In the security context, such network policy conflicts can result in data leakage and isolation breaches in the SFC environment. Therefore, security threats introduced by the multilayer nature of the SFC requires a comprehensive understanding and the implementation of appropriate strategies to mitigate them. The ultimate goal is to assist network operators to deploy cost-effective security hardening measures based on their specific requirements.

#### 6.1.1. Security-aware SFC orchestration

In the initial stage of development, many security problems remain unsolved in SFC orchestration. These problems includes (1) the enrichment of an initial SFC request that considers security requirements, (2) establishment of trust in the network links and service function path, and (3) adaption of automated security management and zero-touch provisioning of security services with minimal intervention. For (1), the authors of [189] reported that, for several deployment use cases, end-users may request SFCs to satisfy their needs without considering security issues. Therefore, the lack of security awareness can lead to a limited protection of end-users from various types of threats. For (2), it is often necessary to deploy the VNFs on physical hosts and to establish a routing path between them when dealing with the SFC requests from end-users. Therefore, the method to find the optimal solution for the placement of SFCs on a shared substrate and to establish the corresponding traffic routes between the involved VNF is known as the SFC embedding problem. Although this problem has received considerable attention in the past years, negligible attention has been directed to the security aspects (*i.e.*, how to ensure that the SFC was created by considering the trustworthiness of a host system, links and network paths) when implementing an SFC embedding strategy [190,191]. The distributed heterogeneous nature of the NFV, wherein network operators leverage multiple infrastructure providers to launch VNFs and create on-demand SFCs, makes it difficult to validate whether deployed SFCs are created on a trustworthy infrastructure. For (3), dealing with automated security management and zero-touch provisioning of security services to secure an SFC domain is another key challenge. The implementation of SFC involves several steps and relies on many technologies, each of which has complex configurations and maintenance requirements. Thus, these operational steps can introduce error-prone processes and administrative burdens while creating a greater attack surface and a considerably extensive action space for attackers. Therefore, it is imperative to have an automated system that can monitor, detect, and adaptively mitigate the anomalies by minimizing the attack surface, performing optimal and autonomic decisions, and considering the multitude of factors for providing security hardening measures with cost-effective security mechanisms. From a conceptual point of view, even though an approach of autonomic cyberdefense (*i.e.*, automated security management and zero-touch provisioning of security services) is currently feasible and beneficial for defending against cyberattacks [192,193], none of them have achieved a fully automated cyberdefense across multidomain environments.

#### 6.1.2. Enforcement of security policies

As discussed in Section 4, the dynamic nature of SFC and provisioning across multiprovider cloud environments introduces an unprecedented challenge to administrators in enforcing security policies. Administrators have become aware about the importance of security policies and their enforcement in the underlying infrastructures of third-party service providers. Considering this fact, each SFC may contain different sets of service functions that run on third-party infrastructures and may have different paths that must be followed by the packets. Further, each path may deal with different service providers and have different terms of security regulation and policy enforcement that must be satisfied [37]. However, from a security point of view, these additional constraints of multicloud environments, multiple types of service functions, and multivendor integration make it extremely difficult to coordinate global security policies and build trust relationships between the service components across multiple SFC-enabled domains.

#### 6.1.3. Cross-domain scenarios

A promising approach to achieve cost-effective resource utilization is to deploy SFC across multiple domains. However, cross-domain operations have a trade-off between security objectives and performance efficiency. Despite its advantages, there is a security risk associated with leveraging cloud network infrastructures across multiple administrative domains. Administrators are forced to share their network information such as topology, resource availability, and internal policy configuration with external parties to improve the efficiency and optimal cost for SFC deployment. In contrast, sharing this network information with untrusted third parties may be harmful. The authors of [194,195] pointed out that the hierarchy of administrators can become a potential source of threats in terms of the delegation of control or privileges between the orchestrators of different administrative domains. For example, they may exploit this opportunity to monitor the network behaviors of others, learn their existing security policies, and maliciously modify the packets for unlawful gain or business competition.

#### 6.1.4. SFC deployment

From a deployment perspective, the authors of [34] pointed out that securing SFC requires operations at different components of the SFC-enabled domain. First, specific requirements (such as protection and confidentiality) need to be enforced at both the ingress and egress boundaries nodes of an SFC-enabled domain to avoid SFC information leakage and protect against various forms of attacks. For example, if untrusted parties inject packets that are treated as properly classified for service chaining, there is a large range of attacks that can be mounted against the system [11]. One of the most common types of attacks is packet spoofing, which is considered a source of creating DDoS and reflection attacks of various kinds. Therefore, the boundaries of an SFC must ensure that outside parties cannot inject spoofed SFC packets that will be accepted into the domain for further processing to achieve packet integrity and validity. Second, the classification process is critical for the security of SFC, and it requires the implementation of the authentication and checkups beforehand [34]. Third, a set of SFC data-plane elements including SFFs and VNFs should establish a relationship of trust to ensure that only legitimate and authorized nodes are allowed to process the tasks [61].

#### 6.1.5. SFC-based traffic steering scheme

Some SFC techniques can introduce security concerns and threaten infrastructure security during implementation [34]. To realize an SFC overlay, SFC leverages various overlay network technologies (*e.g.*, VXLAN, GRE, NSH) to provide connectivity between the SFC data-plane elements and to allow the establishment of a service function path. This overlay network layer is assumed to be transparent [196]; however, because of the lack of a standard definition or applicable characterization for implementing SFC, some traffic steering schemes allow opaque VNFs in the chain to modify the packet headers. Thus, a lack of unawareness of such modifications can lead to inconsistencies in the service function path or convey misleading policies [34]. Another concern is related to metadata sharing. The SFC forwarding operation is based on the service function path information and metadata shared between the SFC data-plane elements; therefore, risks associated with information alteration and traffic manipulation need to be addressed [36].

### 6.2. Quality of service (QoS)

A critical research area in SFC is QoS deployment and delivery. When considering a next-generation network such as a 5G system, dynamic connectivity, agile service chain provisioning, and the support of high QoS are required to deal with the ever-growing demands for bandwidth from specific tenants and industrial critical requirements. Therefore, it is important to analyze all factors that affect the QoS of the overall end-to-end service chain such as cost, energy consumption, network traffic volume, and latency. In the following, we investigate QoS constraints from different SFC perspectives, which include resource allocation, VNF placement, service chain composition, and service function path selection.

#### 6.2.1. Resource allocation

Although optimal resource allocation is a widely studied topic (*e.g.*, [20,37,38,197–200]), it is an open issue for SFC architecture. This is because most methods for resource allocation have a limited constraint, and they focuses on determining the optimal resource allocation in a single cloud environment; in contrast, SFC allows for the spread of resources across multiple clouds in geographically distributed areas. Another challenge is that, in real situations, SFC requests are not known in advance and can be changed dynamically at any time [199]. The resources are not allocated if the incoming requests ask for new services that have not been deployed. The time taken for (1) determining the optimal solutions to achieve the required network costs and service performance, and (2) allocating the resources according to the SFC requests can lead to unavoidable delays while impacting QoS. Therefore, the solutions should address this as an online problem. Current research studies on resource allocation in service chain problems are mostly static. In contrast, the allocation of resources at runtime remains a complex problem. The authors of [38] pointed out that the real-time resource allocation of SFC introduces significant new challenges because of its dynamic features. First, the resources of a specific VNF involved in a service chain can be scaled dynamically based on traffic loads. For example, a DPI may require fewer computing resources when the traffic decreases, and vice versa. Second, the QoS demands for VNF can be changed dynamically because of changes in user requirements. For example, when an SFC request requires low latency, reallocation of VNFs is required. Third, during VNF processing, the VNFs should be periodically monitored to avoid the reliability problem. If there is a possibility of VNF failures, then the reallocation of resources and reassignment of the VNF instance for the corresponding SFC requests is necessary. Therefore, dynamic and real-time optimization solutions for resource allocation according to the varying workloads and SFC requests need to be studied further to overcome these challenges.

#### 6.2.2. VNF placement

VNF placement aims to provide maximum throughput and higher average performance per VNF when the resources are limited via efficient placement, while achieving load balancing and reducing the overall cost. This has been investigated in various ways using optimization algorithms such as ILP, mixed integer linear programming, greedy algorithm, and dynamic programming algorithm. Although several studies have focused on VNF placement, the authors of [34,36,201] pointed out that many key challenges need to be adequately addressed when seeking the optimal location for VNF placement to be embedded in the cloud infrastructure. First, only a limited set of constraints are considered in the previous studies. For example, many studies (*e.g.*, [26,202–205]) considered VNF resource allocation with VNF placement and modeled them as an optimization problem (*e.g.*, ILP) for minimizing the total number of VNF instances, communication costs, and operational expenses under the constraints of network traffic and physical network topology. These works assumed that VNF capacities are known and static, whereas they ignore the performance characterization challenges [201]. Further, they did not consider the effect of dynamic network traffic on different types of VNFs. Therefore, the placement solutions obtained from these proposed optimization models may not work well in real-world deployments. Second, placement algorithms should consider realistic cases in which some VNF instances cannot be deploy and coexist in the same physical node because of conflict or restriction [34]. Although SFC allows for resource sharing among VNF instances in a service chain and across multiple service chains, the result of combining one or more VNFs in the same node may realize efficient VNF placement for a specific chain, whereas it may not be optimal for others. Therefore, the placement problem is even more challenging when these metrics are considered. Third, the service placement problem is critical for the QoS, although resource allocation is not less important; indeed, these two problems impact each other. The authors of [36] pointed out that the flexible placement of VNFs requires a dynamic resource allocation strategy. Although SFC allows for dynamic resource sharing and elastic scaling; this permit VNFs to dynamically acquire the required resources on-demand. Further, the availability of global resources may be limited in practice. Thus, the VNFs can have fewer resources than required, while some of them may run out of resources. Under these circumstances, VNF failures may increase and ultimately impact the overall service delivery and the performance of service chains.

#### 6.2.3. Service chain composition

Despite there being several models and algorithms for service chain composition in the literature, composing SFCs and mapping them to physical resources remains challenging for two main reasons: (1) Several criteria make the composition problem challenging and difficult to solve [34,36]. Several criteria need to be considered in practice when composing an ordered set of VNF instances to form a service chain and to optimally embed it into physical resources based on a predefined objective. These criteria include factors such as actual networking state, infrastructure capabilities, and subscriber preferences. The authors of [34] pointed out that other constraints may be violated to satisfy some constraints. For example, sharing VNFs across multiple SFCs may reduce VNF placement cost and improve resource utilization. However, this can result in longer paths, higher end-to-end latency delays, and more bandwidth consumption. (2) The determination of the service chain composition that induces short latency and low congestion remains a challenge. Existing works investigate the composition problem in a centralized manner, *i.e.*, coordinating the service chains to achieve optimal performance with full knowledge of the network state. Thus, full knowledge of the network state and system parameters is required to calculate an optimal composition solution. In other words, the network must be continuously monitored, and each user needs to disclose their private information that is relevant for calculating their cost functions. These solutions can be impractical because of privacy issues and non-cooperative interactions among users [206,207]. Further, they may fall short in the face of unexpected failures because of user and resource unavailabilities. Therefore, handing these failures can exacerbate the problem to become even more challenging, and this can significant affect system performance [208].

#### 6.2.4. Service function path selection

A service chain in an SFC-enabled domain comprises one or more paths depending on the load at each VNF instance deployed across the network and the total congestion along each path. For example, if the same function of VNF instances is deployed and distributed across multiple servers for load balancing, then there are multiple potential paths for the selection to build this service chain comprising the required VNF instances. However, the path selection problem for SFC primarily focuses on the QoS, resource, and link capacity constraints. Indeed, differentiating paths allows for satisfying different SFC constraints [34]. For example, some policy profiles can prioritize cost minimization along with an increase in latency and link bandwidth, and other cases can prioritize delivery time or the shortest path without considering resource usage (*e.g.*, VNF-node capacity). However, the authors of [34] pointed out that multicriteria path selection is a challenging problem, and current methods for path selection based on the shortest path selection fail to meet the end-to-end QoS requirements. An alternative approach was proposed in [22], wherein the path selection problem is divided into three sub-problems: (1) how to efficiently compute a shortest path between any pair of nodes that satisfies a given SFC constraint, (2) how to find the maximum feasible flow from a source to a destination that can achieve SFC constrains, and (3) how to place VNFs such that the cost (the number of nodes to be virtualized) is minimized, while achieving the maximum flow of the original network under the SFC constraint. Although the proposed approach achieves the original maximum flow in the network while satisfying a given SFC constraint, other key factors such as ensuring the occurrence of trusted path routing are not addressed in contemporary work [209]. It is a challenge to determine an optimal trust path that yields the most trustworthy evaluation and satisfies the service consumer's trust evaluation criteria. One other issue is that, once paths are calculated for the given service chains, it is difficult to update the path in real time and during the service delivery based on the current network state and environmental changes [36].

## 7. Conclusion

The growing demand for cloud services and the diverse requirements of various applications have led to the growth and popularity of SFC because of the advancements in the dynamic provisioning of service chains and the flexible composition of multiple VNFs. SFC combined with other promising technologies (such as NFV and SDN) allows it to be recognized as a key enabler for building a 5G ecosystem in which multiple service chains can be created from the different sets of VNF instances tailored to individual customer requirements. Although SFC offers significant advantages for service providers as well as network operators in the areas of (1) on-demand VNF composition and chaining, (2) dynamic service chain provisioning, and (3) cost-effective resource sharing, the combination of such key technologies unavoidably creates a wide variety of security risks in different levels of SFC stacks. These risks can ultimately result in a greater attack surface. Therefore, this survey investigated and explored the potential security risks in the SFC-enabled domain. To this end, a comprehensive survey about the SFC characteristics, architectural framework, design principles, and deployment use cases was presented to highlight the significant enhancements of adopting SFC to overcome the long-standing problems of the traditional networking model. A layer-specific threat taxonomy was conducted to identify all potential threats and vulnerabilities that can be incurred at each specific level of SFC stacks. Further, based on the established threat taxonomy, a set of security recommendations and security best practices were provided. The goal of this study was to propose security hardening measures with cost-effective security mechanisms to secure the SFC-enabled domain and to highlight threats that can be effectively mitigated or prevented using appropriate security countermeasures.

## Acknowledgement

This work is the output of ASEAN International Virtual Organization (ASEAN IVO) project[5] entitled "*Agricultural IoT based on Edge computing*", and financially supported by National Institute of Information and Communications Technologies (NICT)[6], Japan.

## References

[1] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, A. Meddahi, NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3330–3368.
[2] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F.D. Turck, R. Boutaba, Network function virtualization: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262.
[3] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, IEEE Commun. Surv. Tutor. 17 (1) (2015) 27–51.
[4] D. Kreutz, F.M.V. Ramos, P.E. Veríssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.
[5] ETSI, NFV: An introduction, benefits, enablers, challenges & call for action, 2012, URL https://portal.etsi.org/NFV/NFV_White_Paper.pdf.
[6] ONF, SDN architecture overview, 2013, URL https://opennetworking.org/wp-content/uploads/2013/02/SDN-architecture-overview-1.0.pdf.
[7] ETSI, NFV; ecosystem; report on SDN usage in NFV architectural framework, 2015, URL https://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf.
[8] ONF, OpenFlow switch specification, 2012, URL https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf.
[9] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patneyt, M. Shirazipour, R. Subrahmaniam, C. Truchan, M. Tatipamula, StEERING: A software-defined networking for inline service chaining, in: ICNP' 13, 2013, pp. 1–10.
[10] M. Mechtri, C. Ghribi, O. Soualah, D. Zeghlache, NFV orchestration framework addressing SFC challenges, IEEE Commun. Mag. 55 (6) (2017) 16–23.

[5] https://www.nict.go.jp/en/asean_ivo/index.html
[6] http://www.nict.go.jp

[11] J. Halpern, C. Pignataro, Service function chainning (SFC) architecture, 2015, URL https://tools.ietf.org/html/rfc7665.

[12] A.M. Medhat, T. Taleb, A. Elmangoush, G.A. Carella, S. Covaci, T. Magedanz, Service function chaining in next generation networks: State of the art and research challenges, Comm. Mag. 55 (2) (2017) 216–223.

[13] E. Wang, K. Leung, J. Felix, J. Lyer, P. Patel, Service function chaining use cases for network security, 2017, URL https://tools.ietf.org/html/draft-wang-sfc-ns-use-cases-03.

[14] W. Haeffner, J. Napper, M. Stiemerling, D. Lopez, Service function chaining use cases in mobile networks, 2019, URL https://tools.ietf.org/html/draft-ietf-sfc-use-case-mobility-09.

[15] C. Bernardos, A. Rahman, A. Mourad, Service function chaining use cases in fog RAN, 2021, URL https://tools.ietf.org/html/draft-bernardos-sfc-fog-ran-05.

[16] B. Liu, K. Katsalis, M. Zhang, Service function chaining applicability in industrial edge computing, 2018, URL https://tools.ietf.org/html/draft-liu-iiot-sfc-edge-computing-applicability-00.

[17] H. Huang, S. Guo, J. Wu, J. Li, Service chaining for hybrid network function, IEEE Trans. Cloud Comput. (2017) 1–13.

[18] A.M. Medhat, T.Q. Thanh, S. Covaci, Orchestrating service function chaining in cloud environments, in: IEEE ICCE' 16, 2016, pp. 532–538.

[19] A.M. Medhat, G.A. Carella, M. Pauls, T. Magedanz, Extensible framework for elastic orchestration of service function chains in 5G networks, in: NFV-SDN' 17, 2017, pp. 327–333.

[20] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, On orchestrating virtual network functions, in: CNSM '15, 2015, pp. 50–56.

[21] M. Ghaznavi, N. Shahriar, R. Ahmed, R. Boutaba, Service function chaining simplified, 2016, CoRR, abs/1601.00751.

[22] G. Sallam, G.R. Gupta, B. Li, B. Ji, Shortest path and maximum flow problems under service function chaining constraints, in: IEEE INFOCOM' 18, 2018, pp. 2132–2140.

[23] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, Optimal virtual network function placement in multi-cloud service function chaining architecture, J. Comput. Commun. 102 (2017) 1–16.

[24] A.F. Ocampo, J. Gil-Herrera, P.H. Isolani, M.C. Neves, J.F. Botero, S. Latré, L. Zambenedetti, M.P. Barcellos, L.P. Gaspary, Optimal service function chain composition in network functions virtualization, in: AIM' 17, 2017, pp. 62–76.

[25] M. Mechtri, C. Ghribi, D. Zeghlache, A scalable algorithm for the placement of service function chains, IEEE Trans. on Netw. and Serv. Manag. 13 (3) (2016).

[26] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, L.P. Gaspary, Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions, in: IM' 15, 2015, pp. 98–106.

[27] Z. Wang, J. Zhang, T. Huang, Y. Liu, Service function chain composition, placement, and assignment in data centers, IEEE Trans. Netw. Serv. Manag. 16 (4) (2019) 1638–1650.

[28] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, J. Yusupov, Towards a fully automated and optimized network security functions orchestration, in: ICCCS' 19, 2019, pp. 1–7.

[29] M. Pattaranantakul, Q. Song, Y. Tian, L. Wang, Z. Zhang, A. Meddahi, Footprints: Ensuring trusted service function chaining in the world of SDN and NFV, in: SecureComm' 19, Vol. 305, Springer, 2019, pp. 287–301.

[30] M. Flittner, J.M. Scheuermann, R. Bauer, ChainGuard: Controller-independent verification of service function chaining in cloud computing, in: NFV-SDN' 17, 2017, pp. 1–7.

[31] B. Tschaen, Y. Zhang, T. Benson, S. Banerjee, J. Lee, J. Kang, SFC-checker: Checking the correct forwarding behavior of service function chaining, in: NFV-SDN' 16, 2016, pp. 134–140.

[32] F. Valenza, S. Spinoso, R. Sisto, Formally specifying and checking policies and anomalies in service function chaining, J. Netw. Comput. Appl. 146 (2019) 1–14.

[33] D. Bringhenti, G. Marchetto, R. Sisto, S. Spinoso, F. Valenza, J. Yusupov, Improving the formal verification of reachability policies in virtualized networks, IEEE Trans. Netw. Serv. Manag. 18 (1) (2021) 713–728.

[34] H. Hantouti, N. Benamar, T. Taleb, Service function chaining in 5G beyond networks: Challenges and open research issues, IEEE Network 34 (4) (2020) 320–327.

[35] H.U. Adoga, D.P. Pezaros, Network function virtualization and service function chaining frameworks: A comprehensive review of requirements, objectives, implementations and open research challenges, Future Internet 14 (2) (2022).

[36] H. Hantouti, N. Benamar, T. Taleb, A. Laghrissi, Traffic steering for service function chaining, IEEE Commun. Surv. Tutor. 21 (1) (2019) 487–507.

[37] D. Bhamare, R. Jain, M. Samaka, A. Erbad, A survey on service function chaining, J. Netw. Comput. Appl. 75 (2016) 138–155.

[38] Y. Xie, Z. Liu, S. Wang, Y. Wang, Service function chaining resource allocation: A survey, 2016, CoRR, abs/1608.00095, arXiv:1608.00095.

[39] G. Mirjalily, Z. Luo, Optimal network function virtualization and service function chaining: A survey, Chin. J. Electron. 27 (4) (2018) 704–717.

[40] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu, Research directions in network service chaining, in: 2013 IEEE SDN for Future Networks and Services (SDN4FNS), 2013, pp. 1–7.

[41] G. Brown, Service chaining in carrier networks, 2015, URL https://www.qosmos.com/wp-content/uploads/Service-Chaining-in-Carrier-Networks_WP_Heavy-Reading_Qosmos_Feb2015.pdf.

[42] P. Quinn, T. Nadeau, Problem statement for service function chaining, 2015, URL https://tools.ietf.org/html/rfc7498#page-6.

[43] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, V. Sekar, Making middleboxes someone else's problem: Network processing as a cloud service, in: SIGCOMM '12, 2012, pp. 13–24.

[44] J. Sherry, S. Ratnasamy, A Survey of Enterprise Middlebox Deployments, Tech. rep., EECS Department, University of California, Berkeley, 2012.

[45] H. Jeon, B. Lee, Network service chaining challenges for VNF outsourcing in network function virtualization, in: ICTC' 15, 2015, pp. 819–821.

[46] E. Marku, G. Biczok, C. Boyd, Securing outsourced VNFs: Challenges, state of the art, and future directions, IEEE Commun. Mag. 58 (7) (2020) 72–77.

[47] ONF, L4-l7 service function chaining solution architecture, 2015, URL https://opennetworking.org/wp-content/uploads/2014/10/L4-L7_Service_Function_Chaining_Solution_Architecture.pdf.

[48] ETSI, NFV; management and orchestration, 2014, URL https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf.

[49] P. Quinn, U. Elzur, C. Pignataro, Network service hearder (NSH), 2018, URL https://www.rfc-editor.org/rfc/pdfrfc/rfc8300.txt.pdf.

[50] S. Kumar, M. Tufail, S. Majee, C. Captari, S. Homma, Service function chaining use cases in data centers, 2017, URL https://tools.ietf.org/html/draft-ietf-sfc-dc-use-cases-06.

[51] B. Sarikaya, D. von Hugo, M. Boucadair, Service function chaining: Subscriber and performance policy identification variable-length network service header (NSH) context headers, 2020, URL https://www.ietf.org/archive/id/draft-ietf-sfc-serviceid-header-08.txt.

[52] M. Gramaglia, D. Lopez, W. Haeffner, Service function chaining dataplane elements in mobile networks, 2017, URL https://datatracker.ietf.org/doc/html/draft-aranda-sf-dp-mobile-00.

[53] F5 networks, 2015, URL https://community.f5.com/t5/technical-articles/why-tcp-optimisation-has-become-more-important-than-content/ta-p/278329.

[54] W. Liu, H. Li, O. Huang, M. Boucadair, N. Leymann, Z. Cao, J. Hu, C. Pham, Service function chainning (SFC) use cases, 2013, URL https://datatracker.ietf.org/doc/html/draft-liu-service-chaining-use-cases.

[55] R. Bifulco, A. Matsiuk, A. Silvestro, Ready-to-deploy service function chaining for mobile networks, in: NetSoft' 16, 2016, pp. 175–183.

[56] M. Pattaranantakul, R. He, Z. Zhang, A. Meddahi, P. Wang, Leveraging network functions virtualization orchestrators to achieve software-defined access control in the clouds, IEEE Trans. Dependable Secur. Comput. 18 (1) (2021) 372–383.

[57] Y. Sim, H.Y. Lee, Poster: Denial-of-service attack using host location hijacking in software-defined network, in: Euro S&P' 16, 2016, pp. 1–2.

[58] S. Hong, L. Xu, H. Wang, G. Gu, Poisoning network visibility in software-defined networks: New attacks and countermeasures, in: NDSS' 15, 2015, pp. 1–15.

[59] P. Zhang, Towards rule enforcement verification for software defined networks, in: IEEE INFOCOM' 17, 2017, pp. 1–9.

[60] Q. Li, X. Zou, Q. Huang, J. Zheng, P.P.C. Lee, Dynamic packet forwarding verification in SDN, IEEE TDSC (2018) 1–16.

[61] M. Pattaranantakul, Q. Song, Y. Tian, L. Wang, Z. Zhang, A. Meddahi, C. Vorakulpipat, On achieving trustworthy service function chaining, IEEE Trans. Netw. Service Manage. 18 (3) (2021) 3140–3153.

[62] T.H.-J. Kim, C. Basescu, L. Jia, S.B. Lee, Y.-C. Hu, A. Perrig, Lightweight source authentication and path validation, in: SIGCOMM '14, 2014, pp. 271–282.

[63] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Commun. Surv. Tutor. 18 (1) (2016) 623–654.

[64] F. Reynaud, F.X. Aguessy, O. Bettan, M. Bouet, V. Conan, Attacks against network functions virtualization and software-defined networking: State-of-the-art, in: NetSoft' 16, 2016, pp. 471–476.

[65] M. Zoure, T. Ahmed, L. Réveillère, Network services anomalies in NFV: Survey, taxonomy, and verification methods, IEEE Trans. Netw. Serv. Manag. 19 (2) (2022) 1567–1584.

[66] M.D. Firoozjaei, J.P. Jeong, H. Ko, H. Kim, Security challenges with network functions virtualization, Future Gener. Comput. Syst. 67 (2017) 315–324.

[67] G. Biczok, B. Sonkoly, N. Bereczky, C. Boyd, Private VNFs for collaborative multi-operator service delivery: An architectural case, in: NOMS' 16, 2016, pp. 1249–1252.

[68] Common Vulnerabilities and Exposures, The CVE list, 1999, URL http://cve.mitre.org/cve/.

[69] D.B. Hoang, S. Farahmandian, Security of software-defined infrastructures with SDN, NFV, and cloud computing technologies, in: Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications, 2017, pp. 3–32, http://dx.doi.org/10.1007/978-3-319-64653-4_1.

[70] S. Lal, T. Taleb, A. Dutta, NFV: Security threats and best practices, IEEE Commun. Mag. 55 (8) (2017) 211–217.

[71] A.S. Sendi, Y. Jarraya, M. Pourzandi, M. Cheriet, Efficient provisioning of security service function chaining using network security defense patterns, IEEE Trans. Serv. Comput. 12 (4) (2019) 534–549.

[72] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, E. Salvadori, Application-centric provisioning of virtual security network functions, in: NFV-SDN' 17, 2017, pp. 276–279.

[73] Telco Systems, Protecting SDN and NFV networks from cyber security vulnerabilities, 2016, URL https://walkerfirst.com/uploads/files/literature/Telco_Protecting_SDN_and_NFV_Networks.pdf.

[74] M. Santos, A. Ranjbar, G. Biczók, B. Martini, F. Paolucci, Security requirements for multi-operator virtualized network and service orchestration for 5g, in: Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications, 2017, pp. 253–272, http://dx.doi.org/10.1007/978-3-319-64653-4_10.

[75] A.M. Alwakeel, A.K. Alnaim, E.B. Fernandez, A survey of network function virtualization security, in: SoutheastCon 2018, 2018, pp. 1–8.

[76] T. Takahashi, G. Blanc, Y. Kadobayashi, D. Fall, H. Hazeyama, S. Matsuo, Enabling secure multitenancy in cloud computing: Challenges and approaches, in: BCFIC' 12, 2012, pp. 72–79.

[77] Y. Huang, B. Chen, M. Shih, C. Lai, Security impacts of virtualization on a network testbed, in: 2012 IEEE Sixth International Conference on Software Security and Reliability, 2012, pp. 71–77.

[78] G. Pék, L. Buttyán, B. Bencsáth, A survey of security issues in hardware virtualization, ACM Comput. Surv. 45 (3) (2013) 40:1–40:34.

[79] P. Zhang, H. Wang, C. Hu, C. Lin, On denial of service attacks in software defined networks, IEEE Network 30 (6) (2016) 28–33.

[80] S. Shin, V. Yegneswaran, P. Porras, G. Gu, AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks, in: CCS '13, 2013, pp. 413–424.

[81] L.F. Eliyan, R. Di Pietro, Dos and ddos attacks in software defined networks: A survey of existing solutions and research challenges, Future Gener. Comput. Syst. 122 (2021) 149–171.

[82] R. Kandoi, M. Antikainen, Denial-of-service attacks in OpenFlow SDN networks, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 1322–1326.

[83] J. Cao, Q. Li, R. Xie, K. Sun, G. Gu, M. Xu, Y. Yang, The CrossPath attack: Disrupting the SDN control channel via shared links, in: USENIX Security Symposium 2019, 2019, pp. 19–36.

[84] Open Networking Foundation, Threat analysis for the SDN architecture, 2016, URL https://opennetworking.org/wp-content/uploads/2014/10/Threat_Analysis_for_the_SDN_Architecture.pdf.

[85] Y. Tseng, F. Naït-Abdesselam, A.A. Khokhar, A comprehensive 3-dimensional security analysis of a controller in software-defined networking, Secur. Priv. 1 (2) (2018).

[86] V. Varadharajan, K. Karmakar, U. Tupakula, M. Hitchens, A policy-based security architecture for software-defined networks, IEEE Trans. Inf. Forensics Secur. 14 (4) (2019) 897–912.

[87] CVE, Improper input validation, 2019, URL https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-1010245.

[88] S. Pisharody, J. Natarajan, A. Chowdhary, A. Alshalan, D. Huang, Brew: A security policy analysis framework for distributed SDN-based cloud environments, IEEE Trans. Dependable Secure Comput. 16 (6) (2019) 1011–1025.

[89] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, Security in software defined networks: A survey, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2317–2346.

[90] P. Zhang, H. Li, C. Hu, L. Hu, L. Xiong, R. Wang, Y. Zhang, Mind the gap: Monitoring the control-data plane consistency in software defined networks, 2016, pp. 19–33.

[91] P. Peresini, M. Kuzniar, D. Kostic, Dynamic, fine-grained data plane monitoring with monocle, IEEE/ACM Trans. Netw. 26 (1) (2018) 534–547.

[92] S. Bhowmik, M.A. Tariq, A. Balogh, K. Rothermel, Addressing TCAM limitations of software-defined networks for content-based routing, in: DEBS '17, 2017, pp. 100–111.

[93] J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, J. Zheng, Disrupting SDN via the data plane: A low-rate flow table overflow attack, in: SecureComm 2017, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 238, Springer, 2017, pp. 356–376.

[94] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, DevoFlow: Scaling flow management for high-performance networks, in: SIGCOMM '11, 2011, pp. 254–265.

[95] N. Katta, O. Alipourfard, J. Rexford, D. Walker, CacheFlow: Dependency-aware rule-caching for software-defined networks, in: SOSR '16, 2016, pp. 1–12.

[96] P.-W. Chi, C.-T. Kuo, J.-W. Guo, C.-L. Lei, How to detect a compromised SDN switch, in: NetSoft' 15, 2015, pp. 1–6.

[97] S. Lee, J. Kim, S. Woo, C. Yoon, S. Scott-Hayward, V. Yegneswaran, P. Porras, S. Shin, A comprehensive security assessment framework for software-defined networks, Comput. Secur. 91 (2020) 1–15.

[98] K. Benton, L.J. Camp, C. Small, Openflow vulnerability assessment, in: HotSDN '13, 2013, pp. 151–152.

[99] S. Deng, X. Gao, Z. Lu, X. Gao, Packet injection attack and its defense in software-defined networks, IEEE Trans. Inf. Forensics Secur. 13 (3) (2018) 695–705.

[100] J. Dhawan, R. Poddar, K. Mahajan, V. Mann, SPHINX: Detecting security attacks in software-defined networks, in: NDSS' 15, 2015, pp. 1–15.

[101] T. Sasaki, C. Pappas, T. Lee, T. Hoefler, A. Perrig, SDNsec: Forwarding accountability for the SDN data plane, in: ICCCN' 16, 2016, pp. 1–10.

[102] X. Zhang, Q. Li, Z. Zhang, J. Wu, J. Yang, vSFC: Generic and agile verification of service function chains in the cloud, IEEE/ACM Trans. Netw. 29 (1) (2021) 78–91.

[103] Y. Zhang, W. Wu, S. Banerjee, J. Kang, M.A. Sanchez, SLA-verifier: Stateful and quantitative verification for service chaining, in: IEEE INFOCOM' 17, 2017, pp. 1–9.

[104] F. Brockners, S. Bhandari, T. Mizrahi, S. Dara, S. Youell, Proof of transist, 2020, https://datatracker.ietf.org/doc/html/draft-ietf-sfc-proof-of-transit-08.

[105] R.A. Eichelberger, T. Ferreto, S. Tandel, P.A.P.R. Duarte, SFC path tracer: A troubleshooting tool for service function chaining, in: IM' 17, 2017, pp. 568–571.

[106] L. Durante, L. Seno, F. Valenza, A. Valenzano, A model for the analysis of security policies in service function chains, in: NetSoft' 17, 2017, pp. 1–6.

[107] C. Basile, F. Valenza, A. Lioy, D.R. Lopez, A. Pastor Perales, Adding support for automatic enforcement of security policies in NFV networks, IEEE/ACM Trans. Netw. 27 (2) (2019) 707–720.

[108] M. Hussain, N. Shah, A. Tahir, Graph-based policy change detection and implementation in SDN, Electronics 8 (10).

[109] S. Spinoso, M. Virgilio, W. John, A. Manzalini, G. Marchetto, R. Sisto, Formal verification of virtual network function graphs in an SP-DevOps context, in: ESOCC' 15, in: Service Oriented and Cloud Computing, vol. LNCS-9306, 2015, pp. 253–262.

[110] B. Jaeger, Security orchestrator: Introducing a security orchestrator in the context of the ETSI NFV reference architecture, in: 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, pp. 1255–1260.

[111] M. Pattaranantakul, R. He, A. Meddahi, Z. Zhang, Secmano: Towards network functions virtualization (NFV) based security management and orchestration, in: 2016 IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 598–605.

[112] M. Pattaranantakul, Y. Tseng, R. He, Z. Zhang, A. Meddahi, A first step towards security extension for NFV orchestrator, in: SDN-NFVSec@CODASPY' 17, 2017, pp. 25–30.

[113] A. Molina Zarca, J.B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, P. Gouvas, Security management architecture for NFV/SDN-Aware IoT systems, IEEE Internet Things J. 6 (5) (2019) 8005–8020.

[114] I. Farris, J.B. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, B. Sahlin, Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems, in: CSCN'17, 2017, pp. 169–174.

[115] H.J. Asghar, E. De Cristofaro, G. Jourjon, M.A. Kaafar, L. Mathy, L. Melis, C. Russell, M. Yu, Fast privacy-preserving network function outsourcing, Comput. Netw. 163 (2019) 1–15.

[116] H.J. Asghar, L. Melis, C. Soldani, E. De Cristofaro, M.A. Kaafar, L. Mathy, SplitBox: Toward efficient private network function virtualization, in: HotMiddlebox'16, 2016, pp. 7–13.

[117] C. Lan, J. Sherry, R.A. Popa, S. Ratnasamy, Z. Liu, Embark: Securely outsourcing middleboxes to the cloud, in: NSDI' 16), 2016, pp. 255–273.

[118] H. Wang, X. Li, Y. Zhao, Y. Yu, H. Yang, C. Qian, SICS: secure in-cloud service function chaining, 2016, pp. 1–13, CoRR, abs/1606.07079.

[119] E. Marku, G. Biczok, C. Boyd, Towards protected VNFs for multi-operator service delivery, in: NetSoft'19, 2019, pp. 19–23.

[120] M. Coughlin, E. Keller, E. Wustrow, Trusted click: Overcoming security issues of NFV in the cloud, in: SDN-NFVSec '17, 2017, pp. 31–36.

[121] R. Poddar, C. Lan, R.A. Popa, S. Ratnasamy, SafeBricks: Shielding network functions in the cloud, in: NSDI' 18, 2018, pp. 201–216.

[122] M.-W. Shih, M. Kumar, T. Kim, A. Gavrilovska, S-NFV: Securing NFV states by using SGX, in: SDN-NFV Security '16, 2016, pp. 45–48.

[123] S. Lal, S. Ravidas, I. Oliver, T. Taleb, Assuring virtual network function image integrity and host sealing in telco cloud, in: IEEE ICC' 17, 2017.

[124] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, M. Savi, E. Salvadori, Dynamic and application-aware provisioning of chained virtual security network functions, IEEE Trans. on Netw. and Serv. Manag. 17 (1) (2020) 294–307.

[125] S. Lange, H.-G. Kim, S.-Y. Jeong, H. Choi, J.-H. Yoo, J.W.-K. Hong, Machine learning-based prediction of VNF deployment decisions in dynamic networks, in: APNOMS' 19, 2019.

[126] T. Subramanya, D. Harutyunyan, R. Riggio, Machine learning-driven service function chain placement and scaling in MEC-enabled 5G networks, Comput. Netw. 166 (2020) 106980.

[127] G.L. Santos, P.T. Endo, T. Lynn, D. Sadok, J. Kelner, Automating the service function chain availability assessment, in: ISCC' 21, 2021, pp. 1–7.

[128] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, A novel approach for security function graph configuration and deployment, in: 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), 2021, pp. 457–463.

[129] M. Casazza, P. Fouilhoux, M. Bouet, S. Secci, Securing virtual network function placement with high availability guarantees, in: IFIP Networking and Workshops' 17, 2017, pp. 1–9.

[130] M. Casazza, M. Bouet, S. Secci, Availability-driven NFV orchestration, Comput. Netw. 155 (2019) 47–61.

[131] F. Carpio, A. Jukan, Improving reliability of service function chains with combined VNF migrations and replications, 2017, CoRR, abs/1711.08965.

[132] A. OI, M. Nakajima, Y. Soejima, M. Tahara, Reliable design method for service function chaining, in: APNOMS' 19, 2019, pp. 1–4.

[133] ETSI, NFV; security; security management and monitoring specification, 2017, URL https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/013/03.01.01_60/gs_NFV-SEC013v030101p.pdf.

[134] A.E. din Mady, R. Trapero, A. Skarmeta, S. Bianchi, Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems, in: CENICS' 17, 2017, pp. 61–64.

[135] L. Bondan, T. Wauters, B. Volckaert, F. De Turck, L.Z. Granville, Anomaly detection framework for SFC integrity in NFV environments, in: NetSoft' 17, 2017, pp. 1–5.

[136] R. He, M. Pattaranantakul, Z. Zhang, T. Duval, SDAC: a new software-defined access control paradigm for cloud-based systems, in: ICICS' 17, in: Lecture Notes in Computer Science, vol. 10631, 2017, pp. 570–581.

[137] A. Lioy, T. Su, A. Shaw, H. Attak, D. Lopez, A. Pastor, Guide to security in SDN and NFV; trust in SDN/NFV environments, in: Computer Communications and Networks, 2017, pp. 1–22.

[138] ETSI, NFV; NFV security; report on use cases and technical approaches for multi-layer host administration, 2015, URL https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/009/01.01.01_60/gs_nfv-sec009v010101p.pdf.

[139] R.D.V. Ronald L. Krutz, Cloud Security: A Comprehensive Guide To Secure Cloud Computing, Wiley Publishing, 1981, pp. 1–384.

[140] F. McKeen, I. Alexandrovich, A. Berenzon, C.V. Rozas, H. Shafi, V. Shanbhogue, U.R. Savagaonkar, Innovative instructions and software model for isolated execution, in: HASP' 13, 2013, pp. 1–8.

[141] The Linux Kernel Community, Kernel-based virtual machine (KVM), 2016, URL www.linux-kvm.org.

[142] VMware, Vmware vsphere documentation, 2020, URL https://docs.vmware.com/en/VMware-vSphere/index.html.

[143] L. Zuccaro, F. Cimorelli, F.D. Priscoli, C.G. Giorgi, S. Monaco, V. Suraci, Distributed control in virtualized networks, Procedia Comput. Sci. 56 (2015) 276–283.

[144] F. Bannour, S. Souihi, A. Mellouk, Distributed SDN control: Survey, taxonomy and challenges, IEEE Commun. Surv. Tutor. 20 (1) (2018) 333–354.

[145] H. Li, P. Li, S. Guo, A. Nayak, Byzantine-resilient secure software-defined networks with multiple controllers in cloud, IEEE Trans. Cloud Comput. 2 (4) (2014) 436–447.

[146] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, S. Shenker, Onix: A distributed control platform for large-scale production networks, in: OSDI' 10, 2010, pp. 1–14.

[147] A. Tootoonchian, Y. Ganjali, HyperFlow: A distributed control plane for OpenFlow, in: INM/WREN'10, 2010, pp. 1–6.

[148] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, G. Parulkar, ONOS: Towards an open, distributed SDN OS, in: HotSDN '14, 2014, pp. 1–6.

[149] Floodlight Project, Floodlight OpenFlow controller, 2012, URL https://floodlight.atlassian.net/wiki/spaces/HOME/overview?mode=global.

[150] L.X. Haopei Wang, G. Gu, OF-GUARD: A DoS attack prevention extension in software-defined networks, in: USENIX' 14, 2014, pp. 1–2.

[151] A.F.M. Piedrahita, S. Rueda, D.M.F. Mattos, O.C.M.B. Duarte, Flowfence: a denial of service defense system for software defined networking, in: GIIS' 15, 2015, pp. 1–6.

[152] G. Shang, P. Zhe, X. Bin, H. Aiqun, R. Kui, FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks, in: IEEE INFOCOM' 17, 2017, pp. 1–9.

[153] H. Wang, L. Xu, G. Gu, FloodGuard: A DoS attack prevention extension in software-defined networks, in: DSN' 15, 2015, pp. 239–250.

[154] L. Wei, C. Fung, FlowRanger: A request prioritizing algorithm for controller DoS attacks in software defined networks, in: ICC' 15, 2015, pp. 5254–5259.

[155] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, F. El Moussa, DeepIDS: Deep learning approach for intrusion detection in software defined networking, Electronics 9 (9) (2020) 1–18.

[156] H.-M. Chuang, F. Liu, C.-H. Tsai, Early detection of abnormal attacks in software-defined networking using machine learning approaches, Symmetry 14 (6) (2022) 1–27.

[157] R. Skowyra, L. Xu, G. Gu, V. Dedhia, T. Hobson, H. Okhravi, J. Landry, Effective topology tampering attacks and defenses in software-defined networks, in: DSN' 18, 2018, pp. 374–385.

[158] X. Huang, P. Shi, Y. Liu, F. Xu, Towards trusted and efficient SDN topology discovery: A lightweight topology verification scheme, Comput. Networks 170 (2020) 107–119.

[159] A. Alimohammadifar, S. Majumdar, T. Madi, Y. Jarraya, M. Pourzandi, L. Wang, M. Debbabi, Stealthy probing-based verification (SPV): an active approach to defending software defined networks against topology poisoning attacks, in: ESORICS' 18, vol. 11099, 2018, pp. 463–484.

[160] Y. Tseng, M. Pattaranantakul, R. He, Z. Zhang, F. Naït-Abdesselam, Controller DAC: Securing SDN controller with dynamic access control, in: ICC' 17, 2017, pp. 1–6.

[161] H. Padekar, Y. Park, H. Hu, S.-Y. Chang, Enabling dynamic access control for controller applications in software-defined networks, in: SACMAT '16, 2016, pp. 51–61.

[162] X. Wen, Y. Chen, C. Hu, C. Shi, Y. Wang, Towards a secure controller platform for openflow applications, in: HotSDN '13, 2013, pp. 171–172.

[163] A. Al-Alaj, R. Krishnan, R. Sandhu, SDN-RBAC: An access control model for SDN controller applications, in: ICCCS' 19, 2019, pp. 1–8.

[164] A. Al-Alaj, R. Sandhu, R. Krishnan, A formal access control model for SE-floodlight controller, in: SDN-NFVSec '19, 2019, pp. 1–6.

[165] D.M.F. Mattos, O.C.M.B. Duarte, AuthFlow: authentication and access control mechanism for software defined networking, Ann. Telecommun. 71 (2016) 607–615.

[166] P. Porras, S. Cheung, M. Fong, K. Skinner, V. Yegneswaran, Securing the software-defined network control layer, in: NDSS' 15, 2017, pp. 1–15.

[167] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, M. Tyson, FRESCO: Modular composable security services for software-defined networks, in: NDSS' 13, 2013, pp. 1–15.

[168] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, G. Gu, A security enforcement kernel for OpenFlow networks, in: HotSDN '12, 2012, pp. 121–126.

[169] C. Banse, S. Rangarajan, A secure northbound interface for SDN applications, in: 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, pp. 834–839.

[170] Cybersecurity Working Group, White paper: Considerations for securing SDN/nfv, 2016, https://transition.fcc.gov/bureaus/oet/tac/tacdocs/reports/2016/Securing%20-SDN-NFV%20-SWG-WP-Final.pdf.

[171] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, D. Walker, Abstractions for network update, in: SIGCOMM '12, 2012, pp. 323–334.

[172] S. Ali, M.K. Alvi, S. Faizullah, M.A. Khan, A. Alshanqiti, I. Khan, Detecting ddos attack on SDN due to vulnerabilities in OpenFlow, in: AECT' 20, 2020, pp. 1–6.

[173] J. Boite, P.-A. Nardin, F. Rebecchi, M. Bouet, V. Conan, Statesec: Stateful monitoring for ddos protection in software defined networks, in: NetSoft' 17, 2017, pp. 1–9.

[174] M. Zhang, J. Bi, J. Bai, Z. Dong, Y. Li, Z. Li, FTGuard: A priority-aware strategy against the flow table overflow attack in SDN, in: SIGCOMM Posters and Demos '17, 2017, pp. 141–143.

[175] Z. Guo, R. Liu, Y. Xu, A. Gushchin, A. Walid, H.J. Chao, STAR: Preventing flow-table overflow in software-defined networks, Comput. Netw. 125 (2017) 15–25.

[176] S. Qiao, C. Hu, X. Guan, J. Zou, Taming the flow table overflow in OpenFlow switch, in: SIGCOMM '16, 2016, pp. 591–592.

[177] E. Al-Shaer, S. Al-Haj, FlowChecker: Configuration analysis and verification of federated openflow infrastructures, in: SafeConfig '10, 2010, pp. 37–44.

[178] A. Khurshid, W. Zhou, M. Caesar, P.B. Godfrey, VeriFlow: Verifying network-wide invariants in real time, in: HotSDN '12, 2012, pp. 49–54.

[179] K. Lei, K. Li, J. Huang, W. Li, J. Xing, Y. Wang, Measuring the control-data plane consistency in software defined networking, in: ICC' 18, 2018, pp. 1–7.

[180] K. Bu, X. Wen, B. Yang, Y. Chen, L.E. Li, X. Chen, Is every flow on the right track?: Inspect SDN forwarding with RuleScope, in: IEEE INFOCOM' 16, 2016, pp. 1–9.

[181] V.-C. Nguyen, A.-V. Vu, K. Sun, Y. Kim, An experimental study of security for service function chaining, in: ICUFN' 17, 2017, pp. 797–799.

[182] J. Lam, S.-G. Lee, H.-J. Lee, Y.E. Oktian, Securing SDN southbound and data plane communication with IBC, 2016, pp. 1–13,

[183] M. Furukawa, K. Kuroda, T. Ogawa, N. Miyaho, Highly secure communication service architecture using SDN switch, in: APSITT' 15, 2015, pp. 1–3.

[184] J.H. Jafarian, E. Al-Shaer, Q. Duan, Openflow random host mutation: Transparent moving target defense using software defined networking, in: HotSDN '12, 2012, pp. 127–132.

[185] A. Hakiri, B. Dezfouli, Towards a blockchain-SDN architecture for secure and trustworthy 5G massive IoT networks, in: SDN-NFV Sec'21, 2021, pp. 11–18.

[186] B. Lokesh, N. Rajagopalan, A blockchain-based security model for SDNs, in: CONECCT' 20, 2020, pp. 1–6.

[187] T. Dierks, E. Rescorla, The transport layer security (TLS) protocol version 1.3, 2018, URL https://datatracker.ietf.org/doc/html/rfc8446.

[188] E. Rescorla, N. Modadugu, Datagram transport layer security version 1.3, 2017, URL https://www.rfc-editor.org/rfc/rfc9147.txt.

[189] H. Jmila, G. Blanc, Designing security-aware service requests for NFV-enabled networks, in: ICCCN' 19, 2019, pp. 1–9.

[190] N. Torkzaban, C. Papagianni, J.S. Baras, Trust-aware service chain embedding, in: SDS' 19, 2019, pp. 242–247.

[191] N. Torkzaban, J.S. Baras, Trust-aware service function chain embedding: A path-based approach, in: 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2020, pp. 31–36.

[192] R. Sahay, Policy-driven autonomic cyberdefense using software-defined networking. (Cyberdefense autonome pilotée par règles à l'aide d'un réseau défini par logiciel) (Ph.D. thesis), Telecom & Management SudParis, Évry, Essonne, France, 2017.

[193] M. Pattaranantakul, Moving towards software-defined security in the era of NFV and SDN. (Vers une programmabilité de la sécurité dans les environnements réseaux logiciels et virtualisés (NFV et SDN)) (Ph.D. thesis), University of Paris-Saclay, France, 2019.

[194] M.A.S. Santos, A. Ranjbar, G. Biczók, B. Martini, F. Paolucci, Security requirements for multi-operator virtualized network and service orchestration for 5G, in: Guide to Security in SDN and NFV - Challenges, Opportunities, and Applications, in: Computer Communications and Networks, 2017, pp. 253–272.

[195] K.D. Joshi, K. Kataoka, PSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/sdn, Comput. Netw. 178 (2020) 107–295.
[196] S. Aldrin, C. Pignataro, N. Kumar, R. Krishnan, A. Ghanwani, Service function chainning (SFC) operation, administration and maintenance (OAM) framework, 2020, URL https://datatracker.ietf.org/doc/html/draft-ietf-sfc-oam-framework-14.
[197] L. Wang, Z. Lu, X. Wen, R. Knopp, R. Gupta, Joint optimization of service function chaining and resource allocation in network function virtualization, IEEE Access 4 (2016) 8084–8094.
[198] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, X. Hesselbach, Virtual network embedding: A survey, IEEE Commun. Surv. Tutor. 15 (4) (2013) 1888–1906.
[199] J. Gil Herrera, J.F. Botero, Resource allocation in NFV: A comprehensive survey, IEEE Trans. Netw. Serv. Manag. 13 (3) (2016) 518–532.
[200] A. Belbekkouche, M.M. Hasan, A. Karmouch, Resource discovery and allocation in network virtualization, IEEE Commun. Surv. Tutor. 14 (4) (2012) 1114–1128.
[201] L. Cao, P. Sharma, S. Fahmy, V. Saxena, ENVI: Elastic resource flexing for network function virtualization, in: HotCloud'17, USENIX Association, USA, 2017, pp. 1–8.
[202] H. Moens, F.D. Turck, VNF-P: A model for efficient placement of virtualized network functions, in: CNSM' 14, 2014, pp. 418–423.
[203] R. Cohen, L. Lewin-Eytan, J.S. Naor, D. Raz, Near optimal placement of virtual network functions, in: IEEE INFOCOM' 15, 2015, pp. 1346–1354.
[204] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, R. Boutaba, Elastic virtual network function placement, in: CloudNet' 15, 2015, pp. 255–260.
[205] S. Clayman, E. Maini, A. Galis, A. Manzalini, N. Mazzocca, The dynamic placement of virtual network functions, in: NOMS' 14, 2014, pp. 1–9.
[206] S. Bian, X. Huang, Z. Shao, X. Gao, Y. Yang, Service chain composition with failures in NFV systems: A game-theoretic perspective, in: IEEE ICC' 19, 2019, pp. 1–6.
[207] S. Bian, X. Huang, Z. Shao, X. Gao, a. Yang Yang, Service chain composition with resource failures in NFV systems: A game-theoretic perspective, IEEE Trans. Netw. Serv. Manag. 18 (1) (2021) 224–239.
[208] S. D'Oro, L. Galluccio, S. Palazzo, G. Schembra, Exploiting congestion games to achieve distributed service chaining in NFV networks, IEEE J. Sel. Areas Commun. 35 (2) (2017) 407–420.
[209] G. Liu, Y. Wang, M.A. Orgun, E.-P. Lim, Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks, IEEE Trans. Serv. Comput. 6 (2) (2013) 152–167.

**M. Pattaranantakul** received her B.S. degree in computer science from Prince of Songkla University, Thailand, the M.S. degree in Computer Applications in Bangalore University, India, also the dual M.S. degree in Communication and Computer Security jointly developed by Télécom ParisTech and EURECOM, France, and Ph.D degree in network, information and communications from Télécom SudParis, University of Paris-Saclay, France, in 2019. She was a postdoctoral researcher at IMT Lille Douai, Institut Mines-Télécom, France. She is now a researcher at National Electronics and Computer Technology Center (NECTEC), Thailand. Her current research focuses on security in SDN/NFV environment.

**C. Vorakulpipat** received his Ph.D. in Information Systems from the University of Salford, UK. Currently, He is the head of Information Security Research Team at NECTEC, Thailand. He has been involved in several projects associated with information security (including ThaiCERT), national e-science, context-aware computing, e-health, and mobile application development. He has over seventy refereed publications in these areas. He also serves as a subcommittee member on issues regarding national information security of Thailand. He holds professional certificates including CISSP, CISA, PMP, and IRCA (ISMS Lead Auditor). His current research interests include access control, software security, security policy and standard.

**T. Takahashi** received his Ph.D. degree in telecommunications from Waseda University in 2005. He worked for the Tampere University of Technology from 2002 to 2004 as a researcher, Waseda University as a JSPS research fellow from 2004 to 2006, and Roland Berger Ltd. from 2006 to 2009 as a business consultant. He has been with the National Institute of Information and Communications Technology since 2009 and is currently an associate director. He was a visiting research scholar at the University of California, Santa Barbara, from 2019 to 2020. His research interests include cybersecurity and machine learning.

**ภาคผนวก ช.**

**รายงานผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสารสำนักงานกสทช.**

# โครงการ การพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต

## (ระยะที่ 1)

ดร. มลธิดา ภัทรนันทกุล ดร. ชาลี วรกุลพิพัฒน์ ดร. กลิกา สุขสมบูรณ์ ดร. เฉลิมพล ชาญศรีภิญโญ
ติณณ์ธิษณ์ ใจกล้า อภิวัฒณ์ จันทวิบูลย์ ปารย์ ศิริมนพร และธฤต ทองเปลว
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)
สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.)
อุทยานวิทยาศาสตร์ประเทศไทย ถนนพหลโยธิน ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี 10120

**บทคัดย่อ**

ในยุคที่เทคโนโลยีดิจิทัลที่เติบโตอย่างรวดเร็ว เทคโนโลยีการจำลองระบบเสมือนจริง ดังเช่น เทคโนโลยีการจำลองฟังก์ชันเครือข่ายเสมือน (Network Functions Virtualization: NFV) การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) และการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) กลายเป็นองค์ประกอบสำคัญในการพลิกโฉมโครงสร้างพื้นฐานเครือข่ายแบบดั้งเดิมสู่การให้บริการที่ขับเคลื่อนด้วยซอฟต์แวร์รองรับการใช้งานที่หลากหลาย ไม่ว่าจะเป็นเมืองอัจฉริยะ ระบบจัดการพลังงาน หรือการตรวจสอบและเฝ้าระวังภัยคุกคามทางไซเบอร์ ฯลฯ โดยเฉพาะในบริบทของ 5G ที่ต้องการความยืดหยุ่น ความรวดเร็ว และประสิทธิภาพในการจัดการเครือข่าย อย่างไรก็ตาม ประเทศไทยยังอยู่ในช่วงเริ่มต้นของการการพัฒนาเทคโนโลยีและนวัตกรรมดังกล่าวฯ โดยยังประสบกับข้อจำกัดและความท้าทายด้านองค์ความรู้และความพร้อมของบุคลากร โครงการพัฒนาแพลตฟอร์มบริหารจัดการฟังก์ชันเครือข่ายเสมือนจริงเพื่อสนับสนุนเครือข่าย 5G ในอนาคต (ระยะที่ 1) นี้จึงมีเป้าหมาย 2 ประการหลัก ดังนี้ (1) เพื่อวางรากฐานด้านระบบนิเวศเครือข่ายเสมือน (management and orchestration framework) ให้มีขีดความสามารถในการจัดสรรและบริหารทรัพยากรได้อย่างยืดหยุ่นและอัตโนมัติ; และ (2) เพื่อจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสเสมือน (VNFs as a Service: VNFaaS) โดยเฉพาะอย่างยิ่งเน็ตเวิร์คเซอร์วิสที่สนับสนุนงานด้านความมั่นคงปลอดภัยไซเบอร์ ดังเช่น Deep Packet Inspection (e.g., Suricata), Intrusion Detection System (e.g., Snort) และ Traffic analyzer (e.g., Zeek) เพื่อให้องค์กรสามารถใช้ประโยชน์สำหรับตรวจสอบ เฝ้าระวัง และป้องกันภัยคุกคามทางไซเบอร์ได้อย่างมีประสิทธิภาพ โครงการนี้จึงนับเป็นโครงการนำร่องที่สำคัญ ที่ไม่เพียงแก้ไขข้อจำกัดการให้บริการของระบบเครือข่ายแบบดั้งเดิม แต่ยังแสดงให้เห็นถึงการส่งเสริมให้เกิดการเตรียมความพร้อมต่อการจัดเตรียมระบบนิเวศ 5G ในอนาคต อันนำไปสู่การยกระดับขีดความสามารถด้านการแข่งขัน การพึ่งพาตนเอง และการปรับใช้ประโยชน์จากเทคโนโลยีดิจิทัลได้อย่างยั่งยืน

**คำสำคัญ:** เทคโนโลยีการจำลองฟังก์ชันเครือข่ายเสมือน การจัดการเครือข่ายด้วยซอฟต์แวร์ การประมวลผลข้อมูลบนคลาวด์ การจำลองเน็ตเวิร์คเซอร์วิสเสมือน การจัดการและการประสานงานอัตโนมัติ ความปลอดภัยไซเบอร์

**Abstract**

In today's rapidly evolving digital landscape, virtualization technologies – such as Network Functions Virtualization (NFV), Software-Defined Networking (SDN), and Cloud Computing – are transforming traditional network infrastructure into flexible, scalable, and software-driven infrastructure and services. These technologies support diverse applications including smart cities, energy management systems, and cyber threat monitoring, especially in the context of 5G networks, which demand agility, flexibility, speed, and efficiency. In Thailand, however, development in this domain remains in its early stages, with challenges stemming from knowledge gaps and the limited skilled personnel. Therefore, this project aims to address these gap through two main objectives: (1) establishing a virtual network ecosystem for flexible and automated resource management and orchestration; and (2) offering Virtual Network Functions as a Service (VNFaaS), with particular emphasis on cybersecurity-focused services such as Deep Packet Inspection (e.g., Suricata), Intrusion Detection Systems (e.g., Snort), and Traffic Analyzers (e.g., Zeek) to enhance an organization's capability to effectively monitor, detect, and mitigate against the cyber threats. Thus, this initiative serves as an important pilot project that not only overcomes limitations of legacy networking model, but also positions Thailand's to build a robust 5G ecosystem, boosting national competitiveness, self-reliance, and sustainable digiral adoption.

**Keywords:** Network Functions Virtualization (NFV), Software-Defined Networking (SDN), Cloud Computing, Virtual Network Functions (VNFs), Management and Orchestration, and cybersecurity services

## 1. บทนำ

ในยุคที่สถานการณ์โลกเปลี่ยนแปลงอย่างรวดเร็วและซับซ้อนกว่าที่เคย การเข้าสู่ยุคอุตสาหกรรม 4.0 ได้ส่งผลกระทบในทุกมิติ ทั้งในด้านโครงสร้างพื้นฐาน ICT การดำเนินธุรกิจ และรูปแบบการให้บริการ ทำให้ผู้ใช้งานในภาคส่วนต่าง ๆ ต่างคาดหวังต่อระบบ ICT ที่สามารถตอบสนองต่อความต้องการได้อย่างรวดเร็ว มีความยืดหยุ่นในการจัดสรรทรัพยากร และรองรับปริมาณข้อมูลจำนวนมากที่เกิดขึ้นอย่างต่อเนื่อง ความคาดหวังเหล่านี้ไม่เพียงต้องการเทคโนโลยีที่มีประสิทธิภาพ แต่ยังต้องสามารถปรับขยายการใช้งานได้ตามบริบทของผู้ใช้งานในแต่ละกลุ่มธุรกิจได้อย่างเหมาะสม โดยหนึ่งในกลุ่มเทคโนโลยีที่ได้เข้ามามีบทบาทสำคัญต่อการแก้ไขปัญหาข้างต้น และ

ปรับเปลี่ยนโครงสร้างพื้นฐานเครือข่ายจากรูปแบบดั้งเดิมไปสู่โครงสร้างที่กำหนดโดยซอฟต์แวร์ (Software-Defined Infrastructure) คือ เทคโนโลยีการจำลองระบบเสมือนจริง ประกอบด้วย เทคโนโลยีการจำลองฟังก์ชันเครือข่ายเสมือน (Network Functions Virtualization: NFV), การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) และการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) โดยเทคโนโลยีเหล่านี้ได้รับการยอมรับว่าเป็นกลไกหลักในการเตรียมความพร้อมสำหรับระบบ 5G (5G services) และการให้บริการยุคใหม่ในบริบทต่าง ๆ อาทิ Internet of Things (IoT), smart city และ smart factory เป็นต้น โดยแนวคิดของเทคโนโลยี NFV เป็นการแยกการทำงานของฟังก์ชันเครือข่ายออกจากฮาร์ดแวร์เฉพาะ ให้สามารถนำไปดำเนินงานเป็นซอฟต์แวร์ที่รันอยู่บนโครงสร้างพื้นฐานเสมือนจริง เช่น ระบบคลาวด์ (Cloud infrastructure) ได้ ซึ่งทำให้เกิดรูปแบบการให้บริการในลักษณะการให้บริการเน็ตเวิร์คเซอร์วิสเสมือน (Virtualized Network Functions: VNFs) โดยจุดเด่นของ NFV คือ ขีดความสามารถในการจำลองเน็ตเวิร์คเซอร์วิสได้อย่างรวดเร็ว (rapid deployment) การปรับขยายทรัพยากรได้ตามต้องการ (dynamic scaling) และสามารถลดค่าใช้จ่ายทั้งด้านการลงทุน (Capital Expenditure: CapEx) และค่าดำเนินการ (Operational Expenditure: OpEx) ได้อย่างมีนัยสำคัญ ในขณะที่เทคโนโลยี SDN มีบทบาทเสริมในด้านการบริหารจัดการเครือข่ายผ่านซอฟต์แวร์ โดยสามารถควบคุมเส้นทางการรับส่งข้อมูล ตั้งค่าการจัดลำดับความสำคัญของทราฟฟิก และกำหนดนโยบายการไหลของข้อมูลให้เป็นไปตามความต้องการของระบบได้อย่างยืดหยุ่น

ในอีกมุมหนึ่ง แม้เทคโนโลยีดังกล่าวฯ มีศักยภาพในการยกระดับประสิทธิภาพของการบริหารจัดการเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสผ่านกระบวนทัศน์ใหม่ที่ยืดหยุ่น ปรับขยายได้ และช่วยลดต้นทุนด้านการลงทุนและดำเนินงานได้อย่างชัดเจน แต่ในบริบทประเทศไทยกลับยังคงประสบกับอุปสรรคสำคัญหลายประการ ซึ่งส่งผลให้การพัฒนาเทคโนโลยีและนวัตกรรมที่เกี่ยวข้องยังคงอยู่ในช่วงเริ่มต้นและไม่แพร่หลายเท่าที่ควร อุปสรรคเหล่านี้ได้แก่ การเผชิญกับความท้าทายด้านองค์ความรู้ ความเชี่ยวชาญเฉพาะทาง และความพร้อมของบุคลากรในประเทศ ตลอดจนการพึ่งพาเทคโนโลยีและโซลูชันจากต่างประเทศเป็นหลัก ซึ่งไม่เพียงส่งผลกระทบด้านต้นทุนที่สูงขึ้น แต่ยังส่งผลกระทบต่อความสามารถในการพึ่งพาตนเองและการต่อยอดเทคโนโลยีในระดับประเทศ นอกเหนือจากนี้ ยังพบว่าองค์กรจำนวนมากในประเทศยังขาดความเข้าใจในศักยภาพของเทคโนโลยีดังกล่าวฯ และไม่สามารถกำหนดกรณีศึกษา (Use case) ที่ชัดเจนและเหมาะสมกับลักษณะงานหรือบริบทขององค์กรตนเองได้ ทำให้การตัดสินใจทำเทคโนโลยี NFV และ SDN มาใช้งานอย่างจริงจังยังคงเป็นเรื่องใหม่และท้าทาย

อย่างไรก็ตาม หากพิจารณาในบริบทระดับนานาชาติ พบว่าหลายประเทศในภูมิภาคยุโรปและเอเชียได้ให้ความสำคัญกับการวิจัย พัฒนา และทดสอบระบบที่ใช้เทคโนโลยี NFV และ SDN อย่างเป็นรูปธรรม โดยตระหนักถึงศักยภาพของเทคโนโลยีเหล่านี้ในการยกระดับความสามารถด้าน ICT ให้มีความคล่องตัว ประหยัด และสามารถตอบสนองความต้องการของแอปพลิเคชันยุคใหม่ที่ต้องการความเร็วสูง ความเสถียร และความปลอดภัยสูง เช่น เครือข่าย 5G เมืองอัจฉริยะ รวมถึงระบบควบคุมที่มีความสำคัญต่อภารกิจ (Mission-critical systems) ปัจจัยเหล่านี้ส่งผลให้หลายประเทศจึงได้ให้ความสำคัญกับการลงทุนด้านการวิจัย พัฒนา และทดสอบระบบจริง (Testbed) ที่นำเทคโนโลยี NFV และ SDN มาใช้อย่างเป็นรูปธรรม โดยมีเป้าหมายหลักเพื่อสร้างความพร้อมในการรองรับบริการดิจิทัลแห่งอนาคต

จากข้อเท็จจริงข้างต้น จึงสะท้อนให้เห็นถึงความจำเป็นเร่งด่วนของประเทศไทยในการเร่งวางรากฐานและสร้างต้นแบบการประยุกต์ใช้เทคโนโลยี NFV และ SDN ให้เห็นผลเชิงปฏิบัติอย่างชัดเจน โดยเฉพาะในระดับองค์กรหรือหน่วยงานที่เกี่ยวข้อง การพัฒนาโครงการนำร่องในลักษณะดังกล่าวฯ ไม่เพียงแต่ช่วยยกระดับความเข้าใจในหมู่ผู้บริหารและบุคลากรด้านเทคนิค แต่ยังทำหน้าที่เป็นกลไกสำคัญในการจุดประกายการพัฒนาระบบโครงสร้างพื้นฐานดิจิทัลของประเทศให้สามารถแข่งขันได้ในเวทีสากล และนำไปสู่การเติบโตอย่างยั่งยืนในระยะยาว ด้วยเหตุนี้ โครงการวิจัยจึงมีเป้าหมายในการออกแบบและพัฒนาแพลตฟอร์มบริหารจัดการฟังก์ชันเครือข่ายเสมือน (VNFs as a Service: VNFaaS) เพื่อเป็นต้นแบบในการสร้างระบบนิเวศเครือข่ายเสมือนจริง (Virtualized Network Ecosystem) ที่สามารถตอบสนองต่อความต้องการของภาคอุตสาหกรรมและหน่วยงานรัฐในการเตรียมความพร้อมเข้าสู่ยุคโครงสร้างเครือข่าย 5G ในอนาคต โดยมุ่งเน้นเป้าหมายสำคัญ 3 ประการ ได้แก่

1. การพัฒนาโครงสร้างระบบบริหารจัดการและควบคุมทรัพยากรเครือข่ายแบบอัตโนมัติ (Management and Orchestration Framework) เพื่อรองรับการจัดสรรและควบคุมทรัพยากรโครงสร้างพื้นฐานของระบบได้อย่างมีประสิทธิภาพ
2. การจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสเสมือน โดยเฉพาะอย่างยิ่งเน็ตเวิร์คเซอร์วิสที่สนับสนุนงานด้านความปลอดภัย เช่น Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber Traps และ Traffic Analyzers บนระบบคลาวด์ที่สามารถเข้าถึงและใช้งานได้ตามความต้องการของแต่ละองค์กร
3. การให้บริการแบบ on-demand ที่สามารถสร้างและปรับเปลี่ยนการใช้งานเน็ตเวิร์คเซอร์วิสตามนโยบายของผู้ใช้งาน (Service Function Chaining) อย่างยืดหยุ่นและสอดคล้องกับความต้องการของแต่ละกลุ่มอุตสาหกรรม

แนวคิดและแนวทางการดำเนินงานของโครงการนี้จึงนับเป็นก้าวแรกที่มีนัยสำคัญในการวางรากฐานโครงสร้างพื้นฐานเครือข่ายดิจิทัลของประเทศ โดยเฉพาะอย่างยิ่งการยกระดับงานด้านความมั่นคงปลอดภัยทางไซเบอร์ และยังเป็นเครื่องมือในการเสริมสร้างความสามารถในการแข่งขันของประเทศในยุคเศรษฐกิจและสังคมดิจิทัลได้อย่างยั่งยืนและระยะยาว

**2. ขอบเขตการศึกษา**

โครงการวิจัยนี้มีวัตถุประสงค์หลักเพื่อวิจัย ออกแบบ พัฒนา และสร้างนวัตกรรมสำหรับบริหารและจัดการทรัพยากรเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสเสมือน (Virtual Network Functions as a Service: VNFaaS) อัตโนมัติ โดยจัดเตรียมระบบนิเวศเสมือนจริงที่สนับสนุนการทำงานของเทคโนโลยี NFV และ SDN โดยเฉพาะอย่างยิ่งการจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสที่สนับสนุนงานด้านความปลอดภัยเป็นหลัก เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งานสามารถเรียกใช้บริการด้านความปลอดภัยเครือข่าย (security-related VNFs) ได้ตามความต้องการและสอดคล้องกับบริบทของแต่ละองค์กร อย่างไรก็ดี เนื่องจากเทคโนโลยีดังกล่าวยังถือเป็นเทคโนโลยีใหม่ในบริบทของประเทศไทย โครงการจึงแบ่งการดำเนินงานออกเป็น 4 ระยะย่อย เพื่อวางรากฐานความเข้าใจและการใช้งานจริง ดังนี้

4. ดำเนินการศึกษาเพื่อสำรวจเทคโนโลยีการจำลองระบบเสมือนจริงเชิงลึก โดยคลอบคลุมถึงสถาปัตยกรรม ขีดความสามารถ ช่องโหว่และภัยคุกคาม ตลอดจนบริบทอื่นๆ ที่เกี่ยวข้อง อันเป็นประโยชน์ต่อการสร้างองค์ความรู้ในเทคโนโลยีการจำลองระบบเสมือนจริง
5. ดำเนินการพิสูจน์แนวคิด (proof of concept) และศึกษาถึงความเป็นไปได้สู่การใช้งานจริง โดยการพัฒนาแพลตฟอร์มสำหรับบริหารจัดการ และให้บริการเน็ตเวิร์คเซอร์วิสแบบเสมือนจริงเพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต โดย use case ที่สนใจในระยะเริ่มต้นของการพัฒนาโครงการนี้ (ระยะที่ 1: Proof of Concept) คือ การจัดเตรียมและให้บริการเน็ตเวิร์คเซอร์วิสเสมือนด้านความปลอดภัยต่างๆ ไว้บนคลาวด์ ดังเช่น DPI (e.g., Suricata), IDS (e.g., Snort), Cyber traps (e.g., Honeypots), Traffic analyzer (e.g., Zeek) ฯลฯ เพื่อให้ผู้รับบริการเครือข่ายหรือผู้ที่เกี่ยวข้องสามารถเรียกใช้และใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสดังกล่าวฯ ได้ตรงกับตามความต้องการใช้งาน (security requirements) สำหรับการรักษาความมปลอดภัยภายในองค์กร โดยการพัฒนาระบบจะประกอบด้วย 2 ระบบย่อยโดยมีรายละเอียดดังนี้
    a. ระบบหลังบ้าน (Back-end framework)
        i. พัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง (management and orchestration platform) เพื่อทำหน้าที่หลักในการจัดเตรียมทรัพยากรต่างๆ ได้แก่ CPU, หน่วยความจำ พื้นที่จัดเก็บข้อมูล และระบบเครือข่าย ให้พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิสต่างๆ ตลอดจนควบคุมลำดับการทำงานให้เป็นไปอย่างอัตโนมัติสอดคล้องกับนโยบายที่กำหนดไว้
        ii. พัฒนาส่วนเชื่อมต่อเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNFs) ด้านความปลอดภัยต่างๆ เช่น Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Cyber traps, Traffic analyzer ฯลฯ เพื่อให้องค์กรที่รับบริการสามารถดำเนินการตั้งค่าและเรียกใช้งานเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยที่ต้องการได้อย่างเหมาะสมสำหรับรักษาความปลอดภัยในองค์กร
        iii. พัฒนาช่องทาง REST-based APIs ไว้สำหรับดึงข้อมูลจากเน็ตเวิร์คเซอร์วิสมาต่างๆ มาแสดงผลผ่าน dashboard เช่น ข้อมูลแสดงผลประเภทภัยคุกคามที่ตรวจพบ จำนวนภัยคุกคามที่เกิดขึ้นในช่วงเวลาต่างๆ ฯลฯ
        iv. พัฒนาโมดูลสำหรับตรวจสอบการใช้งานทรัพยากรภายในระบบ (resource monitoring) เพื่อเพิ่มขีดความสามารถในการปรับเพิ่ม/ลด หรือขยายทรัพยากรให้กับเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น (scaling in/out network resources) เพื่อให้มีความเหมาะสมและสอดคล้องกับปริมาณงานที่เข้ามาในช่วงเวลานั้นๆ
    b. ระบบหน้าบ้าน (Front-end framework)

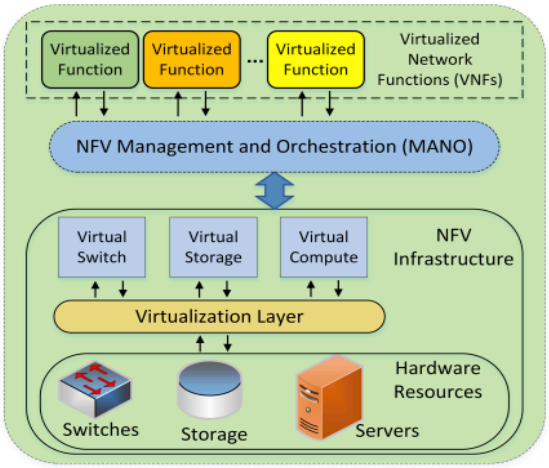
i. พัฒนา web portal ซึ่งเป็นส่วนการจัดการข้อมูลหน่วยงานผู้ใช้บริการและจัดการสิทธิ์การเข้าถึงข้อมูลตามบทบาทและหน้าที่ที่ได้รับ รวมถึงการรับเข้าข้อมูลจากผู้ใช้งาน เช่น ข้อมูลการตั้งค่านโยบายความปลอดภัย

ii. พัฒนา dashboard เพื่อนำผลลัพธ์ที่ได้จากการประมวลผลของเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ ที่ถูกเชื่อมต่อเข้าด้วยกันในระบบหลังบ้าน (back-end) มาแสดงผลในรูปแบบต่างๆ ที่ผู้ใช้งานสามารถเข้าใจได้ง่าย เช่น ในรูปแบบกราฟแสดงผลหรือรายงาน ฯลฯ

6. ดำเนินการทดสอบประสิทธิภาพของระบบตามคุณสมบัติ 3 ด้าน ได้แก่ การทำงานเชิงหน้าที่ (functionality) ด้านการใช้งาน (feasibility) และด้านประสิทธิภาพ (performance)
7. ถ่ายทอดองค์ความรู้และขยายนวัตกรรมไปสู่การใช้งานจริง มุ่งเน้นการเผยแพร่ประชาสัมพันธ์ผลงาน การถ่ายทอดองค์ความรู้จากงานวิจัยไปสู่ผู้มีส่วนเกี่ยวข้อง และสนับสนุนให้เกิดการต่อยอดและขยายผลนวัตกรรมไปสู่การใช้งานจริงอย่างแพร่หลายในวงกว้างไปสู่ภาคส่วนต่างๆ ทั้งภาคอุตสาหกรรม ภาคอุดมศึกษา ภาครัฐ และภาคเอกชน

## 3. การทบทวนวรรณกรรม

ในหัวข้อนี้จะเกี่ยวข้องกับการสำรวจ ศึกษา และคัดเลือกเครื่องมือโอเพ่นซอร์ส (Open-source software) ที่เหมาะสมเพื่อใช้เป็นองค์ประกอบในการออกแบบและพัฒนาแพลตฟอร์ม โดยจะทำการวิเคราะห์เครื่องมือที่เกี่ยวข้องใน 3 หมวดหมู่ตามลักษณะของชั้นการทำงาน ดังนี้ (แสดงดังภาพที่ 1)




*รูปที่ 1: สถาปัตยกรรมการออกแบบระบบจำลองเครือข่ายเสมือนจริง (NFV)*

### 3.1 เครื่องมือที่สนับสนุนชั้นโครงสร้างพื้นฐาน (NFV infrastructure layer)

หากอ้างอิงสถาปัตยกรรมการออกแบบระบบจำลองเครือข่ายเสมือนจริง (รูปที่ 1) จะพบว่า หน้าที่หลักของชั้นนี้ คือ การจัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรเครือข่ายต่างๆ ได้แก่ หน่วยประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูล และระบบเครือข่ายต่างๆ ที่จำเป็นต้องใช้สำหรับการจำลองเน็ตเวิร์คเซอร์วิส กล่าวอีกนัยหนึ่ง หน้าที่หลักของชั้นนี้ คือ (1) การจัดสรรทรัพยากรเครือข่ายตามการร้องขอ (2) การจัดเตรียมเครื่อง

เสมือนจริง (VMs) ให้พร้อมสำหรับการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ และ (3) ดำเนินการตรวจสอบและวิเคราะห์ประสิทธิภาพการใช้งานทรัพยากรเครือข่ายในแต่ละเน็ตเวิร์คเซอร์วิส เพื่อปรับเพิ่ม/ลดจำนวนทรัพยากรประมวลผลและเครือข่ายให้มีความเหมาะสมต่อจำนวนเวิร์คโหลดที่เข้ามาในช่วงเวลานั้นๆ โดยในปัจจุบันนี้ พบว่ามีหลายหน่วยงานที่ดำเนินการให้บริการโครงสร้างพื้นฐานระบบคลาวด์เพื่อสนับสนุนการทำงานในชั้นนี้ ได้แก่ Amazon Web Services (AWS) [1-2], Google Cloud Platform (GCP) [3-4], Microsoft Azure [5], VMware Cloud [6] และ OpenStack [7-8] โดยเฉพาะอย่างยิ่ง OpenStack ซึ่งเป็นโอเพ่นซอร์สแพลตฟอร์มที่ได้รับความนิยมสูงสุดในการจัดการโครงสร้างพื้นฐานแบบเสมือน

### 3.2 เครื่องมือที่สนับสนุนชั้นเน็ตเวิร์คเซอร์วิสแบบเสมือน (VNF layer)

ชั้นนี้เป็นส่วนสำคัญในสถาปัตยกรรมของ NFV ทำหน้าที่ คือ (1) การจำลองเน็ตเวิร์คเซอร์วิสประเภทต่างๆ เช่น Firewall, IDS, DPI, ฯลฯ และจัดการวงจรชีวิตของแต่ละเน็ตเวิร์คเซอร์วิส (VNF) ที่ได้ถูกจำลองขึ้นให้มีขีดความสามารถในการทำงานบนทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ได้กำหนดขึ้น (2) ดำเนินการตั้งค่าแต่ละเน็ตเวิร์คเซอร์วิสตามความต้องการใช้งานหรือตามนโยบาย (policy) ที่ผู้ดูแลระบบกำหนดไว้ (3) ทำการปรับลดหรือขยายเน็ตเวิร์คเซอร์วิส (scaling in/out VNFs) เมื่อประสิทธิภาพของทรัพยากรที่ใช้สำหรับการประมวลผลเกินกว่าค่า threshold ที่กำหนดไว้ (4) อัพเกรดซอฟต์แวร์ที่เกี่ยวข้อง และ (5) ถอดถอนการติดตั้งเน็ตเวิร์คเซอร์วิสเมื่อไม่มีการใช้งาน กล่าวอีกนัยหนึ่ง ชั้นนี้เปรียบเสมือนแหล่งรวมเน็ตเวิร์คเซอร์วิสแบบเสมือนประเภทต่างๆ ที่ต้องการจำลองขึ้นไว้ด้วยกัน โดยคลอบคลุมถึง (1) กลุ่มเน็ตเวิร์คเซอร์วิสสนับสนุนงานด้านความปลอดภัยบนเครือข่าย ได้แก่ firewall, IDS, DPI, ฯลฯ (2) กลุ่มเน็ตเวิร์คเซอร์วิสที่มุ่งเน้นการเพิ่มคุณภาพการให้บริการและการตอบสนองต่อการให้บริการเครือข่าย (QoS) ได้แก่ load balancer, WAN optimizer, web proxy, CDN, ฯลฯ และ (3) กลุ่มเน็ตเวิร์คเซอร์วิสที่เน้นการจัดเตรียมโครงสร้างพื้นฐานเครือข่ายเพื่อสนับสนุนเครือข่าย 5G ในอนาคต เช่น switch, packet gateway, HSS, MME ฯลฯ อย่างไรก็ตาม โครงการนี้ ให้ความสำคัญโดยมุ่งเน้นการสำรวจเน็ตเวิร์คเซอร์วิสที่สนับสนุนงานด้านความปลอดภัยเป็นหลัก ได้แก่ Intrusion Detection System (IDS) (e.g., Snort [9-10]), Deep Packet Inspection (DPI) (e.g., Suricata [11-13]), Cyber traps (e.g., Honeypot [14-15]), Security Information and Event Management (SIEM) (e.g., Wazuh [16-17]) และ Network traffic analyzer (e.g., Zeek [18-19])

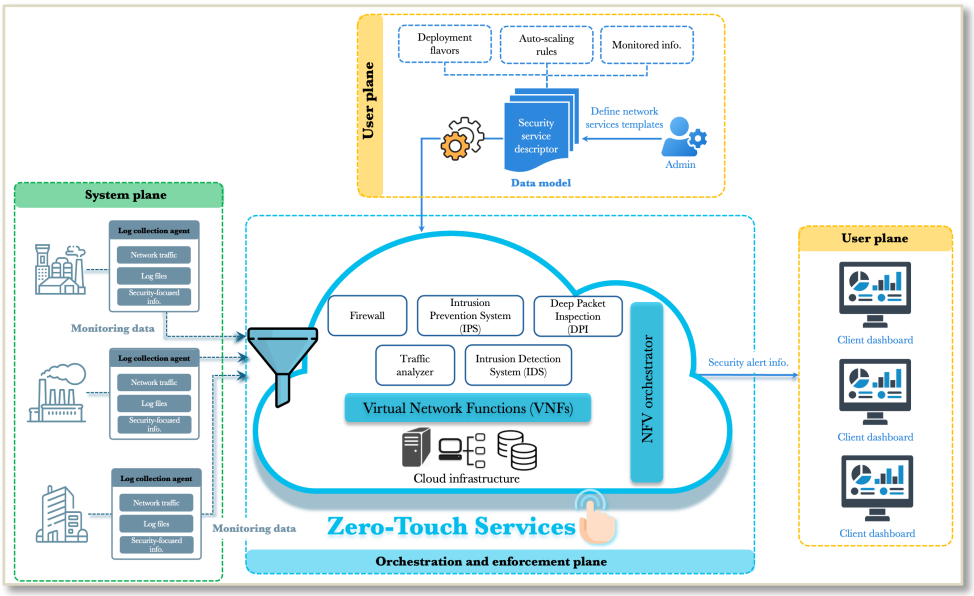
### 3.3 เครื่องมือที่สนับสนุนชั้นบริหารและจัดการ (Management and Orchestration Layer: MANO layer)

ชั้นนี้มีบทบาทสำคัญในการควบคุมลำดับการทำงานภายในระบบ และประสานให้การทำงานระหว่างโมดูลต่างๆ ให้เป็นไปอย่างอัตโนมัติ สอดคล้องกับบริบทความต้องการใช้งานหรือนโยบายที่กำหนดไว้ โดยหน้าที่หลักๆ ของชั้นนี้ คือ (1) บริหารและจัดการเน็ตเวิร์คโทโพโลยีในภาพรวม (2) ตรวจสอบสิทธิการเข้าถึงทรัพยากรเครือข่ายเมื่อมีการร้องขอ (3) ประสานงานกับโมดูลต่างๆ ที่เกี่ยวข้องทั้งในชั้น infrastructure layer เพื่อจัดสรรทรัพยากรเครือข่ายให้พร้อมสำหรับการใช้งาน และชั้น VNF layer เพื่อจำลองเน็ตเวิร์คเซอร์วิสตามการร้องขอ และ (4) บริหารและจัดการวงจรชีวิตของเน็ตเวิร์คเซอร์วิสที่ได้ถูกจำลองขึ้น โดยเครื่องมือ open source ที่ทำหน้าที่บริหาร

และจัดการเน็ตเวิร์คเซอร์วิส และได้ถูกนำมาใช้งานอย่างแพร่หลาย คือ Open source MANO (OSM) [20-23] และ OpenStack Tacker [24-25] ทั้งสองเครื่องมือต่างมีเป้าหมายหลักที่เหมือนกัน คือ การพัฒนาแพลตฟอร์มสนับสนุนการบริหารและจัดการเน็ตเวิร์คเซอร์วิสที่สามารถรองรับมาตราฐานด้านโทรคมนาคมยุโรป (European Telecommunications Standards Institute: ETSI NFV [24]) ได้

## 4. การออกแบบสถาปัตยกรรม

รูปที่ 2 แสดงการออกแบบสถาปัตยกรรมโดยภาพรวม ซึ่งแบ่งการทำงานออกเป็น 3 ระดับชั้น ดังต่อไปนี้

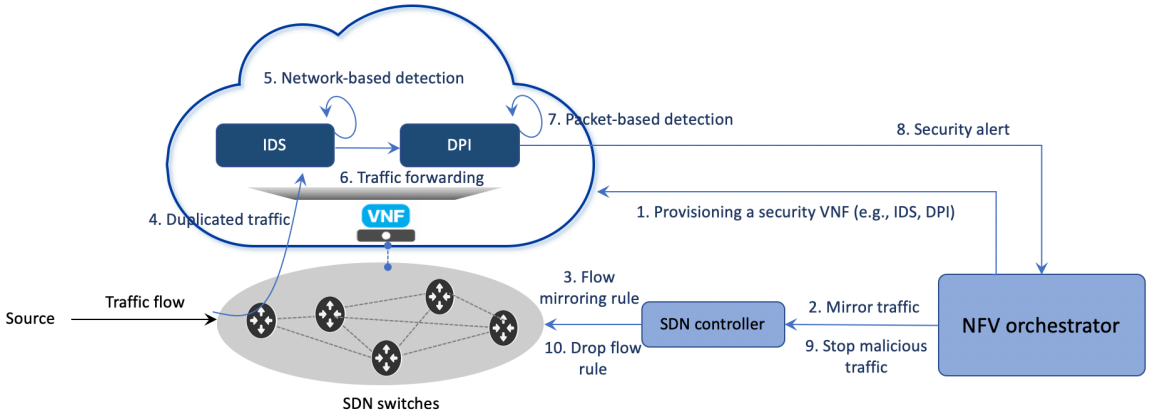



*รูปที่ 2: แนวคิดการออกแบบสถาปัตยกรรม*

- **System plane:** เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริงกับเครือข่ายผู้รับบริการ เนื่องจากการทำงานของแพลตฟอร์มซึ่งอยู่ในชั้นถัดไปอาศัยการวิเคราะห์ข้อมูลที่มาจากเครือข่ายผู้รับบริการแบบ real-time เพื่อตรวจสอบและวิเคราะห์หาความผิดปกติถึงรูปแบบพฤติกรรมในการรับส่งแพ็กเก็ตข้อมูล (packet) ในเครือข่ายว่าก่อให้เกิดช่องโหว่หรือความเสี่ยงด้านความปลอดภัยหรือไม่ ทำให้ผู้ดูแลระบบสามารถตัดสินใจได้อย่างทันท่วงทีในการดำเนินการยกระดับความปลอดภัยให้แก่เครือข่ายภายใต้การกำกับดูแล โดยขั้นตอนพัฒนาระบบจะกำหนดให้แต่ละ network domain องค์กรที่สนใจทำการติดตั้งและเชื่อมต่อกับ Log collection engine ซึ่งเปรียบเสมือน Agent service ถูกติดตั้งที่ network domain ขององค์กรผู้รับบริการ เพื่อทำหน้าที่ในการรวบรวม logs การทำงานต่างๆ ประกอบด้วย network traffic, log files, critical files ที่สนใจ เช่น failed-login attempts, suspicious network traffic, ฯลฯ และส่งต่อไปยังแพลตฟอร์มฯ เพื่อทำการตรวจสอบและวิเคราะห์หาความผิดปกติหรือความเสี่ยงด้านความปลอดภัยที่อาจจะเกิดขึ้นต่อไป
    - **Orchestration and enforcement plane:** เป็นชั้นที่แพลตฟอร์มฯ ติดตั้งอยู่ เพื่อทำหน้าที่ในการตรวจสอบและวิเคราะห์หาความเสี่ยงและช่องโหว่ด้านความปลอดภัยบนระบบเครือข่ายของ

ผู้รับบริการจากข้อมูล logs ที่ได้รับมาจาก Log collection engine ที่ได้ถูกติดตั้งในชั้น System plane โดยรูปแบบการทำงานของแพลตฟอร์มมีหน้าที่หลัก 2 ประการ คือ

- **Orchestration service:** ทำหน้าที่ (1) จัดเตรียมโครงสร้างพื้นฐานคลาวด์และทรัพยากรเครือข่ายต่างๆ ได้แก่ หน่วยประมวลผล หน่วยความจำ แหล่งจัดเก็บข้อมูล และระบบเครือข่ายต่างๆ ที่จำเป็นต้องใช้ในการจำลองเน็ตเวิร์คเซอร์วิสที่ต้องการ และ (2) ทำการจำลอง ตั้งค่า และให้บริการเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยต่างๆ เพื่อให้องค์กรที่รับบริการสามารถเรียกใช้ประโยชน์จากเน็ตเวิร์คเซอร์วิสที่จำลองขึ้นได้ เช่น การจำลองระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ดังเช่น Snort เพื่อคอยตรวจสอบและวิเคราะห์ถึงความผิดปกติบนระบบเครือข่ายของผู้รับบริการแบบ real-time จากข้อมูล logs ที่ได้รับ โดยวิธีการตรวจสอบนั้นจะอ้างอิงข้อมูลลายเซ็นหรือลักษณะเฉพาะของการบุกรุก (signature-based IDS) และนโยบายความปลอดภัย (security policy) ที่ผู้ใช้งานได้ทำการกำหนดไว้ล่วงหน้ามาช่วยในการตรวจสอบ
- **Enforcement service:** หากระบบตรวจพบความผิดปกติบนระบบเครือข่ายของผู้รับบริการ ระบบจะดำเนินการแจ้งเตือนให้ผู้ที่เกี่ยวข้องทราบ และดำเนินการออกคำสั่งให้สร้างเน็ตเวิร์คเซอร์วิสด้านความปลอดภัย (VNFs) ที่เกี่ยวข้อง หรือปรับปรุงการตั้งค่านโยบายการรักษาความปลอดใหม่ เพื่อจัดการกับความผิดปกติที่ตรวจพบโดยทันที รูปที่ 3 แสดงตัวอย่างแนวทางการตรวจสอบและการป้องกันเมื่อตรวจพบความผิดปกติบนระบบเครือข่ายของหน่วยงานผู้ขอรับบริการ จะเห็นได้ว่าเบื้องต้นระบบจะทำการ mirror traffic จากหน่วยงานที่สนใจ (source) และส่ง traffic ดังกล่าวฯ ไปยังระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ดังเช่น Snort และระบบตรวจจับการบุกรุกเชิงลึก (Deep Packet Inspection: DPI) [5] ดังเช่น Suricata ตามลำดับ เพื่อทำการตรวจสอบและ




*รูปที่ 3: แสดงตัวอย่างการแผนภาพกระแสข้อมูลการบริหาร จัดการ และตรวสอบความปลอดภัยบนเครือข่าย*

[5] หมายเหตุ: ความแตกต่างระหว่างทั้งสอบประเภท คือ (1) การตรวจจับแบบ IDS จะใช้วิธีการตรวจสอบและวิเคราะห์ข้อมูลเครือข่ายเพื่อตรวจจับเหตุการณ์หรือกิจกรรมที่อาจก่อให้เกิดการบุกรุก โดยอ้างอิงลายเซ็นหรือลักษณะเฉพาะของการบุกรุกที่เป็นที่รู้จักและเก็บไว้ล่วงหน้า เมื่อหนึ่งเมื่อใดที่ตรวจพบเหตุการณ์ที่ตรงกับลายเซ็นเหล่านี้ ก็สามารถสรุปได้ว่ามีความเสี่ยงเกิดขึ้น ในขณะที่ (2) การตรวจจับแบบ DPI คือ การตรวจสอบแพ็กเก็ตข้อมูลเชิงลึกในระดับ packet headers และ packet contents เพื่อค้นหามัลแวร์ สแปม หรือการโจมตีที่อาจแอบแฝงมากับรูปแบบของ traffic ปกติ

วิเคราะห์หาความผิดปกติที่เกิดขึ้น หลังจากนั้นหากตรวจพบถึงความผิดปกติ ระบบจะส่งการแจ้งเตือนกลับมายัง NFV orchestrator เพื่อดำเนินการออกคำสั่งไปยัง SDN controller ปรับปรุงการตั้งค่า policy และ forwarding rules ใหม่

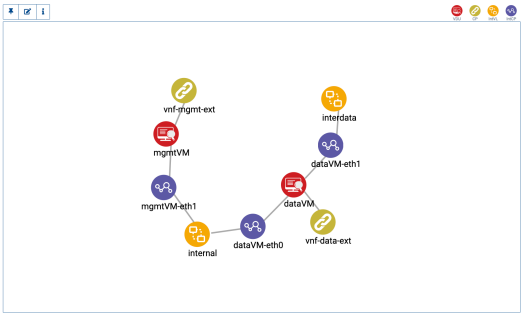
- **User plane:** เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มและหน้าเว็บการจัดการและการแสดงผล เพื่ออำนวยความสะดวกแก่ดูและระบบและผู้รับบริการในการเรียกดูสถานะต่างๆ ที่เกี่ยวข้องกับการดำเนินการ โดยสิทธิ์ในการเข้าถึงจำแนกผู้ใช้งานออกเป็น 2 ประเภท คือ (1) แพลตฟอร์มอำนวยความสะดวกให้ผู้ดูแลระบบสามารถกำหนดหรือออกแบบ network service descriptors[6] ได้ตามที่ต้องการ รูปที่ 4 และ 5 แสดงตัวอย่างและผล network service descriptor ที่มีการระบุลักษณะเฉพาะต่างๆ เช่น ระบบปฏิบัติที่ใช้ จำนวนหน่วยความจำและหน่วยประมวลผล จำนวนพอร์ตการเชื่อมต่อ ฯลฯ ซึ่งเป็นข้อมูลสำคัญในการจำลองเครื่องเสมือน (VM) ฯลฯ (2) Dashboard สำหรับการแสดงผลข้อมูล โดยข้อมูลสามารถแบ่งออกเป็น 2 ประเภท ได้แก่ admin view ที่สามารถมองเห็นสถานะทำงานของระบบหลังบ้าน และ tenant view การแสดงผลข้อมูลภัยคุกคามโดยแบ่งตามลักษณะของผู้ใช้งาน
    - **User plane:** เป็นส่วนที่เชื่อมต่อระหว่างแพลตฟอร์มและหน้าเว็บการจัดการและการแสดงผล เพื่ออำนวยความสะดวกแก่ดูและระบบและผู้รับบริการในการเรียกดูสถานะต่างๆ ที่เกี่ยวข้องกับการดำเนินการ โดยสิทธิ์ในการเข้าถึงจำแนกผู้ใช้งานออกเป็น 2 ประเภท คือ (1) แพลตฟอร์มอำนวยความสะดวกให้ผู้ดูแลระบบสามารถกำหนดหรือออกแบบ network service descriptors[7] ได้ตามที่ต้องการ รูปที่ 4 และ 5 แสดงตัวอย่างและผล network service descriptor ที่มีการระบุลักษณะเฉพาะต่างๆ เช่น ระบบปฏิบัติที่ใช้ จำนวนหน่วยความจำและหน่วยประมวลผล จำนวนพอร์ต

[6] หมายเหตุ: Network service descriptors ในที่นี้คือ การเขียนชุดคำสั่งต่างๆ ที่เกี่ยวข้องในการจัดสรรทรัพยากรโครงสร้างพื้นฐานคลาวด์ การสร้างเครื่องเสมือน (Virtual Machines: VMs) การเชื่อมต่อเน็ตเวิร์ค และการจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง โดยชุดคำสั่งดังกล่าวฯ จะถูกเขียนให้อยู่ในรูปแบบ data sequentialization model ด้วยภาษาที่เข้าใจได้ง่าย เช่น YAML (Yet Another Markup Language)

[7] หมายเหตุ: Network service descriptors ในที่นี้คือ การเขียนชุดคำสั่งต่างๆ ที่เกี่ยวข้องในการจัดสรรทรัพยากรโครงสร้างพื้นฐานคลาวด์ การสร้างเครื่องเสมือน (Virtual Machines: VMs) การเชื่อมต่อเน็ตเวิร์ค และการจำลองเน็ตเวิร์คเซอร์วิสเสมือนจริง โดยชุดคำสั่งดังกล่าวฯ จะถูกเขียนให้อยู่ในรูปแบบ data sequentialization model ด้วยภาษาที่เข้าใจได้ง่าย เช่น YAML (Yet Another Markup Language)

```
mgmt-cp: vnf-mgmt-ext
product-name: simple_2vm-vnf
sw-image-desc:
- id: ubuntu20.04
  image: ubuntu20.04
  name: ubuntu20.04
- id: ubuntu20.04-azure
  name: ubuntu20.04-azure
  image: Canonical:0001-com-ubuntu-server-focal:20_04-lts:latest
  vim-type: azure
- id: ubuntu20.04-gcp
  name: ubuntu20.04-gcp
  image: ubuntu-os-cloud:image-family:ubuntu-2004-lts
  vim-type: gcp
virtual-compute-desc:
- id: mgmtVM-compute
  virtual-memory:
    size: 1.0
  virtual-cpu:
    num-virtual-cpu: 1
- id: dataVM-compute
  virtual-memory:
    size: 1.0
  virtual-cpu:
    num-virtual-cpu: 1
virtual-storage-desc:
- id: mgmtVM-storage
  size-of-storage: 10
- id: dataVM-storage
  size-of-storage: 10
```

*รูปที่ 3: ตัวอย่าง network service descriptor สำหรับใช้ในการจำลองเครื่องเสมือน (VM)*




*รูปที่ 4: แสดงตัวอย่างผลการจำลองเครื่องเสมือน (VM) ตาม network service descriptor ที่ได้กำหนดขึ้น*

การเชื่อมต่อ ฯลฯ ซึ่งเป็นข้อมูลสำคัญในการจำลองเครื่องเสมือน (VM) ฯลฯ และ (2) หน่วยงานผู้ขอรับบริการสามารถเรียกดูภาพรวมสถานการณ์การทำงานของเครือข่าย การแสดงผลข้อมูลที่ได้จากการวิเคราะห์ภัยคุกคาม และผลการตรวจสอบความผิดปกติบนระบบเครือข่าย ฯลฯ ผ่านทาง dashboard และ โดยการติดต่อสื่อสารระหว่าง 2 ชั้นนี้จะรับส่งข้อมูลผ่าน RESTful API เป็นหลัก

## 5. การทดสอบและวัดประสิทธิภาพ

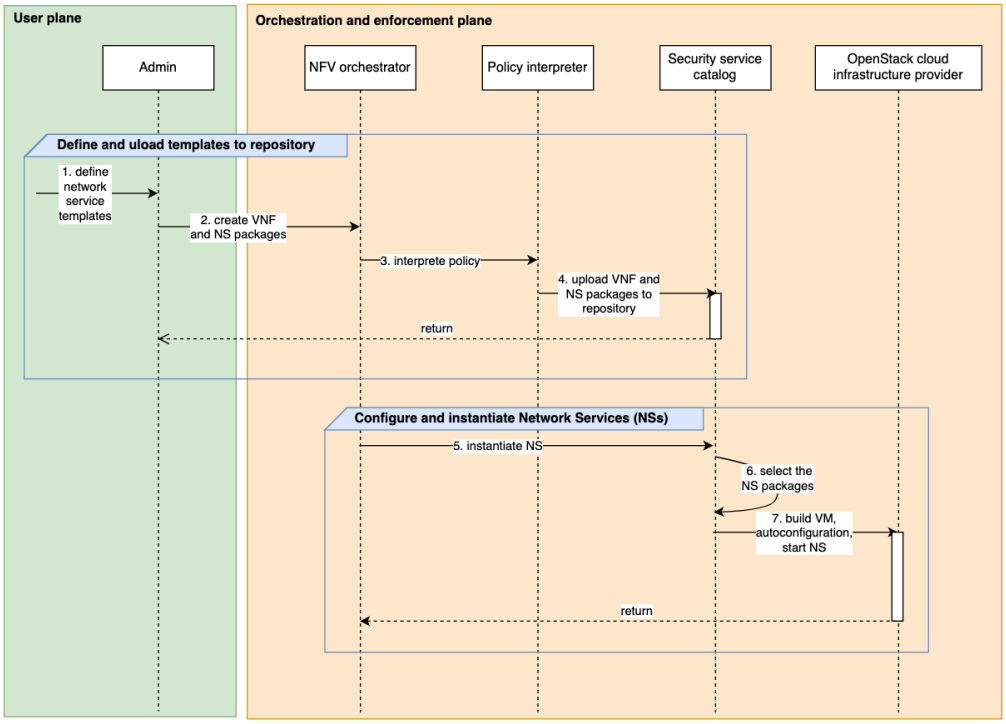
ทีมวิจัยได้แบ่งแนวทางการทดสอบและวัดประสิทธิภาพของระบบ โดยแบ่งเกณฑ์การวัดตามคุณสมบัติ 3 ด้าน ดังนี้ การทำงานเชิงหน้าที่ (functionality) การใช้งาน (feasibility) และการวัดประสิทธิภาพของระบบแบบองค์รวม (performance)

### 5.1 การทำงานเชิงหน้าที่ (funcitonality)

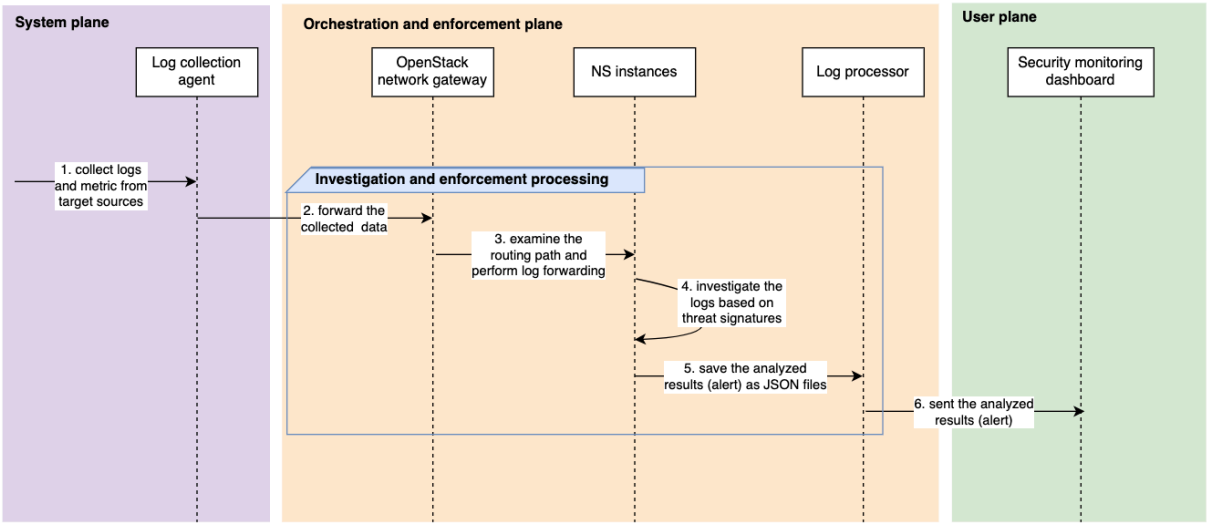
เพื่อให้ง่ายต่อการทำความเข้าใจ การทำงานของแพลตฟอร์มฯ จะแบ่งออกเป็น 2 ระยะหลักๆ ดังนี้

- Pre-configuration and deployment time: โดยในระยะนี้จะเกี่ยวข้องกับการสร้างและออกแบบพิมพ์เขียวเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยประเภทต่างๆ (ได้แก่ Suricata, Snort, Zeek) การตั้งค่านโยบายความปลอดภัยให้กับแต่ละเน็ตเวิร์คเซอร์วิส การจัดเตรียมทรัพยากรโครงสร้างพื้นฐาน การตั้งค่าการเชื่อมต่อเครือข่าย และการจำลองเน็ตเวิร์คเซอร์วิสให้พร้อมสำหรับการใช้งาน รูปที่ 5 แสดงภาพรวมการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้อง เพื่อสร้าง ออกแบบ ตั้งค่า และจำลองเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยดังกล่าวฯ สำหรับรองรับการตรวจสอบและวิเคราะห์ภัยคุกคามที่จะเกิดขึ้นในระยะต่อไป

- Run-time: ระยะนี้มีความเกี่ยวข้องกับการตรวจสอบและวิเคราะห์หาความผิดปกติจากข้อมูล logs ที่ได้รับมาจาก log collection agent ซึ่งได้ถูกติดตั้งไว้ที่ network gateway ของเครือข่ายผู้รับบริการ เพื่อตรวจหาความผิดปกติหรือความเสี่ยงที่อาจก่อให้เกิดภัยคุกคาม ซึ่งผลลัพธ์ที่ได้จากการประมวลผลของแต่ละเน็ตเวิร์คเซอร์วิส จะถูกนำมาแสดงผลในรูปแบบต่างๆ ที่ผู้ใช้สามารถเข้าใจได้ง่าย เช่น กราฟแสดงผลผ่าน dashboard รูปที่ 6 แสดงภาพรวมการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้องกับการวิเคราะห์และตรวจสอบภัยคุกคาม




*รูปที่ 5: แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ*

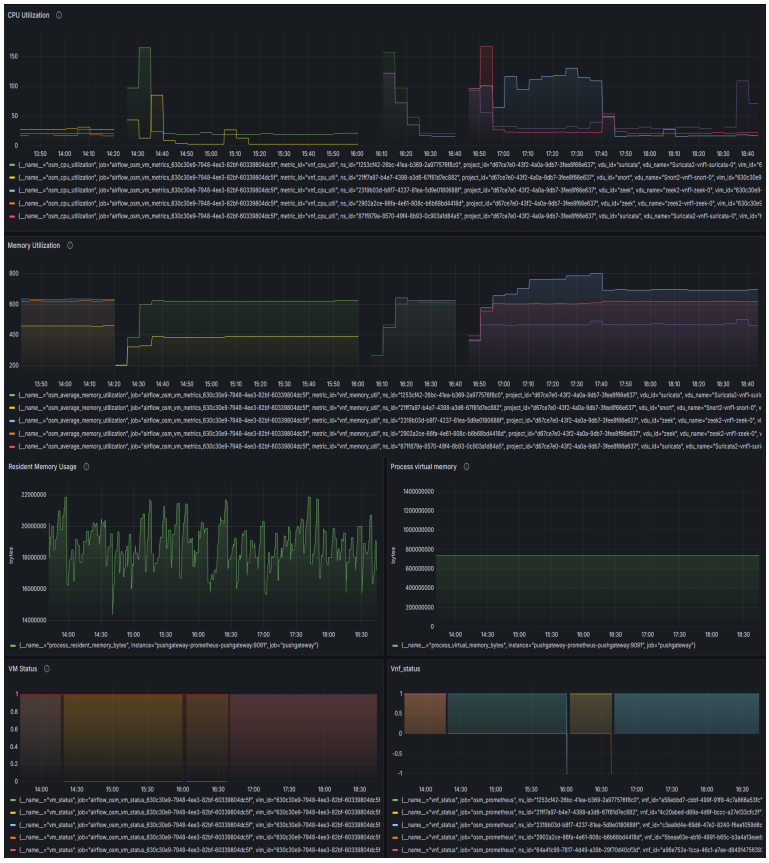



*รูปที่ 6: แสดงการไหลของข้อมูลระหว่างโมดูลต่างๆ ที่เกี่ยวข้องกับการตรวจสอบและวิเคราะห์ความเสี่ยงด้านภัยคุกคาม*

## 5.2 ด้านการใช้งาน

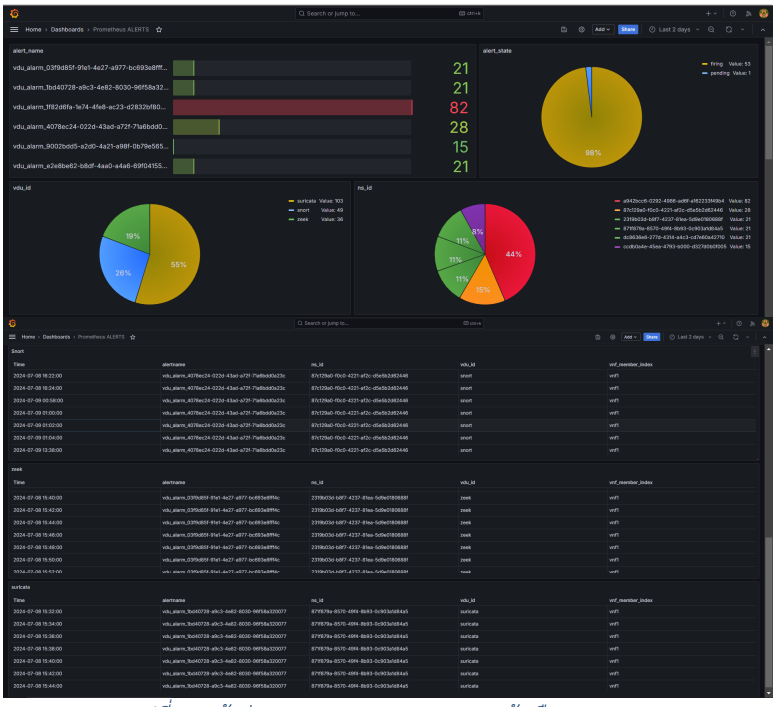
ในหัวข้อนี้จะพูดถึงรายละเอียดการใช้งานและส่วนแสดงผลของระบบ ซึ่งข้อมูลที่นำมาแสดงสามารถแบ่งออกได้เป็น 3 กลุ่มข้อมูลหลักๆ ได้ดังนี้

- *สถานะการทำงานของระบบ:* เป็นข้อมูลที่เกี่ยวข้องกับสถานะทำงานของระบบหลังบ้าน ซึ่งโดยผู้ดูแลระบบสามารถติดตามและตรวจสอบสถานะการทำงานของ instances ต่างๆ ประกอบด้วย CPU utilization, Memory utilization, Resident memory usage, Process virtual memory, VM status และ VNF status (ดังรูปที่ 7)




***รูปที่ 7:*** *หน้าต่าง dashboard แสดงข้อมูลแจ้งสถานะทำงานของ instances ต่างๆ ได้แก่ CPU utilization และ memory utilization, resident memory usage, process virtual memory, VM status และ VNF status*

- *สถานะการแจ้งเตือนของระบบ:* ระบบยังเพิ่มขีดความสามารถในการตรวจสอบ ติดตามประสิทธิภาพการดำเนินงาน และแจ้งเตือน (alerts) แก่ผู้ดูแลหากพบว่ามีความผิดปกติภายในระบบเกิดขึ้น ประกอบด้วย (1) Time วันที่และเวลาที่ข้อมูลหรือเหตุการณ์ที่เกิดการแจ้งเตือน (Alerts) ที่เกิดขึ้น (2) Alert Name ชื่อของการแจ้งเตือนที่เกิดขึ้นในระบบ (3) VDU Id แสดงถึงของ Virtual Deployment Unit (VDU) ที่เกี่ยวข้องกับการแจ้งเตือน ในที่นี้ตัวอย่างเช่น Snort, Zeek, และ Suricata (4) NS Id แสดงถึงรหัสของ Network Service (NS) ที่เกี่ยวข้องกับการแจ้งเตือน และ (5) VNF Member Index แสดง VNF member ที่เกี่ยวข้อง (ดังรูปที่ 8)



*รูปที่ 8: หน้าต่าง dashboard แสดงผลการแจ้งเตือน (alerts)*

- *ผลการวิเคราะห์ภัยคุกคาม:* ข้อมูลที่เกี่ยวข้องกับผลการวิเคราะห์และตรวจสอบภัยคุกคามที่มาจากเน็ตเวิร์คเซอร์วิสที่ได้จำลองขึ้น ในที่นี้ได้แก่ Suricata, Snort และ Zeek (ดังรูปที่ 9)

(a) Suricata (b) Snort (c) Zeek

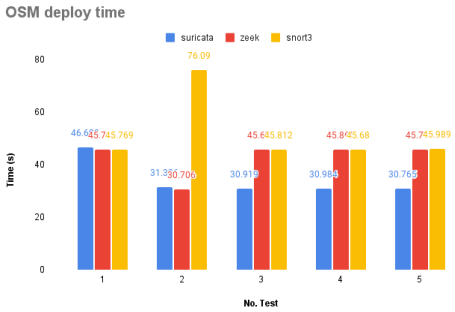
*รูปที่ 9: หน้าต่างแสดงผลวิเคราะห์ผลคุกคาม*

## 5.3 การวัดประสิทธิภาพ

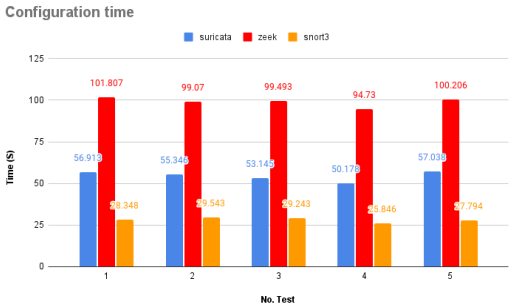
สำหรับการวัดประสิทธิภาพของระบบจะใช้เกณฑ์ในการพิจารณาด้วยกัน 3 ด้าน โดยแนวทางการทดสอบได้ค่าทรัพยากรของเครื่อง VM ให้เหมือนกันคือ CPU จำนวน 4 core และ RAM จำนวน 4 GB

- *Deployment time:* เวลาที่ใช้ในการจำลองเน็ตเวิร์กเซอร์วิสด้านความปลอดภัยแต่ละประเภท (ในที่นี้คือ Suricata (DPI), Snort (IDS/IPS), Zeek (traffic analyzer)) โดยค่าเฉลี่ยที่วัดได้แสดงดังรูปที่ 10

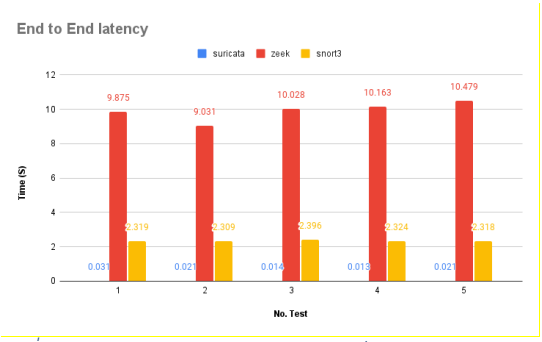
- *Configuration time:* เวลาที่ใช้ในการตั้งค่าข้อมูลเบื้องต้นในแต่ละเน็ตเวิร์กเซอร์วิสที่ได้ถูกจำลองขึ้น โดยค่าเฉลี่ยเวลาที่วัดได้แสดงดังรูปที่ 11
- *End*-to-end latency: เวลาแฝงโดยรวมทั้งหมดจากต้นทางถึงปลายทาง โดยจะนับจากเวลาที่ได้รับ event logs ทำการประมวลผล และส่ง alert log จากตัวเน็ตเวิร์กเซอร์วิส (NS instances) ไปแสดงผลที่ Wazuh dashboard โดยผลที่วัดได้แสดงดังรูปที่ 12




***รูปที่ 5:*** *กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในการติดตั้งเน็ตเวิร์คเซอร์วิสด้านความปลอดภัยแต่ละประเภท*




***รูปที่ 6:*** *กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในการตั้งค่าเบื้องต้นในแต่ละเน็ตเวิร์กเซอร์วิส (**หมายเหตุ: (1) สาเหตุที่ Zeek ใช้เวลานานนั้นเกิดจากการที่ต้องมีการเปิดใช้งานโดยใช้คำสั่ง sudo zeekctl deploy; (2) สาเหตุที่ Suricata ใช้เวลานานกว่า Snort เกิดจากการใช้คำสั่ง sudo systemctl restart suricata.service ส่งผลให้ใช้เวลาเพิ่มขึ้นในส่วนนี้)*




***รูปที่ 7:*** *กราฟแสดงข้อมูลเปรียบเทียบระยะเวลาแฝงเฉลี่ย (end-to-end latency) ของแต่ละเน็ตเวิร์คเซอร์วิส (**หมายเหตุ:จากผลการทดลองพบข้อสันนิฐานว่าสาเหตุที่ Suricata มีการส่งที่รวดเร็วเป็นเพราะมีการรองรับ JSON log ไฟล์อยู่แล้ว ซึ่งแตกต่างจาก Zeek และ Snort ที่จะต้องมีการแปลงข้อมูลหรือเขียน log ให้อยู่ในรูป JSON)*

## 5. การถ่ายทอดองค์ความรู้และขยายผลนวัตกรรม

ในหัวข้อนี้จะพูดถึงแนวทางการถ่ายทอดองค์ความรู้ ขยายผลนวัตกรรมสู่การใช้งานจริง โดยทางทีมวิจัยได้กำหนดแนวทางการดำเนินงานเอาไว้ ดังนี้ การเข้าร่วมและจัดกิจกรรมสัมมนาเชิงวิชาการ การออกบูธจัดแสดงนิทรรศการ และ การสร้างความร่วมมือกับหน่วยงานที่เกี่ยวข้อง

## 6. บทสรุป

โครงการวิจัยนี้มุ่งเน้นการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการเน็ตเวิร์คเซอร์วิสเสมือนจริง (VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเครือข่ายเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต ด้วยการ

ประยุกต์ใช้เทคโนโลยีหลัก NFV, SDN และ Cloud เพื่อสร้างระบบนิเวศที่สามารถบริหารจัดการทรัพยากรและเน็ตเวิร์คเซอร์วิสได้อย่างหยืดหยุ่น อัตโนมัติ และมีประสิทธิภาพ โดยไม่จำเป็นต้องพึ่งพาทรัพยากรโครงสร้างพื้นฐานที่มีต้นทุนสูงหรือความรู้เฉพาะทางจากผู้ใช้งาน โดยแนวคิดการออกแบบสถาปัตยกรรมระบบ จะแบ่งการทำงานออกเป็น 3 ระดับชั้น ดังนี้ (1) System plane ซึ่งเป็นส่วนที่ทำหน้าที่เชื่อมต่อแพลตฟอร์มฯ กับเครือข่ายผู้รับบริการ เพื่อทำหน้าที่รวบรวมข้อมูล logs การทำงานต่างๆ ที่สนใจ และส่งต่อไปยังแพลตฟอร์มฯ สำหรับการตรวจสอบ; (2) Orchestration and enforcement plane เป็นแกนหลักของระบบ ทำหน้าที่ควบคุม บริหารจัดการ และประสานการทำงานต่างๆ ภายในแพลตฟอร์ม ทั้งการจัดสรรทรัพยากร การจำลองเน็ตเวิร์คเซอร์วิสและการตรวจสอบความเสี่ยงด้านความปลอดภัยจากข้อมูล logs; และ (3) User plane เป็นจุดเชื่อมต่อระหว่าง กับผู้ใช้งาน โดยอำนวยให้แก่ผู้ดูแลระบบสามารถสร้างและออกแบบพิมพ์เขียว (network service descriptor) ของแต่ละเน็ตเวิร์คเซอร์วิสได้ตามที่ต้องการ รวมไปถึงการเรียกดูสถานะๆ ต่างๆ ที่เกิดขึ้นภายในระบบ รวมไปถึงผู้ใช้บริการสามารถเข้าถึงข้อมูลการวิเคราะห์ผลภัยคุกคาม และผลการตรวจอสอบข้อมูลผ่าน dashboard ที่ใช้งานง่าย โดยเป้าหมายปลายทางของทีมวิจัยมุ่งผลสัมฤทธิ์ 3 ด้านหลัก ได้แก่ (1) *ด้านต้นทุน:* โดยการช่วยลดภาระค่าใช้จ่ายในการลงทุนและดำเนินการด้านความปลอดภัยขององค์กร; (2) *ด้านการรักษาความปลอดภัย:* ส่งนวัตกรรมที่สามารถบริหารจัดการทรัพยากรและเน็ตเวิร์คเซอร์วิสได้อย่างยืดหยุ่น อัตโนมัติ และมีประสิทธิภาพ; และ (3) *ด้านประสิทธิภาพ:* เพิ่มความรวดเร็วและยืดหยุ่นในการบริหารจัดการด้วยแนวทางแบบรวมศูนย์และเวิร์กโฟลว์อัตโนมัติ แนวทางนี้จะช่วยส่งเสริมให้องค์กรต่างๆ สามารถเข้าถึงบริการด้านความมั่นคงปลอดภัยเครือข่ายที่มีคุณภาพได้อย่างทั่วถึง และมีศักยภาพในการรับมือกับความเสี่ยงทางไซเบอร์ได้ดียิ่งขึ้น

## 6. กิตติมากรรมประกาศ

รายงานฉบับนี้ได้รับทุนอุดหนุนวิจัย จากเงินกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ (กทปส.) สำนักงานกสทช. สัญญารับทุนเลขที่ B 64-3-(2)-004

## บรรณานุกรม

[1] AWS Implementation Guide: “Workload Discovery on AWS Implementation Guide”. Accessed on October 16. Available URL: https://docs.aws.amazon.com/pdfs/solutions/latest/workload-discovery-on-aws/workload-discovery-on-aws.pdf#architecture-overview

[2] AWS Whitepaper 2023: "ETSI NFVO Compliant Orchestration in the Kubernetes/Cloud Native World." Accessed on October 16. Available URL: https://docs.aws.amazon.com/pdfs/whitepapers/latest/ETSI-NFVO-compliant-orchestration-in-kubernetes/ETSI-NFVO-compliant-orchestration-in-kubernetes.pdf

[3] Google Cloud products. (2023). Accessed on October 20. Available URL: https://cloud.google.com/products#section-1

[4] Google Cloud Architecture Framework. (2023). Accessed on October 20. Available URL: https://cloud.google.com/architecture/framework

[5] Microsoft Azure. Accessed on October 21. Available URL: https://www.isaca.org/resources/isaca-journal/issues/2020/volume-5/auditing-the-cloud-microsoft-azure

[6] Vmware-Cloud. Accessed on October 22. Available URL: https://www.vmware.com/cloud-solutions/vmware-cloud.html#resources

[7] OpenStack Foundation. (2023). OpenStack Deployment and Operations Guide. Release 2020.2. Accessed on October 13. Available URL: https://pure-storage-openstack-docs.readthedocs.io/_/downloads/en/victoria/pdf/

[8] OpenStack Overview. (2023). OpenStack - A Modular Collection of Cloud Services. Accessed on October 13. Available URL: https://pure-storage-openstack-docs.readthedocs.io/en/zed_a/openstack-overview/section_modular-collection.html

[9] The Snort Project. (2020, April 8). Snort Users Manual (Version 2.9.16). [Online]. Available: https://snortorgite.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf. [Accessed: Mar 29, 2023].

[10] Gavrilović, N., Ćirić, V., & Lozo, N. (2022). Snort IDS System Visualization Interface for Alert Analysis. Serbian Journal of Electrical Engineering, 19(1), 67-78. DOI: https://doi.org/10.2298/SJEE2201067G

[11] Wandhare, G. M., Gujar, P. S. N., & Thakare, V. M. (2015). Research article: Network intrusion detection technique for regular expression detection using DPI in ad-hoc.

[12] OISF, "Suricata User Guide Release 5.0.3." [Online]. Available: https://suricata.readthedocs.io/_/downloads/en/suricata-5.0.3/pdf/. [Accessed: April 7, 2023].

[13] Fadhilah, D., & Marzuki, M. I. (2020). Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against DoS/DDoS Attacks. In Authorized licensed use limited to: National Science & Technology Development Agency. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 78-1-7281-7450).

[14] Provos, N. (2023). Honeypot Background. [Online]. Available: https://www.honeyd.org/background/. [Accessed: November 22, 2023].

[15] Provos, N. (2023). Honeypot concepts. [Online]. Available: https://www.honeyd.org/concepts/. [Accessed: December 22, 2023].

[16] González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. Sensors, 21(14), 4759. https://doi.org/10.3390/s21144759

[17] Wazuh Inc. (2023). Wazuh platform overview. [Online]. Available: https://wazuh.com/platform/overview/ [Accessed: December 16, 2023].

[18] Zeek, 2022. “The zeek network security monitor”. [Online]. Available: https://zeek.org/. [Accessed: April 25, 2023].

[19] The Cyber Center for Security and Analytics. (2020). Zeek Intrusion Detection Series: Lab 1 - Introduction to the Capabilities of Zeek. Document Version: 03-13-2020.

[20] Open source MANO project, Sep 2021. Available URL: https://osm.etsi.org/

[21] An ETSI OSM Community White Paper. (2017). Available URL: https://osm.etsi.org/images/OSM-Whitepaper-TechContentReleaseTHREE-FINAL.PDF

[22] ETSI OSM Community. (2017). "OSM RELEASE THREE A TECHNICAL OVERVIEW." An ETSI OSM Community White Paper. October 2017. Accessed on October 5, 2023 Available at: https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseTHREE-FINAL.PDF

[23] OSM. (n.d.). "OSM Scope and Functionality." Accessed on October 6, 2023. Available at: https://osm.etsi.org/wikipub/index.php/OSM_Scope_and_Functionality#SDN_Controllers/WIMs_as_managers_of_Software-Defined_Network_Platforms

[24] OpenStack Tacker, 2019. [Online]. Available: https://docs.openstack.org/tacker/latest/. [Accessed: Dec 5, 2023].

[25] OpenStack Foundation. "Accelerating NFV Delivery with OpenStack Global Telecoms Align Around Open Source Networking Future" [Online]. Available at: https://object-storage-ca-ymq-1.vexxhost.net/swift/v1/6e4619c416ff4bd19e1c087f27a43eea/www-assets-prod/marketing/OpenStack-NFV-Print.pdf. [Accessed: Jan 5, 2024].

# ภาคผนวก ซ.

## การจัดซื้อครุภัณฑ์ (คอมพิวเตอร์ส่วนบุคคล และจอแสดงผล)

| | | | |
|---|---|---|---|
| NECTEC a member of NSTDA | **ชื่อเอกสาร** | ข้อกำหนดรายละเอียดคุณลักษณะเฉพาะของพัสดุ (Terms of Reference :TOR): จัดหาเครื่องคอมพิวเตอร์ส่วนบุคคลแบบพกพา จำนวน 2 เครื่อง และจอภาพแสดงผลสาธิตงาน จำนวน 1 จอ | |
| | **ผู้จัดทำ/หน่วยงาน** | นายขจรศักดิ์ ปิยังกร / SEC | **หน้า** 1 of 4 |

**ข้อกำหนดรายละเอียดคุณลักษณะเฉพาะของพัสดุ (Terms of Reference :TOR)**
**จัดหาเครื่องคอมพิวเตอร์ส่วนบุคคลแบบพกพา จำนวน 2 เครื่อง และจอภาพแสดงผลสาธิตงาน จำนวน 1 จอ**

**1. หลักการและเหตุผล**

สถานการณ์โลกในปัจจุบันมีการเปลี่ยนแปลงอย่างรวดเร็วและสลับซับซ้อนมากกว่าในอดีต ผลกระทบที่กำลังเกิดขึ้นจากการเคลื่อนตัวสู่ยุคอุตสาหกรรม 4.0 ได้นำมาซึ่งการเปลี่ยนแปลงในทุกมิติ ทั้งด้านการพัฒนาโครงสร้างพื้นฐาน ICT ด้านธุรกิจและการบริการ ผู้บริโภคส่วนใหญ่ต่างคาดหวังว่าโครงสร้างพื้นฐาน ICT และระบบนิเวศดิจิทัลจะสามารถรองรับการขยายตัวที่เพียงพอต่อความต้องการขององค์กรและปริมาณการรับส่งข้อมูลที่ไหลเข้ามาอย่างต่อเนื่องและมหาศาลได้ อีกทั้งความคาดหวังว่าการพึ่งพาเทคโนโลยีนั้นจะสามารถยกระดับประสิทธิภาพการประมวลผลและการดำเนินงาน และมีศักยภาพมากพอในการจัดสรรหรือขยายทรัพยากรเน็ตเวิร์คได้อย่างยืดหยุ่นและอิสระ

เพื่อให้สอดคล้องกับความต้องการที่หลากหลายของกลุ่มผู้ใช้งานในแต่ละภาคธุรกิจได้ โครงการพัฒนาแพลตฟอร์มสำหรับบริหารและจัดการฟังก์ชั่นเครือข่ายเสมือนจริง (VNFs as a Service: VNFaaS) เพื่อเตรียมพร้อมสำหรับสร้างระบบนิเวศเสมือนจริงสนับสนุนเครือข่าย 5G ในอนาคต จึงถูกพัฒนาขึ้น โดยมีจุดประสงค์เพื่อพัฒนาองค์ความรู้และให้บริการเกี่ยวกับกลุ่มเทคโนโลยีการจำลองระบบเสมือนจริง (virtualization technology) ได้แก่ (1) การทำงานเสมือนจริงเครือข่าย (Network Functions Virtualization: NFV) (2) การจัดการเครือข่ายด้วยซอฟต์แวร์ (Software-Defined Networking: SDN) และ (3) การจัดเก็บและการประมวลผลข้อมูลบนคลาวด์ (Cloud computing) กลายเป็นกลุ่มเทคโนโลยีที่ได้เข้ามามีบทบาทสำคัญในการขับเคลื่อนโครงสร้างพื้นฐานระบบเครือข่ายและการให้บริการเน็ตเวิร์คเซอร์วิสแบบดั้งเดิม (traditional networking model) ไปสู่โครงสร้างเครือข่ายยุคใหม่ที่กำหนดโดยซอฟต์แวร์ (Software-defined infrastructure) เพื่อรองรับสภาพแวดล้อม 5G ในอนาคตในทุกมิติของภาคอุตสาหกรรม (5G services) รวมถึง Internet of Things (IoTs), smart city, smart factory เป็นต้น

โครงการฯ มีความจำเป็นต้องจัดหาเครื่องคอมพิวเตอร์ส่วนบุคคลแบบพกพา ที่มีประสิทธิภาพสูงรองรับการทำงานกับปริมาณข้อมูลขนาดใหญ่ และมีความรวดเร็วในการประมวลผล ซึ่งต้องมี CPU, RAM, Disk Space เพียงพอต่อการประมวลผลเหมาะสมกับการนำไปใช้เพื่อการวิจัยพัฒนา ออกแบบ และทดสอบระบบการทำงาน ระหว่างการพัฒนาทั้งในและนอกสถานที่สามารถจำลองสภาพแวดล้อมการทำงานที่มีผู้ใช้งานพร้อมกันและต้องการทรัพยากรการคำนวณเป็นจำนวนมากได้ รวมไปถึงจอแสดงผลสาธิตงาน เพื่อการประชาสัมพันธ์นำเสนอผลงานหรือถ่ายทอดความรู้เชิงวิจัยนวัตกรรม และจัดกิจกรรมสัมมนาแนะนำแพลตฟอร์ม

**2. วัตถุประสงค์**

1. ใช้เป็นเครื่องมือในการออกแบบ พัฒนาเทคโนโลยีงานวิจัยขั้นสูง งานวิเคราะห์ข้อมูลขนาดใหญ่ งานทดสอบระบบ รองรับการประมวลผลได้อย่างรวดเร็ว จำลองระบบงานในการพัฒนาต่าง ๆ ที่มีข้อมูลในระบบจำนวนมากและเพื่อใช้นำเสนอระบบงานให้หน่วยงานต่าง ๆ
2. เพื่อให้บุคลากรสามารถปฏิบัติหน้าที่ได้อย่างต่อเนื่องมีประสิทธิภาพและได้ผลลัพธ์ตามข้อกำหนดโครงการ

**3. รายละเอียดคุณลักษณะเฉพาะ**

3.1 เครื่องคอมพิวเตอร์ชนิดพกพา ระบบปฏิบัติการ macOS จำนวน 1 เครื่อง

3.1.1 หน่วยประมวลผล (Central Processing Unit) มีแกนประมวลผลคอร์ (Core) จำนวนไม่น้อยกว่า 8 คอร์ ซึ่งมีคอร์ด้านประสิทธิภาพ (high-performance) ไม่น้อยกว่า 4 คอร์

3.1.2 มีหน่วยความจำ (Memory) ชนิด LPDDR5 ทำงานที่ความเร็ว 6400 MHz ขึ้นไป ขนาดรวมกันไม่น้อยกว่า 16 GB ติดตั้งไว้ในตัวเครื่องพร้อมใช้งาน

3.1.3 มีหน่วยจัดเก็บข้อมูลรูปแบบ SSD (Solid State Drive) ขนาดความจุ 512 GB หรือมากกว่า จำนวน 1 หน่วย ติดตั้งภายในตัวเครื่องพร้อมใช้งาน

3.1.4 มีพัดลมระบายความร้อนในตัวเครื่องอย่างน้อย 1 ตัว ติดตั้งไว้ภายในตัวเครื่องพร้อมใช้งาน

3.1.5 มีหน่วยประมวลผลกราฟิก (Graphic Processing Unit) ทำงานที่ความเร็ว 1300 MHz ขึ้นไป มีแบนด์วิดท์หน่วยความจำ 100GB/s ขึ้นไป ติดตั้งไว้ภายในตัวเครื่องพร้อมใช้งาน

3.1.6 มีหน่วยแสดงผลภาพ (Display) ขนาด 13 นิ้วขึ้นไป โดยวัดจากเส้นทแยงมุม สามารถแสดงผลที่ความละเอียดไม่น้อยกว่า 2560 x 1600 pixels ความถี่ (Refresh rate) 120 Hz ขึ้นไป พร้อมเทคโนโลยี IPS

3.1.7 มี Built-in Camera ขนาดเล็กติดตั้งมากับตัวเครื่อง ติดตั้งไว้พร้อมใช้งานบริเวณจอภาพ ความละเอียดไม่ต่ำกว่า 720p

3.1.8 มีเทคโนโลยีไร้สาย Wi-Fi 6 มาตรฐาน 802.11ax ติดตั้งในตัวเครื่องพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.1.9 มีเทคโนโลยีไร้สาย Bluetooth 5.0 ขึ้นไป ติดตั้งในตัวเครื่องพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.1.10 มีพอร์ต Thunderbolt อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.1.11 มีพอร์ต USB-C อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.1.12 มีระบบเสียง ช่องต่อหูฟัง 3.5 มม. และลำโพงติดตั้งพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.1.13 มี AC Adapter แปลงไฟแบบ USB-C ขนาด 65W หรือสูงกว่า จำนวน 1 หน่วย

3.1.14 มีสายอุปกรณ์ต่อพ่วงสำหรับชาร์จไฟ จำนวน 1 หน่วย

3.2 เครื่องคอมพิวเตอร์ชนิดพกพา ระบบปฏิบัติการ Window 10, 11 รูปแบบ 64 bit จำนวน 1 เครื่อง

3.2.1 หน่วยประมวลผล (Central Processing Unit) Intel Processor มีแกนประมวลผลคอร์ (Core) จำนวนไม่น้อยกว่า 24 คอร์ ซึ่งมีคอร์ด้านประสิทธิภาพ (high-performance) ไม่น้อยกว่า 8 คอร์

3.2.2 มีหน่วยความจำ (Memory) ชนิด DDR5 ทำงานที่ความเร็ว 4800 MHz ขึ้นไป ขนาดรวมกันไม่น้อยกว่า 16 GB ติดตั้งไว้ในตัวเครื่องพร้อมใช้งาน

3.2.3 มีหน่วยจัดเก็บข้อมูลรูปแบบ PCIe/NVMe M.2 SSD ขนาดความจุ 512 GB หรือมากกว่า จำนวน 1 หน่วย ติดตั้งภายในตัวเครื่องพร้อมใช้งาน

3.2.4 มีหน่วยประมวลผลกราฟิก (Graphic Processing Unit) Nvidia ทำงานที่ความเร็ว 2300 MHz ขึ้นไป มีแบนด์วิดท์หน่วยความจำ 192GB/s ขึ้นไป ติดตั้งไว้ภายในตัวเครื่องพร้อมใช้งาน

3.2.5 มีหน่วยแสดงผลภาพ (Display) ขนาด 16 นิ้วขึ้นไป โดยวัดจากเส้นทแยงมุม สามารถแสดงผลที่ความละเอียดไม่น้อยกว่า 1920 x 1200 pixels ความถี่ (Refresh rate) 165 Hz ขึ้นไป พร้อมเทคโนโลยี IPS

3.2.6 มี Built-in Camera ขนาดเล็กติดตั้งมากับตัวเครื่อง ติดตั้งไว้พร้อมใช้งานบริเวณจอภาพ ความละเอียดไม่ต่ำกว่า 720p

3.2.7 มีเทคโนโลยีไร้สาย Wi-Fi 6E มาตรฐาน 802.11ax ติดตั้งในตัวเครื่องพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.2.8 มีเทคโนโลยีไร้สาย Bluetooth 5.2 ขึ้นไป ติดตั้งในตัวเครื่องพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.2.9 มีพอร์ต LAN (10/100/1000 Mbps) รูปแบบ RJ45 จำนวน 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.2.10 มีพอร์ต HDMI อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.2.11 มีพอร์ต USB 3.2 Gen 2 Type-A และ C อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.2.12 มีระบบเสียง ช่องต่อหูฟัง 3.5 มม. และลำโพงติดตั้งพร้อมใช้งานโดยไม่ต้องใช้อุปกรณ์เสริม

3.2.13 มี AC Adapter แปลงไฟ ขนาด 280W หรือสูงกว่า จำนวน 1 หน่วย

3.2.14 มีสายอุปกรณ์ต่อพ่วงสำหรับชาร์จไฟ จำนวน 1 หน่วย

3.3 จอภาพแสดงผลสาธิตงาน จำนวน 1 จอ

3.3.1 ขนาดจอ (Screen Size) 34 นิ้ว ขึ้นไป

3.3.2 รูปแบบจอ (Panel Type) ชนิด VA (Vertical Alignment)

3.3.3 ความละเอียดจอ (Resolution) 3440 x 1440 ขึ้นไป

3.3.4 อัตราความถี่ (Refresh rate) 75 Hz ขึ้นไป

3.3.5 ความเร็วในการตอบสนอง (Response Time) มีค่าไม่สูงกว่า 4 ms

3.3.6 มีพอร์ต HDMI อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

3.3.7 มีพอร์ต DisplayPort อย่างน้อย 1 พอร์ต ติดตั้งในตัวเครื่องพร้อมใช้งาน

**4. สถานที่ส่งมอบ**

ผู้รับจ้างต้องส่งมอบรายการตามข้อ 3 ของข้อกำหนดคุณลักษณะเฉพาะ ณ ห้อง 520 ชั้น 5 ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ 112 อุทยานวิทยาศาสตร์ประเทศไทย ถนนพหลโยธิน ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี 12120 ภายใน 60 วัน นับถัดจากวันลงนามในสัญญา

**5. วงเงินในการจัดหา**

ศูนย์ฯ กำหนดวงเงินในการจัดหาเครื่องคอมพิวเตอร์ส่วนบุคคลแบบพกพา จำนวน 2 เครื่อง และจอภาพแสดงผลสาธิตงาน จำนวน 1 จอ รวมเป็นเงินทั้งสิ้น 130,000 บาท (หนึ่งแสนสามหมื่นบาทถ้วน) ซึ่งรวมภาษีอากร และค่าใช้จ่ายทั้งปวงไว้เรียบร้อยแล้ว

**6. เงื่อนไขการชำระเงิน**

ศูนย์ฯ จะชำระเงินให้กับผู้รับจ้างเพียงงวดเงินเดียว ทั้งนี้จะชำระเงินภายใน 30 วัน หลังจากคณะกรรมการตรวจรับพัสดุ ดำเนินการตรวจรับในแต่ละงวดเรียบร้อยแล้ว

**7. ผู้ติดต่อประสานงาน**


นายขจรศักดิ์ ปิยังกร
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
112 อุทยานวิทยาศาสตร์ประเทศไทย ถนนพหลโยธิน
ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี 12120
โทรศัพท์ 0 2564 6900 ต่อ 2579
อีเมล kajornsak.piyoungkorn@nectec.or.th