รายงานฉบับสมบูรณ์

โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
ประเภทที่ ๑/๒๕๕๘ ประจำงวดที่ ๕

โครงการระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายบนโทรศัพท์เคลื่อนที่

ผศ.ดร. กิ่งกาญจน์ สุขคณาภิบาล
(ผู้รับผิดชอบโครงการ)

ได้รับทุนอุดหนุนจาก
กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

รายงานฉบับสมบูรณ์		
ชื่อโครงการ	โครงการระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายบนโทรศัพท์เคลื่อนที่	
ข้อมูลโครงการ	รหัสอ้างอิง: T2-1-0004/57	
	วันเริ่มต้นโครงการ: 22 กันยายน 2558	
	วันสิ้นสุดโครงการ: 21 กันยายน 2560	
หน่วยงานผู้รับทุน	มหาวิทยาลัยกรุงเทพ	วิทยาเขตกล้วยน้ำไท เลขที่ 119 ถนนพระราม 4 แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10110 โทรศัพท์ 02-350-3500 ต่อ 1693 ติดต่อ **ผศ.ดร.กิ่งกาญจน์ สุขคณาภิบาล** (หัวหน้าโครงการ)
คณะผู้ดำเนินโครงการ	ผู้ช่วยศาสตราจารย์ ดร. กิ่งกาญจน์ สุขคณาภิบาล (มหาวิทยาลัยกรุงเทพ) รองศาสตรจารย์ ดร.ประสงค์ ปราณีตพลกรัง (มหาวิทยาลัยศรีปทุม) ผู้ช่วยศาสตราจารย์ ดร. มัชฌิกา อ่องแตง (มหาวิทยาลัยรุกิจบัณฑิตย์) ผู้ช่วยศาสตราจารย์ ดร. วรวัฒน์ เชิญสวัสดิ์ (มหาวิทยาลัยกรุงเทพ)	
ที่ปรึกษาโครงการ	ศาสตรจารย์ ดร. ชิดชนก เหลือสินทรัพย์ (จุฬาลงกรณ์มหาวิทยาลัย)	
ข้อมูลรายงานที่ส่ง	วันที่ส่งรายงานฉบับสมบูรณ์ (งานงวดห้า): 21 กันยายน 2560	
	จัดทำรายงานและนำเสนอโดย: มหาวิทยาลัยกรุงเทพ	

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

บทที่ 1
บทนำ

1.1 หลักการและเหตุผลความจำเป็น

ในปี พ.ศ. 2558 กสทช. รายงานตัวเลขการลงทะเบียนเปิดใช้เบอร์โทรศัพท์เคลื่อนที่ในประเทศไทยมีถึง 97.6 ล้านหมายเลข คิดเป็นร้อยละ 145.64 ต่อประชากร แสดงถึงพฤติกรรมของผู้บริโภคที่ใช้โทรศัพท์เคลื่อนที่เสมือนปัจจัยหนึ่งในชีวิตประจำวัน ข้อมูลจากธนาคารแห่งประเทศไทยยังระบุว่ามูลค่าของการชำระเงินทางอิเล็กทรอนิกส์ (e-Payment) มีถึง 743 ล้านล้านบาท และมูลค่าการค้าอิเล็กทรอนิกส์ (e-Commerce) มีถึง 744 ล้านบาท

จากการศึกษาและสำรวจพบว่าร้อยละ 70 ของมัลแวร์บนโทรศัพท์เคลื่อนที่ (Mobile malware) มุ่งขโมยข้อมูลส่วนตัว (Felt et al.,2011) เช่น ข้อมูลเกี่ยวกับบัตรเครดิตและบัญชีธนาคาร ล็อคอินและพาสเวิร์ดเป็นต้น มัลแวร์ในปัจจุบันมีรูปแบบที่หลากหลายและปรับเปลี่ยนเพื่อหลีกเลี่ยงการตรวจจับจากโปรแกรมแอนติไวรัสที่มีอยู่ในท้องตลาด เหตุการณ์ที่สำคัญ เช่น มัลแวร์ ชื่อว่า "Android.Dropdialer" บนกูเกิ้ลเพลย์สโตว์ (Google Play Store) นั้น แอปพลิเคชั่นที่มีมัลแวร์นี้ฝังตัวอยู่จะมีลักษณะของโค้ดและมีพฤติกรรมไม่แตกต่างจากแอปพลิเคชั่นปกติทั่ว ๆ ไปแต่อย่างใด แต่หลังจากที่ผู้ใช้ทำการติดตั้งแอปพลิเคชั่นนี้บนอุปกรณ์แอปพลิเคชั่นจะดาวน์โหลดฟังก์ชั่นอันตรายมาที่เครื่องอย่างอัตโนมัติ เหตุการณ์นี้มีผู้เสียหายจำนวนมากเพราะผู้ใช้โดยส่วนใหญ่ให้ความไว้ใจในเพลย์สโตว์ (Play Store) ว่าไม่มีแอปพลิเคชั่นอันตราย

ฟังก์ชั่นดาวน์โหลดอัตโนมัติเป็นฟังก์ชั่นที่มีในแอปพลิเคชั่นปกติใช้สำหรับอัพเดตเวอร์ชั่นเพื่อเพิ่มประสิทธิภาพการใช้งาน เพิ่มฟังก์ชั่นการใช้งานและแก้ไขจุดบกพร่องในโค้ด แอปพลิเคชั่นที่ประกอบด้วยฟังก์ชั่นดาวโหลดจึงไม่ถูกจัดว่าเป็นแอปพลิเคชั่นอันตราย การใช้โปรแกรมแอนติไวรัสที่มีอยู่ในท้องตลาดจึงไม่เพียงพอต่อการป้องกันมัลแวร์ประเภทที่สามารถดาวโหลดอัตโนมัติได้ การแยะแยะแอปพลิเคชั่นที่มีพฤติกรรมน่าสงสัยจากแอปพลิเคชั่นปกติทั่วไปจึงสามารถทำได้โดยใช้การเฝ้าระวังและตรวจสอบขณะที่แอปพลิเคชั่นทำงานบนอุปกรณ์ เพื่อตรวจจับพฤติกรรมของแอปพลิเคชั่นที่เปลี่ยนแปลงจากเดิมอย่างมีนัยสำคัญและแจ้งให้ผู้ใช้งานทราบถึงภัยที่กำลังคุกคาม

โครงการนี้จึงมุ่งพัฒนาระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายบนโทรศัพท์เคลื่อนที่เพื่อปกป้องข้อมูลส่วนบุคคลและข้อมูลแสดงตัวตนจากภัยคุกคาม และการพัฒนาระบบนี้เป็นการเพิ่มขีดความสามารถของประเทศในการป้องกันและรับมือภัยคุกคามด้านไซเบอร์อย่างมีประสิทธิภาพ โดยร า ย ง า น

เบื้องต้นฉบับนี้ได้นำเสนอรายละเอียดการทำงานในลำดับต่อไป อันประกอบด้วยกรอบแนวคิดในการดำเนินงาน วิธีการ และแผนการทำงานสำหรับโครงการนี้

1.2 วัตถุประสงค์ของโครงการ

โครงการวิจัยนี้มุ่งเน้นการตรวจจับมัลแวร์ที่ฝังตัวกับแอปพลิเคชั่นปกติ ซึ่งจะไม่แสดงพฤติกรรมที่น่าสงสัยและเป็นอันตรายในตอนต้น และเพิ่มประสิทธิภาพของการตรวจจับโค้ดอันตรายบนอุปกรณ์เคลื่อนที่ ซึ่งมีความสอดคล้องกับนโยบายภาครัฐและแผน ดังนี้

1) แผนแม่บทกิจการโทรคมนาคม ฉบับที่ 1 พ.ศ. 2555 - 2559
2) แผนยุทธศาสตร์การวิจัยและพัฒนาเกี่ยวกับการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ ฉบับที่ 1 พ.ศ. 2556 - 2560

วัตถุประสงค์ของโครงการมีดังนี้

1) เพื่อวิจัยและพัฒนาอัลกอริทึมที่ใช้ในการตรวจจับแอปพลิเคชั่นอันตราย
2) เพื่อออกแบบและพัฒนาระบบการเตือนภัยผู้ใช้ให้ตระหนักถึงระดับอันตรายของแอปพลิเคชั่นที่มีอยู่บนอุปกรณ์เคลื่อนที่
3) เพื่อทำการป้องกันแอปพลิเคชั่นอันตรายและแก้ไขระบบปฏิบัติการให้สามารถทำงานได้ตามปกติหรือฟื้นสภาพได้

1.3 ขอบเขตของโครงการ

ขอบเขตการดำเนินงานที่สำคัญของโครงการ มีดังนี้

1) การพัฒนาและออกแบบระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายบนอุปกรณ์เคลื่อนที่โดยเน้นอุปกรณ์เคลื่อนที่ที่ใช้ระบบปฏิบัติการแอนดรอยด์ ด้วยเหตุผล 2 ข้อดังนี้
    ก) จำนวนผู้ใช้และอัตราเติบโตของผู้ใช้อุปกรณ์เคลื่อนที่ด้วยระบบปฏิบัติการแอนดรอยด์สูงกว่าระบบปฏิบัติการอื่น ๆ เช่น iOS, windows 8 เป็นต้น ตั้งแต่ปี พ.ศ. 2554 จนถึงปัจจุบัน

ข) มัลแวร์ที่พบบนระบบปฏิบัติการแอนดรอยด์มีจำนวนมากกว่าและมีแนวโน้มเพิ่มขึ้นสูงกว่าระบบปฏิบัติการอื่น ๆ ในอนาคต

2) โครงสร้างของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นมีลักษณะของการทำงานแบบแม่ข่ายลูกข่าย (Client-server architecture) โดยที่ฝั่งรับหมายถึงอุปกรณ์เคลื่อนที่ ฝั่งส่งหมายถึงเซิร์ฟเวอร์

ก) ฝั่งอุปกรณ์เคลื่อนที่ (Client-side) ทำหน้าที่เฝ้าระวังและตรวจจับพฤติกรรมของแอปพลิเคชั่นที่ติดตั้งและประมวลผลบนอุปกรณ์แล้ว แล้วเตือนภัยผู้ใช้ว่ากำลังถูกคุกคามเมื่อแอปพลิเคชั่นมีพฤติกรรมที่เปลี่ยนไปในเชิงน่าสงสัยหรืออันตราย

ข) ฝั่งเซิร์ฟเวอร์ (Server-side) ทำหน้าที่เป็นฐานข้อมูลของแอปพลิเคชั่นและรูปแบบพฤติกรรมน่าสงสัยขณะทำงาน เพื่อเตือนผู้ใช้ถึงระดับอันตรายของแอปพลิเคชั่นก่อนการติดตั้ง

3) องค์ประกอบของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นมีดังนี้

ก) ส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI) แสดงข้อมูลของการเฝ้าระวังและเตือนภัยและอนุญาตให้ผู้ใช้สามารถกำหนดค่าพารามิเตอร์ของการใช้งานได้ ได้แก่ การตั้งเวลาเปิด-ปิดระบบเฝ้าระวัง เป็นต้น

ข) ส่วนจัดการการเตือนภัย (Alert handler) ส่งสัญญาณเตือนภัยและรอรับการตอบกลับจากผู้ใช้ว่าจะจัดการอย่างไร

ค) ส่วนเฝ้าระวังและตรวจจับ (Monitoring and detecting module) เป็นส่วนหลักของระบบทำหน้าที่แยกแยะแอปพลิเคชั่นอันตรายจากแอปพลิเคชั่นปกติ โดยเฝ้าดูพฤติกรรมของแต่ละแอปพลิเคชั่นในการรับส่งข้อมูลบนอินเตอร์เน็ต

ง) ส่วนเชื่อมต่อกับเซิร์ฟเวอร์ (Client-server connector) เป็นตัวกลางเชื่อมต่อและจัดการเครือข่ายเพื่อเชื่อมต่อ

4) การประเมินประสิทธิภาพของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย มีดังนี้

ก) ด้านการใช้หน่วยความจำที่เพิ่มขึ้น (Memory consumption)

ข) ด้านการใช้ซีพียูในการประมวลผลที่เพิ่มขึ้น (CPU consumption)

5) การจัดทำเว็บไซต์เพื่อประชาสัมพันธ์และเผยแพร่ระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย ประกอบด้วยข้อมูลต่อไปนี้

ก) คู่มือการใช้งานและคำแนะนำระบบ

ข) ข่าวสารเกี่ยวกับมัลแวร์บนอุปกรณ์เคลื่อนที่

ค) รายชื่อมัลแวร์บนอุปกรณ์เคลื่อนที่ที่พบจนถึงปัจจุบัน

ง) แบบประเมินความพึงพอใจของผู้ใช้งาน

จ) แบบฟอร์มแจ้งปัญหาการใช้งานระบบ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

กลุ่มเป้าหมายของโครงการนี้มุ่งเน้นผู้บริโภคที่ใช้อุปกรณ์เคลื่อนที่ (ผู้ใช้อุปกรณ์เคลื่อนที่) โดยเฉพาะอุปกรณ์เคลื่อนที่ในปัจจุบันที่สามารถเชื่อมต่ออินเทอร์เน็ตได้ (สมาร์ตโฟน) โดยที่ประโยชน์ที่คาดว่าจะได้รับจากโครงการมีดังนี้

1) เพิ่มความตระหนักและประสิทธิภาพการตรวจจับมัลแวร์ที่ไม่แสดงตัวตนในตอนต้น แต่มีพฤติกรรมอันตรายขณะใช้งาน
2) ป้องกันการขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่โดยมีการแจ้งเตือนผู้ใช้ถึงภัยคุกคาม
3) ลดความเสี่ยงการเกิดอาชญากรรมทางไซเบอร์ที่อาจก่อให้เกิดความเสียหายต่อเศรษฐกิจในระดับบุคคลและองค์กร
4) เพิ่มขีดความสามารถของประเทศไทยในการป้องกันภัยคุกคามด้านไซเบอร์เพื่อผลักดันการเจริญเติบโตทางเศรษฐกิจดิจิทัลอย่างยั่งยืน

1.5 ตัวชี้วัดผลผลิตและตัวชี้วัดผลลัพธ์

ตัวชี้วัดผลผลิตของโครงการที่เกี่ยวข้องกับกระบวนการทำงานและผลการดำเนินงานมีดังนี้

1) ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายบนอุปกรณ์เคลื่อนที่พัฒนาได้เสร็จตามระยะเวลาและงบประมาณที่กำหนด
2) ระบบที่พัฒนาสามารถตรวจจับแอปพลิเคชั่นอันตรายได้อย่างถูกต้องแม่นยำและมีประสิทธิภาพ

3) ระบบที่พัฒนาสามารถเตือนภัยให้ผู้ใช้ตระหนักถึงระดับอันตรายของแอปพลิเคชั่นได้อย่างถูกต้องและมีความเที่ยงตรง

ตัวชี้วัดผลลัพธ์หรือประโยชน์ที่เกิดจาการนำผลผลิตจากโครงการไปใช้ให้เกิดประโยชน์มีดังนี้

1) อุปกรณ์เคลื่อนที่มีความมั่นคงปลอดภัยจากการโจรกรรมข้อมูลส่วนบุคคลและข้อมูลแสดงตัวตนเพิ่มมากขึ้น
2) ผู้บริโภคมีความเชื่อมั่นต่อระบบรักษาความปลอดภัยในอุปกรณ์เคลื่อนที่เพิ่มมากขึ้น
3) หน่วยงานรัฐบาลและภาคเอกชนในประเทศไทยมีขีดความสามารถเพิ่มขึ้นในการป้องกันภัยคุกคามด้านไซเบอร์เพิ่มขึ้น

บรรณานุกรม

สำนักงานคณะกรรมการวิจัยแห่งชาติ. (ตุลาคม, 2555).แผนยุทธศาสตร์การวิจัยและพัฒนาเกี่ยวกับการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ ฉบับที่ 1 พ.ศ. 25562560 (หน้า 6).

คณะกรรมการความมั่นคงปลอดภัยไซเบอร์แห่งชาติกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร. (มีนาคม, 2556).กรอบนโยบายการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ พ.ศ. 2555-2559, การประชุมคณะกรรมการความมั่นคงปลอดภัยไซเบอร์แห่งชาติ, 11 มิถุนายน พ.ศ. 2556.

สำนักงานคณะกรรมการวิจัยแห่งชาติ. (สิงหาคม, 2555).นโยบายและยุทธศาสตร์การวิจัยของชาติฉบับที่ 8, 1 สิงหาคม พ.ศ. 2555.

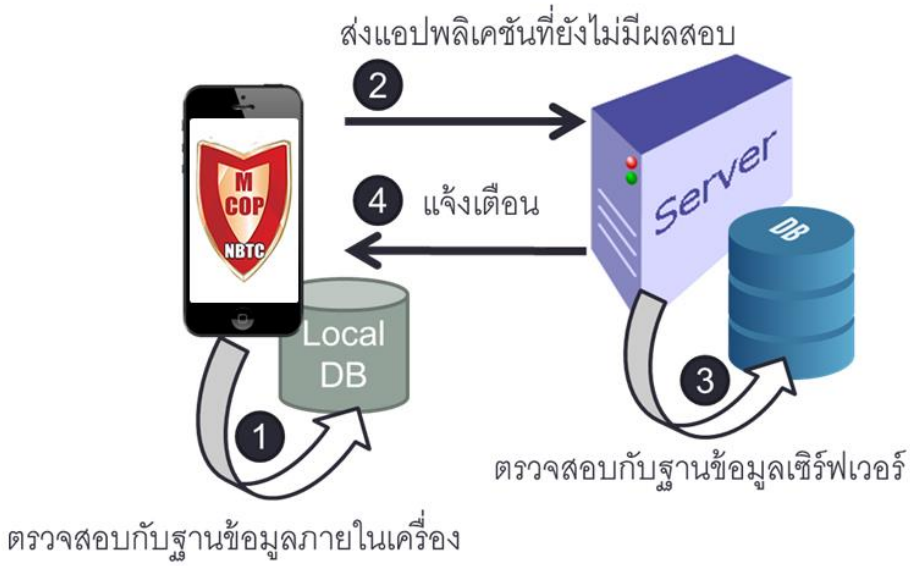
Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011, October). A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices* (pp. 3-14). ACM.

บทที่ 2
กรอบแนวคิดและหลักการทำงานของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายบนโทรศัพท์เคลื่อนที่

2.1 กรอบแนวคิดและหลักการทำงานของระบบโดยรวม

กล่าวถึงระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายที่นำเสนอนี้ เพื่อให้ง่ายต่อความเข้าใจ ชื่อเรียกของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายนี้คือ “mCOP by NBTC” ซึ่งมีที่มาจากคำ Malware Cop หรือ ตำรวจสำหรับแอปพลิเคชั่นอันตราย สถาปัตยกรรมของระบบเป็นแบบแม่ข่ายลูกข่าย ซึ่งฝ่ายแม่ขายหมายถึงเซิร์ฟเวอร์ระบบ และฝ่ายลูกข่ายหมายถึงแอปพลิเคชั่น mCOP ที่ติดตั้งบนอุปกรณ์มือถือของผู้ใช้

หลักตรวจจับแอปพลิเคชั่นอันตราย (มัลแวร์) โดย mCOP System มีดังนี้ ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์ว่าเป็นแอปพลิเคชั่นอันตรายหรือไม่ โดยส่งข้อมูลไปตรวจสอบที่เซิร์ฟเวอร์และและส่งผลสอบกลับไปยังผู้ใช้ โดยมีเงื่อนไขว่าก่อนการส่งข้อมูลไปยังเซิร์ฟเวอร์ การตรวจสอบสามารถทำที่ฝั่งอุปกรณ์ได้โดยใช้ฐานข้อมูลแอปพลิเคชั่นที่ได้รับผลสอบแล้ว เพื่อลดจำนวนการส่งข้อมูลจากอุปกรณ์ไปยังเซิร์ฟเวอร์ (ดังแสดงในภาพที่ 1)



ภาพที่ 1. ภาพรวมของการตรวจสอบแอปพลิเคชั่นอันตรายโดย mCop System

สำหรับมุมมองของผู้ใช้ mCOP สามารถเฝ้าระวังและเตือนภัยแอปพลิเคชั่นอันตราย แจกแจงเป็นกรณีได้ดังนี้

กรณีที่ 1) ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์ (คำอธิบายในตารางที่ 1)

กรณีที่ 2) หลังจากที่ได้ติดตั้ง mCOP App ระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายที่แฝงตัวมากับแอปพลิเคชั่นปกติ (คำอธิบายในตารางที่ 2)

ตารางที่ 1. การทำงานของระบบสำหรับกรณีที่ 1

(ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App)

กรณี	การทำงาน	ฝั่งอุปกรณ์	ฝั่งเซิร์ฟเวอร์	ผู้ใช้
1) ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App	ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์	1. ตรวจสอบกับฐานข้อมูลภายในอุปกรณ์		
		2. ส่งแอปพลิเคชั่นที่ไม่พบไปยังเซิร์ฟเวอร์		
			3. ตรวจสอบกับฐานข้อมูลภายในเซิร์ฟเวอร์	
			4. ส่งผลสอบ	
		5. อัพเดตฐานข้อมูลภายในอุปกรณ์		
				6. ผู้ใช้ได้รับผลสอบและสามารถเลือกจัดการกับแอปพลิเคชั่นอันตรายได้ดังนี้ - ลบแอปพลิเคชั่นนั้นออกจากอุปกรณ์ (Uninstall) - เพิกเฉย (Ignore)

ารางที่ 2. การทำงานของระบบสำหรับกรณีที่ 2 (หลังจากที่ได้ติดตั้ง mCOP App)

กรณี	การทำงาน	ฝั่งอุปกรณ์	ฝั่งเซิร์ฟเวอร์	ผู้ใช้
2) หลังจากที่ได้ติดตั้ง mCOP App	ระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายที่แฝงตัวมากับแอปพลิเคชั่นปกติ	1. สกัดฟีเจอร์การใช้งานของแต่ละแอปพลิเคชั่น		
		2. คำนวณค่าสถิติของฟีเจอร์ที่ได้		
		3. ตรวจสอบโดยใช้โมเดลตรวจจับแอปพลิเคชั่นที่น่าสงสัย		
		4. ตรวจพบแอปพลิเคชั่นที่น่าสงสัย		
			5. ส่งแอปพลิเคชั่นที่น่าสงสัยไปยังเซิร์ฟเวอร์	
			6. ตรวจสอบกับฐานข้อมูลเซิร์ฟเวอร์	
				7. ผู้ใช้ได้รับผลสอบและสามารถเลือกจัดการกับแอปพลิเคชั่นอันตรายได้ดังนี้ - ลบแอปพลิเคชั่นนั้นออกจากอุปกรณ์ (Uninstall) - เพิกเฉย (Ignore)

2.2 แนวคิดการเฝ้าสังเกตพฤติกรรมการใช้ทรัพยากรบนโทรศัพท์เคลื่อนที่

การเฝ้าระวังและตรวจจับแอปพลิเคชั่นขณะใช้งานบนอุปกรณ์จึงเป็นเรื่องจำเป็น เช่น ในช่วงห้าวันแรกหลังจากติดตั้งแอปพลิเคชั่นการอัพเดตเวอร์ชั่นเป็นการอัพเดตแบบปกติ แต่ในวันที่หกแอปพลิเคชั่นนี้มีพฤติกรรมอันตรายกลายเป็นมัลแวร์ด้วยการดาวน์โหลดโค้ดอันตรายมาที่อุปกรณ์ เป็นต้น การแยะแยะแอปพลิเคชั่นที่มีพฤติกรรมน่าสงสัยจากแอปพลิเคชั่นปกติทั่วไปจึงสามารถทำได้โดยใช้การเฝ้าระวังและตรวจสอบขณะที่แอปพลิเคชั่นทำงานบนอุปกรณ์ เพื่อตรวจจับพฤติกรรมของแอปพลิเคชั่นที่เปลี่ยนแปลงจากเดิมอย่างมีนัยสำคัญและแจ้งให้ผู้ใช้งานทราบถึงภัยที่กำลังคุกคาม โดยย่อการเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายบนโทรศัทพ์เคลื่อนที่มีขั้นตอนดังนี้ (ดังแสดงในภาพที่ 2)

1) การเฝ้าสังเกตพฤติกรรมการใช้ทรัพยากรของแอปพลิเคชั่น โดยเก็บข้อมูลทรัพยากรที่แอปพลิเคชั่นใช้และจองโดยแอปพลิเคชั่น เช่น การใช้งานหน่วยประมวลผลการ การใช้หน่วยความจำ การรับส่งข้อมูลผ่านอินเตอร์เน็ต เป็นต้น
2) การตรวจพบพฤติกรรมการใช้ทรัพยากรของแอปพลิเคชั่นที่ผิดปกติ โดยเปรียบเทียบกับรูปแบบมาตรฐานว่าแตกต่างออกไปอย่างมีนัยสำคัญทางสถิติ และใช้เทคนิคการเรียนรู้ของเครื่องในการแยกแยะพฤติกรรมที่ผิดปกติ
3) การแจ้งเตือนผู้ใช้ถึงความผิดปกติที่เกิดขึ้น โดยผู้ใช้สามารถสั่งระงับหรือหยุดการทำงานของแอปพลิเคชั่น และลบแอปพลิเคชั่นออกจากอุปกรณ์ (ถอนการติดตั้ง)

ภาพที่ 2. ภาพรวมของการเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตราย

ลักษณะของแอปพลิเคชั่นอันตรายที่แฝงตัวอยู่ในอุปกรณ์พกพาและสามารถผ่านการตรวจสอบของแอนติไวรัสได้ ส่วนใหญ่อยู่ในรูปแบบแอปพลิเคชั่นปลอม[1] (หนึ่งในวิธีอำพรางตัวของโค้ดอันตราย) การเฝ้าสังเกตพฤติกรรมการใช้งานของแอปพลิเคชั่นสามารถแบ่งได้เป็น 2 ระดับ ดังนี้

1) พฤติกรรมการใช้งานที่ผิดปกติที่เกิดขึ้นในระดับแอปพลิเคชั่น หมายถึง รูปแบบการใช้งาน ณ ขณะนั้นแตกต่างจากรูปแบบมาตรฐานของแอปพลิเคชั่น การตรวจสอบระดับนี้เหมาะกับตรวจสอบแอปพลิเคชั่นปลอม

2) พฤติกรรมการใช้งานที่ผิดปกติที่เกิดขึ้นเทียบในระดับประเภทของแอปพลิเคชั่น หมายถึง รูปแบบการใช้งาน ณ ขณะนั้นแตกต่างจากรูปแบบมาตรฐานของแอปพลิเคชั่น การตรวจสอบระดับนี้เหมาะกับตรวจสอบแอปพลิเคชั่นอันตรายที่รูปแบบมาตรฐานของแอปพลิเคชั่นยังไม่มีในฐานข้อมูล

รูปแบบมาตรฐานการใช้งานของแอปพลิเคชั่น หมายถึง โมเดลการใช้งานของแอปพลิเคชั่นปกติ (ไม่ใช่แอปพลิเคชั่นปลอม) โดยคำนวณจากข้อมูลการใช้ทรัพยากรของแอปพลิเคชั่นที่เก็บรวบรวมจากผู้ใช้จำนวน 100 คน และเก็บข้อมูลเป็นระยะเวลา 30 ถึง 120 วัน[2] ขณะที่รูปแบบมาตรฐานการใช้งานแยกเป็นประเภทของแอปพลิเคชั่นสามารถคำนวณต่อจากรูปแบบมาตรฐานระดับแอปพลิเคชั่น (ดังแสดงในภาพที่ 3)

ภาพที่ 3. ภาพรวมของการสร้างรูปแบบมาตรฐานการใช้งานของแอปพลิชั่น

---

[1]แอปพลิเคชั่นปกติที่มีโค้ดอันตรายแฝงตัวอยู่ ส่วนใหญ่เป็นแอปพลิเคชั่นที่มีผู้ใช้ให้ความนิยม เช่น Facebook, Skype, WhatsApp เป็นต้น

[2]การเก็บรวบรวมข้อมูลตามแผนงานในช่วง เดือนที่ 6 - 9 ของโครงการ (วันที่ 23 มีนาคม 2559 ถึง 22 มิถุนายน 2559)

2.3 ความปลอดภัยระดับแอปพลิเคชั่น (Application-Layer Security)

ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นฯ ที่พัฒนาขึ้น อาศัยข้อมูลในระดับ Application layer ในการประเมินพฤติกรรมของแต่ละแอปพลิเคชั่น เพื่อระบุแนวโน้มที่แต่ละ แอปพลิเคชั่น จะเป็นแอปพลิเคชั่นอันตราย ทั้งนี้ การใช้ข้อมูลในระดับ application layer เพื่อการวิเคราะห์และตรวจจับ เป็นรูปแบบการตรวจจับที่นิยมกันแพร่หลาย ตัวอย่างเช่น การใช้ข้อมูลดังกล่าว เพื่อสนับสนุน Application Awareness บน Next Generation Firewall (NGFW)

2.3.1 Next Generation Firewall (NGFA)

Next Generation Firewall (NGFW) เป็น Firewall รูปแบบใหม่ที่รวมเอา 3 เทคนิคหลักเข้าด้วยกัน ได้แก่ การทำงานของ Firewall มาตรฐาน, ระบบ Intrusion Prevention System (IPS) และเทคนิค Application Control ทั้งนี้ NGFW ปัจจุบันยังรวมเอาฟังก์ชั่นหรือบริการด้าน networking และ security อื่นๆเอาไว้ด้วย เช่น Packet-filtering พร้อมทั้ง deep-packet inspection, NAT, VPN support, QoS Support เป็นต้น

เพื่อสนับสนุนการทำงานของระบบ IPS และการทำ Application Control ที่ควบคุมการใช้งานแอปพลิเคชั่น NGFW จึงจำเป็นต้องรับและเข้าใจ application-layer traffic เพื่อใช้ข้อมูลดังกล่าวมาประเมินความเป็นไปได้ที่ traffic ดังกล่าวจะสนับสนุนหรือเอื้อต่อการโจมตี หรือละเมิดนโยบายการใช้งาน โดยมีจุดมุ่งหมายที่จะสกัดกั้น (block) traffic เหล่านั้น

อย่างไรก็ตาม ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นฯ มีความแตกต่างจาก NGFW ในหลากหลายด้าน ตั้งแต่คุณลักษณะของระบบ รูปแบบการรวบรวมข้อมูล จนไปถึงจุดมุ่งหมายในการให้บริการ

จุดประสงค์ของระบบ

1) NGFW มุ่งสกัด (block) traffic ที่เป็นอันตราย ละเมิด security policy หรือ สงสัยว่าจะเป็นส่วนหนึ่งของการโจมตี เป็นการป้องกันระบบ (Prevention) โดยทำการ classify และ block packet รวมถึงเพื่อประเมินระดับความสำคัญของ traffic สำหรับให้ priority หรือ throttle traffic ตาม QoS ที่ตั้ง
2) ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นฯ ที่พัฒนาขึ้น เป็นระบบตรวจจับความผิดปกติโดยพิจารณาจากพฤติกรรมของระบบ (Behavior-based Detection) โดยเน้นการแจ้งเตือนและไม่มีการสกัด (block) การเชื่อมต่อ หรือการทำงานใดๆ

2.3.2 ขอบเขตและตำแหน่งการทำงาน

Firewall ถูกออกแบบมาเพื่อให้การป้องกันระดับรอบนอก (Network Perimeter Defense) ให้กับโฮสต์ (Host-based Firewall) หรือเครือข่าย (Network-based Firewall) โดย NGFW ที่พบในท้องตลาดจะอยู่ในรูปแบบของ Network-based Firewall ดังนั้น NGFW จึงได้พิจารณาข้อมูลจากหลาย endpoint แต่ไม่มีข้อมูลเชิงลึกของแต่ละ endpoint ไม่มีได้รับข้อมูลอื่นที่ไม่ได้ถูกส่งออกไปกับ traffic

ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นฯที่พัฒนาขึ้น ถูกออกแบบมาให้ทำงานที่ endpoint (Host-based) โดยพิจารณาพฤติกรรมของแอปพลิเคชั่นต่างๆที่ทำงานอยู่บนแต่ละ Mobile Endpoint จึงมีข้อมูลเชิงลึกของแต่ละ Mobile endpoint เป็นอย่างดี แต่ไม่มีข้อมูลเกี่ยวกับ entities อื่นที่อยู่ใน network เดียวกัน (ถ้ามี)

2.3.3 ที่มาของข้อมูล

NGFW ได้รับข้อมูลระดับ application layer ผ่านกลไก Deep packet inspection (DPI) โดยเป็นข้อมูลจาก application payload ของ network packet อันได้แก่ เนื้อหา (data หรือ code) ใน payload รวมถึงระบุแอปพลิเคชั่นที่เป็นต้นทางของ packet

ระบบการตรวจจับและเตือนภัยแอปพลิเคชั่นฯ ใช้ข้อมูล แอปพลิเคชั่น รวมถึงระดับและรูปแบบการใช้ resource ของ แอปพลิเคชั่น จากระบบปฏิบัติการ (Operating System) จึงได้ข้อมูลหลากหลายและในเชิงลึก เท่าที่ระบบปฏิบัติการจะอนุญาต

บรรณานุกรม

เว็บไซต์ทางการของระบบปฏิบัติการแอนดรอยด์, https://www.android.com/, สืบคืนเมื่อ 15 เมษายน 2559

Enck, William, Machigar Ongtang, and Patrick McDaniel. "Understanding android security." IEEE security & privacy 1 (2009): 50-57.

Annuzzi Jr, Joseph, Lauren Darcey, and Shane Conder. Advanced Android Application Development. Pearson Education, 2014.

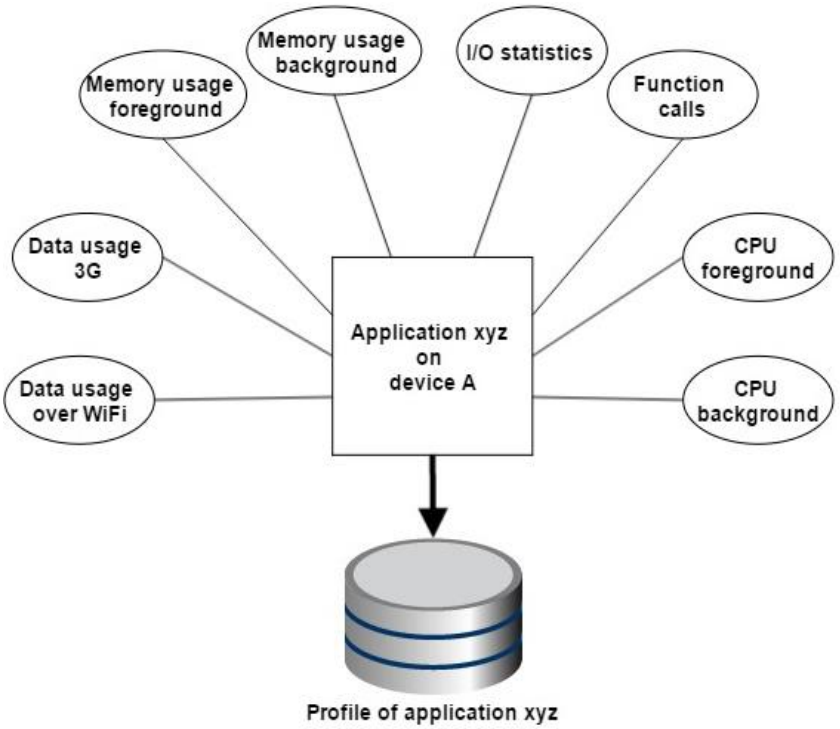
Gandhewar, Nisarg, and Rahila Sheikh. "Google Android: An emerging software platform for mobile devices." International Journal on Computer Science and Engineering 1.1 (2010): 12-17.

บทที่ 3
การสร้างและติดตั้งโมดูลสกัดฟีเจอร์บนโทรศัพท์เคลื่อนที่

3.1 กรอบการทำงานโดยรวมของโมดูลสกัดฟีเจอร์

การเก็บข้อมูลการใช้งานของแต่ละแอปพลิเคชั่นเป็นการเก็บข้อมูลการใช้ทรัพยากร เช่น สมมติแอปพลิเคชั่น xyz รันบนอุปกรณ์ A (แสดงในภาพที่ 4) โมดูลสกัดฟีเจอร์ทำหน้าที่เฝ้าสังเกตพฤติกรรมการใช้ทรัพยากรของแอปพลิเคชั่น xyz โดยเก็บข้อมูลต่อไปนี้

1) ปริมาณข้อมูลที่รับส่งผ่าน WiFi (Data usage over WiFi)
2) ปริมาณข้อมูลที่รับส่งผ่าน Cellular (Data usage over 3G/4G)
3) ขนาดของหน่วยความจำที่ใช้ในพื้นหน้า (Memory usage on foreground)
4) ขนาดของหน่วยความจำที่ใช้ในพื้นหลัง (Memory usage on background)
5) สถิติการใช้ I/O (I/O statistics)
6) ปริมาณการใช้งานหน่วยประมวลผลกลางในพื้นหน้า (CPU usage on foreground)
7) ปริมาณการใช้งานหน่วยประมวลผลกลางในพื้นหลัง (CPU usage on background)

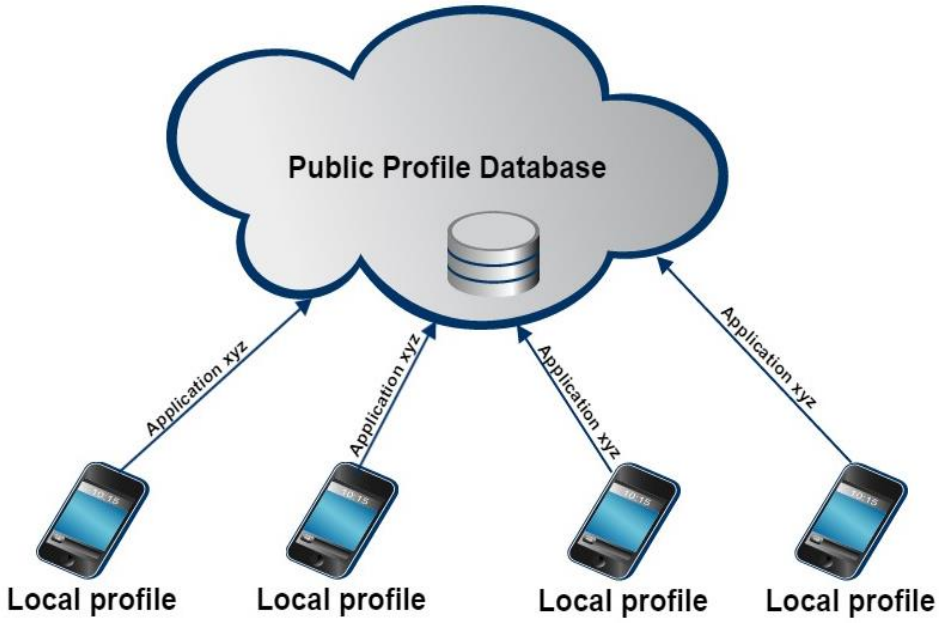


ภาพที่ 4. รายชื่อทรัพยากรที่ระบบเฝ้าสังเกตการณ์ขณะรันแอปพลิเคชั่น xyz บนอุปกรณ์ A

ความถี่ของการสกัดฟีเจอร์ (เก็บข้อมูลทรัพยากร) อาจส่งผลกระทบต่อโหลดงานที่เพิ่มขึ้นบนอุปกรณ์ ดังนั้นความถี่ที่เหมาะสมในการเก็บข้อมูลทรัพยากรขึ้นกับปัจจัยต่อไปนี้ 1) ระดับการรักษาความปลอดภัยของอุปกรณ์ที่ผู้ใช้กำหนด และ 2) ระดับความเสี่ยงของแอปพลิเคชั่นที่ติดตั้งอยู่บนอุปกรณ์ ระบบจะกำหนดค่าเริ่มต้นให้กับความถี่ที่เหมาะสมในการเก็บข้อมูลซึ่งอ้างอิงจากผลการทดลอง (แสดงในหัวข้อที่ 2.5)

3.2 โมดูลสกัดฟีเจอร์ใช้ในการสร้างโมเดลตรวจจับพฤติกรรมการใช้งานที่ผิดปกติ

สำหรับการสร้างโมเดลตรวจจับพฤติกรรมการใช้งานที่ผิดปกติ การเก็บรวมรวมข้อมูลในเฟสของการพัฒนา คณะผู้จัดทำโครงการจำเป็นต้องออกแบบสถาปัตยกรรมโมดูลสกัดฟีเจอร์ในรูปแบบแม่ข่ายลูกข่าย โดยโมดูลสกัดฟีเจอร์จะรันทั้งสองฝั่ง หลักการทำงานเป็นดังนี้ ชุดของข้อมูลที่ได้เปรียบเสมือนโปรไฟล์ (Profile) ของแอปพลิเคชั่น โมดูลสกัดฟีเจอร์บันทึกโปรไฟล์ไว้ในฐานข้อมูลของอุปกรณ์ (Local profile) เป็นระยะเวลาหนึ่งก่อนส่งโปรไฟล์ไปที่เซิร์ฟเวอร์ของระบบ (Public profile database) ดังแสดงในภาพที่ 5



ภาพที่ 5. โปรไฟล์ของแอปพลิเคชั่น xyz ของแต่ละอุปกรณ์ส่งต่อไปที่เซิร์ฟเวอร์ของระบบ

การส่งโปรไฟล์ไปที่เซิร์ฟเวอร์ระบบต้องเป็นโปรไฟล์ของแอปพลิเคชั่นที่ใช้งานอยู่ (กำลังรัน) ดังแสดงในตัวอย่าง แอปพลิเคชั่น xyz เป็นแอปพลิเคชั่นที่ใช้งานอยู่บนอุปกรณ์ A และอุปกรณ์อื่นอีก 3 เครื่อง โดยปกติแล้วแอปพลิเคชั่นที่ใช้งานอยู่บนอุปกรณ์มีจำนวนมากกว่าหนึ่งแอปพลิเคชั่น การเชื่อมต่อกับเซิร์ฟเวอร์แต่ละครั้งจะส่งเป็น Log file (ลักษณะของ Log file แสดงในภาพที่ 6) เพื่อลดภาระการประมวลผลบนอุปกรณ์ และระบบสามารถสร้างโปรไฟล์ของแอปพลิเคชั่นหลังจากส่งไปที่เซิร์ฟเวอร์แล้ว

id	logtime	application_id	is_main_proc...	is_interacti...	pcy	cpu	vss	rss	threads	priority	status	bg_up_data	bg_down_data	fg_up_data	fg_down_data	bg_up_wifi	bg_down_wifi	fg_up_wifi	fg_down_wifi
1	2016-01-11 14:56:47....	30	1	0	0	0	957004	38764	13	1	S	0	0	0	0	0	0	0	0
2	2016-01-11 14:56:47....	31	1	0	0	0	1226180	110936	73	0	S	0	0	0	0	0	0	0	0
3	2016-01-11 14:56:47....	32	1	0	0	0	1103964	109604	62	0	S	0	0	0	0	0	0	0	0
4	2016-01-11 14:56:47....	33	1	0	0	0	979036	47692	17	1	S	0	0	0	0	0	0	0	0
5	2016-01-11 14:56:47....	34	1	0	0	0	1062224	62560	50	1	S	0	0	0	0	0	0	0	0
6	2016-01-11 14:56:47....	35	1	0	0	0	1049088	88720	45	0	S	0	0	0	0	0	0	0	0
7	2016-01-11 14:56:47....	36	1	0	0	0	1015184	58576	27	0	S	0	0	0	0	0	0	0	0
8	2016-01-11 14:56:47....	37	1	0	0	0	978888	55228	27	1	S	0	0	0	0	0	0	0	0
9	2016-01-11 14:56:47....	38	1	1	0	0	984732	56688	19	0	S	0	0	0	0	0	0	0	0
10	2016-01-11 14:56:51....	30	1	0	0	0	957004	38764	13	1	S	0	0	0	0	0	0	0	0
11	2016-01-11 14:56:51....	31	1	0	0	0	1226180	110936	73	0	S	0	0	0	0	0	0	0	0
12	2016-01-11 14:56:51....	32	1	0	0	0	1103964	109604	62	0	S	0	0	0	0	0	0	0	0
13	2016-01-11 14:56:51....	33	1	0	0	0	979036	47692	17	1	S	0	0	0	0	0	0	0	0
14	2016-01-11 14:56:51....	34	1	0	0	0	1062224	62560	50	1	S	0	0	0	0	0	0	0	0
15	2016-01-11 14:56:51....	35	1	0	0	0	1049088	88720	45	0	S	0	0	0	0	0	0	0	0
16	2016-01-11 14:56:51....	36	1	0	0	0	1015184	58576	27	0	S	0	0	0	0	0	0	0	0
17	2016-01-11 14:56:51....	37	1	0	0	0	978888	55228	27	1	S	0	0	0	0	0	0	0	0
18	2016-01-11 14:56:51....	38	1	1	0	0	984576	57352	19	0	S	0	0	0	0	0	0	0	0

ภาพที่ 6. ตัวอย่าง Log file

3.3 หน้าที่การทำงานของโมดูลสกัดฟีเจอร์

หน้าที่ของโมดูลสกัดฟีเจอร์ คือ สกัดข้อมูลการใช้งานทรัพยากรที่แอปพลิเคชั่นใช้งานจริงขณะรันบนอุปกรณ์ ทรัพยากรบนอุปกรณ์ที่ระบบให้ความสนใจเพื่อใช้ในการวิเคราะห์และแยกแยะแอปพลิเคชั่นอันตรายได้ คือ ข้อมูลการใช้งานหน่วยประมวลผลกลาง (CPU Usage) และข้อมูลจราจรบนเครือข่าย (Network traffic)

3.3.1 ข้อมูลการใช้งานหน่วยประมวลผลกลาง

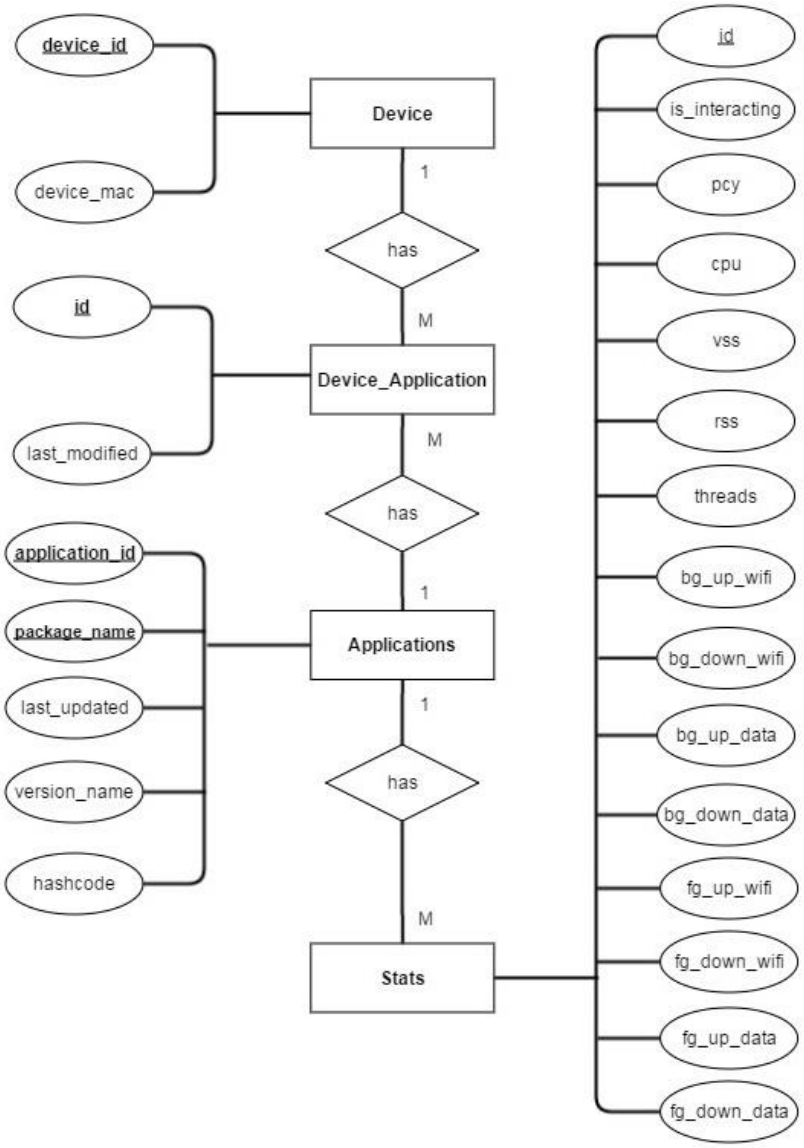
ข้อมูลการใช้งานหน่วยประมวลผลกลางได้จากการใช้คำสั่ง Top command ในระบบปฏิบัติการแอนดรอยด์ เป็นคำสั่งที่เหมือนกับในระบบปฏิบัติการบนพีซี การใช้คำสั่งนี้สามารถได้ข้อมูลการใช้งานซีพียู ที่เป็นปัจจุบัน (เรียลไทม์) ข้อมูลที่แสดงมีทั้ง รายชื่อโปรเซสหรือเทรดที่กำลังรัน และได้ข้อมูลสำคัญเช่น จำนวนของเทรดที่สร้างขึ้นจากแต่ละโปรเซส (No. of Threads) ร้อยละการใช้งานของซีพียู (CPU%) พื้นที่ในหน่วยความจำที่ใช้ (Memory size) ลำดับความสำคัญของโปรเซส (Process priority) ลักษณะการรันแบบพื้นหน้าหรือพื้นหลัง (Foreground/Background) สถานะของโปรเซส เช่น หลับ รัน พัก เป็นต้น (Sleeping, running, waiting)

แอปพลิเคชั่นที่ติดตั้งบนระบบปฏิบัติการแอนดรอยด์จะได้รับเลขประจำตัว (UID) ที่ไม่ซ้ำกับแอปพลิเคชั่นอื่น แอปพลิเคชั่นสามารถมีโปรเซสที่กำลังรันได้มากกว่า 1 โปรเซส แต่ทุกโปรเซสภายใต้แอปพลิเคชั่นต้องได้รับเลขประจำตัวเดียวกัน ตัวอย่างเช่น แอปพลิเคชั่นเฟซบุ๊คสร้างโปรเซส com.facebook.kantan และโปรเซสลูก com.facebook.katana:dash และ com.facebook.katana.videoplayer ทั้งหมดมีเลขประจำตัวเดียวกัน

3.3.2 ข้อมูลจราจรบนเครือข่าย

ระบบปฏิบัติการแอนดรอยด์เก็บข้อมูลจราจรบนเครือข่ายไว้ในไฟล์ “/proc/net/xt_qtaguid/stats” การบันทึกข้อมูลจราจรจะแยกตามแอปพลิเคชั่นซึ่งระบุด้วยเลขประจำตัวของแอปพลิเคชั่นนั้น โดยมีรายละเอียดของข้อมูลว่าเป็นโปรเซสลักษณะที่รันบนพื้นหน้าหรือพื้นหลัง (background/foreground) รายละเอียดของข้อมูลจราจรจะกล่าวในบทที่ 3

3.3.3 ฐานข้อมูลการใช้ทรัพยากรของแอปพลิเคชั่นที่อยู่บนเซิร์ฟเวอร์



ภาพที่ 7. แผนภาพความสัมพันธ์ระหว่างข้อมูลการใช้ทรัพยากรของแอปพลิเคชั่น

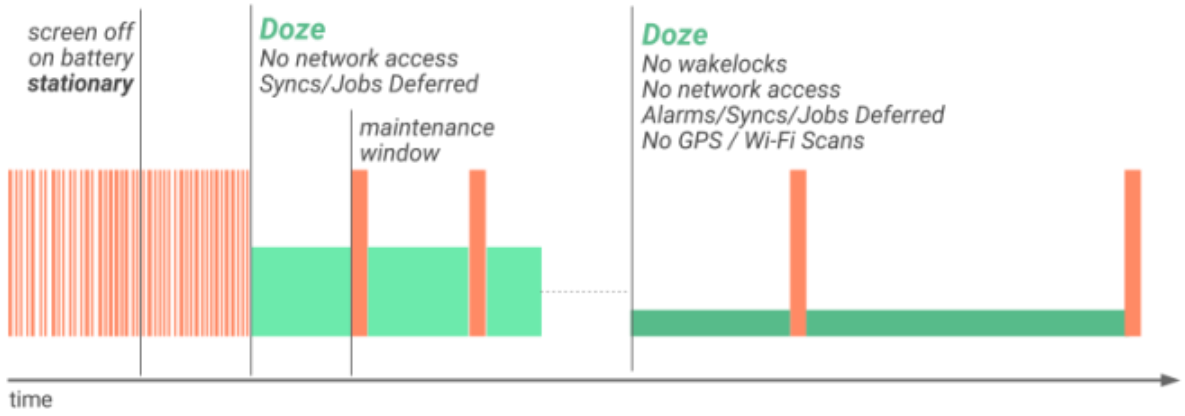
แผนภาพความสัมพันธ์ระหว่างข้อมูล (Entity Relationship Diagram : E-R Diagram) ที่แสดงในภาพที่ 7 ข้อมูลการใช้งานของแต่ละแอปพลิเคชั่นได้จัดเก็บในฐานข้อมูลบนเครื่องแม่ข่าย จากภาพแสดงให้เห็นว่าอุปกรณ์มือถือจำนวนหนึ่งเครื่องสามารถมีได้หลายแอปพลิเคชั่น ซึ่งแอปพลิเคชั่นแต่ละแอปพลิเคชั่นยังสามารถมีได้หลายเวอร์ชั่นอีกด้วย ระบบได้จำแนกแอปพลิเคชั่นตาม “Package name” ยกตัวอย่างเช่น แอปพลิเคชั่น xyz มีจำนวน 3 เวอร์ชั่น แต่ละเวอร์ชั่นจะมีชื่อแพ็คเก็ตเดียวกันทั้งหมด แต่จะมีเลขเวอร์ชั่น ชื่อ

เวอร์ชั่น และ Hash code (สามารถดูรายละเอียดของ ER-diagram ในภาคผนวก ก.) ข้อมูลการใช้งานแอปพลิเคชั่นของแต่ละเวอร์ชั่นจะเก็บลงในตาราง Stats

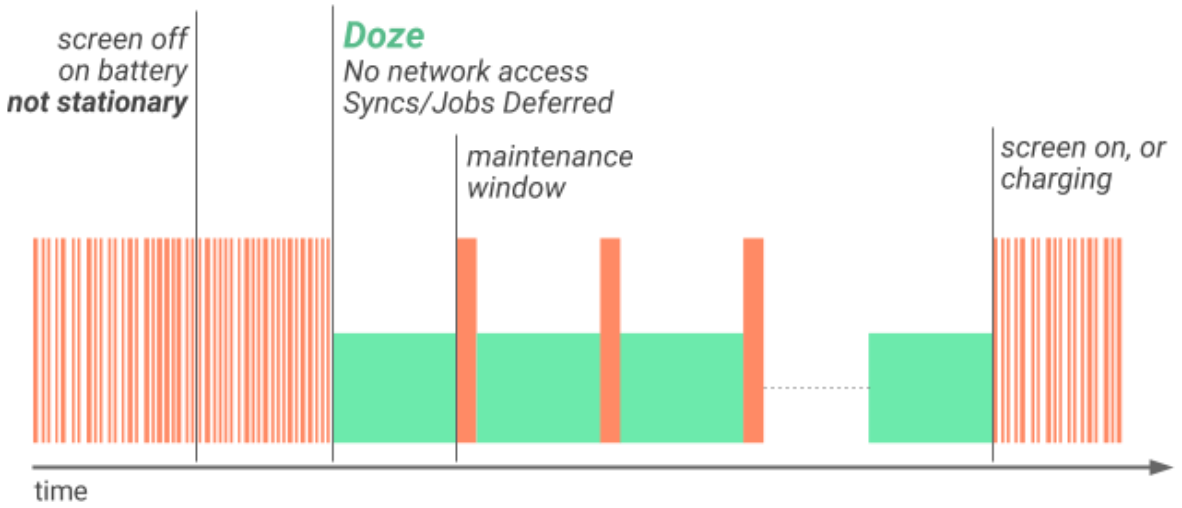
3.4 กรณีปิดหน้าจอและรันบนแบตเตอรี่ (Screen off on battery)

การเฝ้าระวังและตรวจจับแอปพลิเคชั่นขณะใช้งานบนอุปกรณ์ จำเป็นต้องคำนึงถึงการใช้ทรัพยากรของแอปพลิเคชั่นที่รันบนอุปกรณ์ กรณีที่หน้าจอของอุปกรณ์ปิดแต่ไม่ได้เสียบสายชาร์จ (รันบนแบตเตอรี) การเฝ้าระวังยังมีความจำเป็นแต่ความถี่ของการเก็บข้อมูลการใช้ทรัพยากรต้องมีรูปแบบที่เปลี่ยนไป เพื่อประหยัดพลังงานที่ใช้จากแบตเตอรี่

สำหรับระบบปฏิบัติการแอนดรอยด์ 6.0 (Marshmallow) การปิดหน้าจอและรันบนแบตเตอรี่เรียกสั้นๆ ว่า “Doze mode” ในเวอร์ชั่น 6.0 นี้ แอนดรอยด์ได้ใส่ฟีเจอร์การทำงานใน Doze mode ด้วยการบังคับให้ทุกแอปพลิเคชั่นลด/หยุดการทำงาน โดยแบ่งเป็น 2 ลักษณะ 1) อุปกรณ์วางนิ่ง และ 2) อุปกรณ์ไม่ได้วางนิ่ง



ภาพที่ 8. การใช้ทรัพยากรของอุปกรณ์ใน Doze mode ขณะที่อุปกรณ์วางนิ่ง
ที่มา: http://www.androidauthority.com/android-n-doze-678982/ สืบค้นวันที่ 15 เมษายน 2559



ภาพที่ 9. การใช้ทรัพยากรของอุปกรณ์ใน Doze mode ขณะที่อุปกรณ์ไม่ได้วางนิ่ง
ที่มา: http://www.androidauthority.com/android-n-doze-678982/ สืบค้นวันที่ 15 เมษายน 2559

สำหรับระบบปฏิบัติการแอนดรอยด์ เวอร์ชั่นที่ต่ำกว่า 6.0 โมดูลสกัดฟีเจอร์จำเป็นต้องปรับความถี่การเก็บข้อมูลให้สอดคล้องกับการทำงานใน Doze mode เพื่อประหยัดพลังงาน ความถี่ในการเก็บข้อมูลใช้เทคนิค Step-down การลดลงทีละขั้น

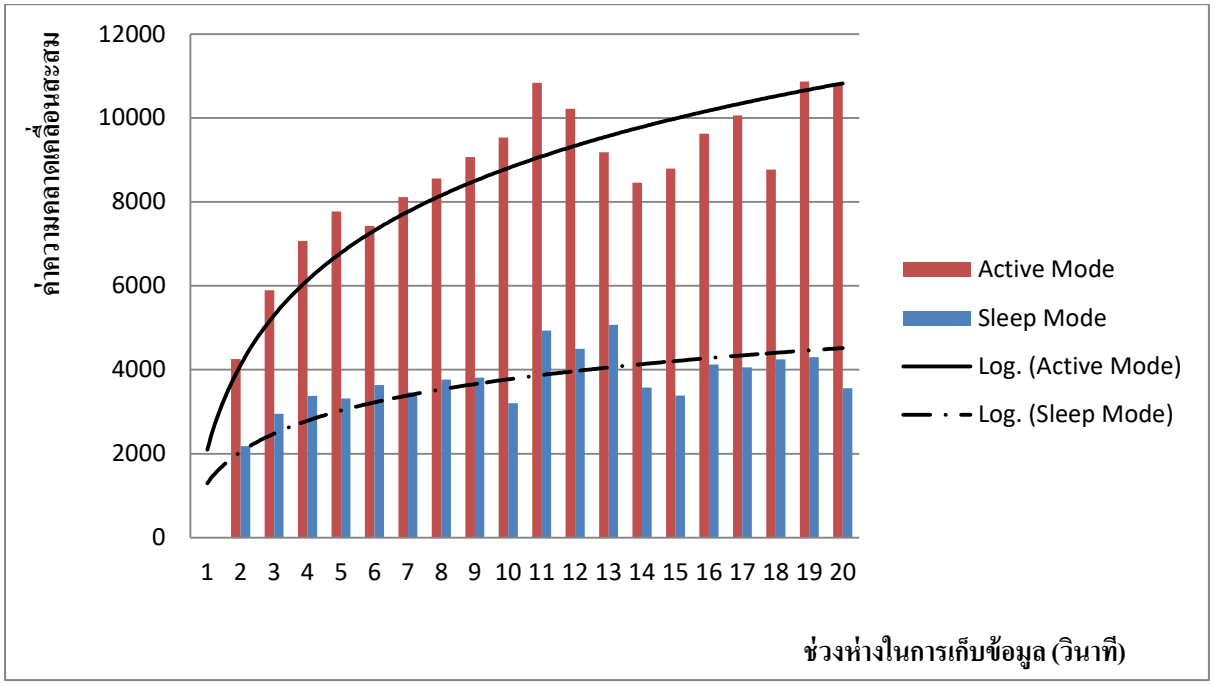
3.5 การกำหนดค่าความถี่ในการเก็บข้อมูลการใช้ทรัพยากร

โมดูลสกัดฟีเจอร์ควรต้องเฝ้าสังเกตพฤติกรรมของแอปพลิเคชั่นตลอดเวลา เพื่อตรวจจับพฤติกรรมที่ผิดปกติ ค่าความถี่ที่เหมาะสมในการเก็บข้อมูลการใช้ทรัพยากรขึ้นกับ 1) ระดับการรักษาความปลอดภัยของอุปกรณ์ที่ผู้ใช้กำหนด และ 2) ระดับความเสี่ยงของแอปพลิเคชั่นที่ติดตั้งอยู่บนอุปกรณ์

การกำหนดค่าความถี่ (ช่วงห่าง) ของการเก็บข้อมูล ในเบื้องต้นสามารถอ้างอิงจากผลการทดลองที่แสดงในภาพที่ 10 เป็นกราฟแสดงค่าความคาดเคลื่อนสะสมเมื่อกำหนดช่วงห่างของการเก็บข้อมูลตั้งแต่ 1-20 วินาที ค่าความคลาดเคลื่อนคำนวณจากข้อมูลจริงที่เก็บทุกวินาที เปรียบเทียบกับข้อมูลที่ได้จากฟังก์ชั่นการประมาณค่าในช่วง

ฟังก์ชั่นการประมาณค่าในช่วงที่นำมาใช้เป็นเชิงเส้น (Linear interpolation) โดยข้อมูลที่ป้อนเข้าไปนั้นมีเป็นชุดข้อมูลที่ได้จาก ข้อมูลที่เก็บทุก 2 นาที (ช่วงห่าง = 2 นาที) เก็บทุก 3นาที จนถึง เก็บทุก 20 นาที

กราฟแสดงข้อมูลที่ทั้งใน Active mode และ Sleep mode โดยที่ชัดเจนว่าค่าความคาดเคลื่อนใน Active mode จะสูงกว่า Sleep mode ถึงสองเท่าตัว กราฟนี้ยังแสดงให้เห็นเส้นโค้งประมาณค่าความคาดเคลื่อนในลักษณะของฟังก์ชันลอการิทึม (Logarithmic function)



ภาพที่ 10. กราฟค่าความคาดเคลื่อนสะสมเมื่อกำหนดช่วงห่างของการเก็บข้อมูลตั้งแต่ 1-20 วินาที

เมื่อพิจารณาค่าความถี่ในการเก็บข้อมูลที่เหมาะสมแบ่งเป็น 4 ระดับ ความถี่มากที่สุด มาก ปานกลาง และน้อย เพื่อให้ระดับความถี่สอดคล้องกับระดับการรักษาความปลอดภัยของอุปกรณ์ที่ผู้ใช้กำหนด ค่าความถี่ควรกำหนดดังนี้

1) ระดับการรักษาความปลอดภัยสูงสุด ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 2 วินาที (ช่วงห่าง=2)
2) ระดับการรักษาความปลอดภัยสูง ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 4 วินาที (ช่วงห่าง=4)
3) ระดับการรักษาความปลอดภัยปานกลาง ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 8 วินาที (ช่วงห่าง=8)
4) ระดับการรักษาความปลอดภัยน้อย ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 15 วินาที (ช่วงห่าง=15)

สำหรับใน Sleep mode ค่าความถี่ควรกำหนดดังนี้

1) ระดับการรักษาความปลอดภัยสูงสุด ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 2 วินาที (ช่วงห่าง=2)
2) ระดับการรักษาความปลอดภัยสูง ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 12 วินาที (ช่วงห่าง=12)

3) ระดับการรักษาความปลอดภัยปานกลาง ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 90 วินาที (ช่วงห่าง=90)

4) ระดับการรักษาความปลอดภัยน้อย ควรใช้ค่าความถี่ในการเก็บข้อมูลทุก 600 วินาที (ช่วงห่าง=600)

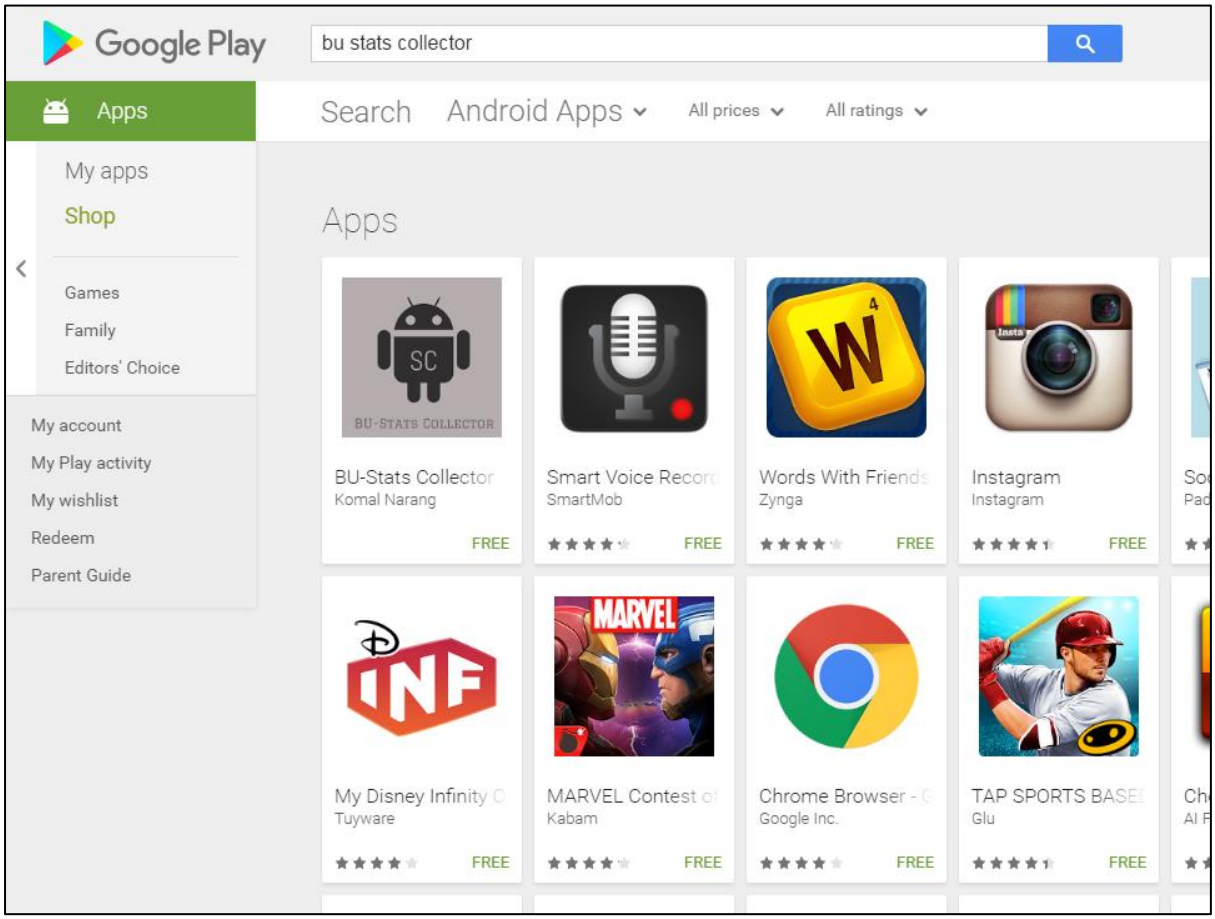
3.6 การติดตั้งและทดสอบโมดูลสกัดฟีเจอร์ผ่านอุปกรณ์มือถือ

การติดตั้งโมดูลสกัดเจอร์สามารถทำได้โดยดาวโหลดแอปพลิเคชั่น ชื่อว่า “BU-Stats Collector” นี้ผ่าน Google Play (ในภาพที่ 11) หรือ URL ด้านล่าง พร้อมคำอธิบายแอปพลิเคชั่นแสดงในภาพที่ 12

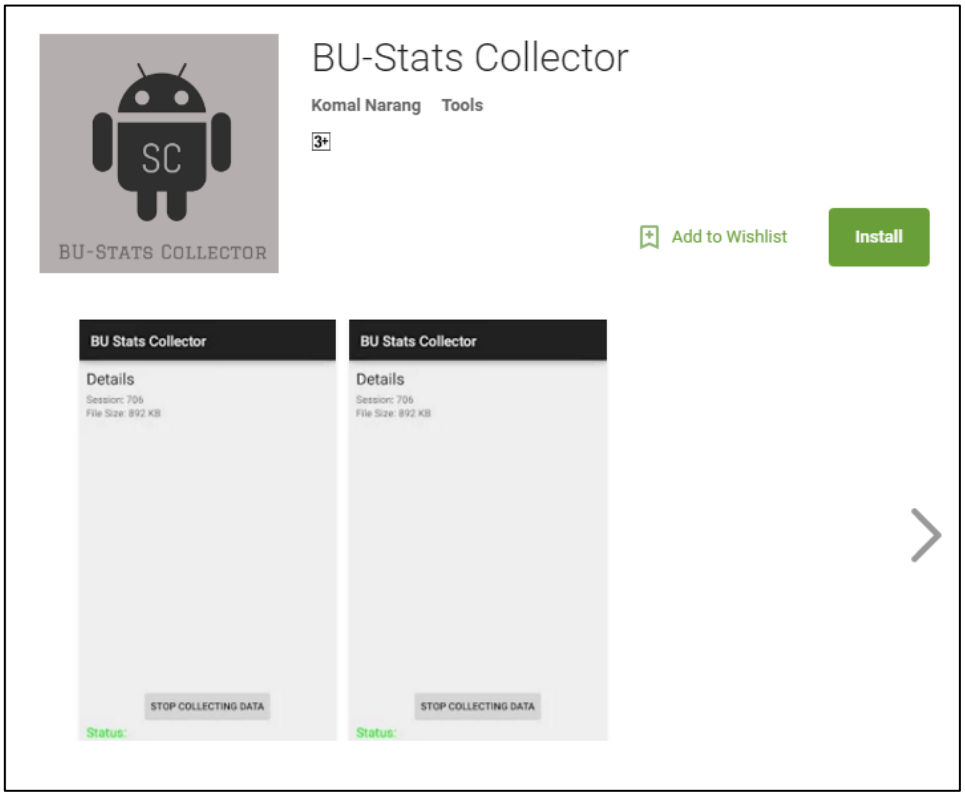
https://play.google.com/store/apps/details?id=th.ac.bu.science.mit.allappstatscollector&hl=en

แอปพลิเคชั่นในเวอร์ชันนี้มีเป้าหมายเพื่อเก็บข้อมูลการใช้ทรัพยากรของแอปพลิเคชั่นบนอุปกรณ์ของผู้ใช้ การเก็บข้อมูลนี้จะนำมาใช้ในการสร้างโมดูลตรวจจับแอปพลิเคชั่นที่มีพฤติกรรมที่น่าสงสัย

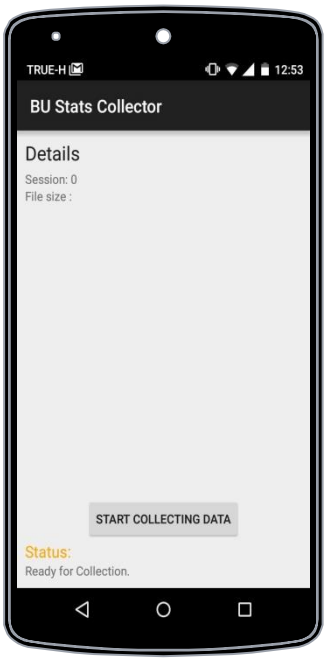
เมื่อผู้ใช้ค้นหาแอปพลิเคชั่นโดยพิมพ์ “BU stats collector” ผลลัพธ์ที่ได้จะแสดงในภาพที่ 11 ซึ่งสะดวกในการให้อาสาสมัครดาวโหลดและติดตั้งแอปพลิเคชั่นนี้ และเมื่อเริ่มใช้งานจะเห็นหน้าตาของแอปพลิเคชั่นดังภาพที่ 13



ภาพที่ 11. BU-Stats Collector อยู่ใน Google Play



ภาพที่ 12. หน้าตาของ BU-Stats Collector บน Google Play



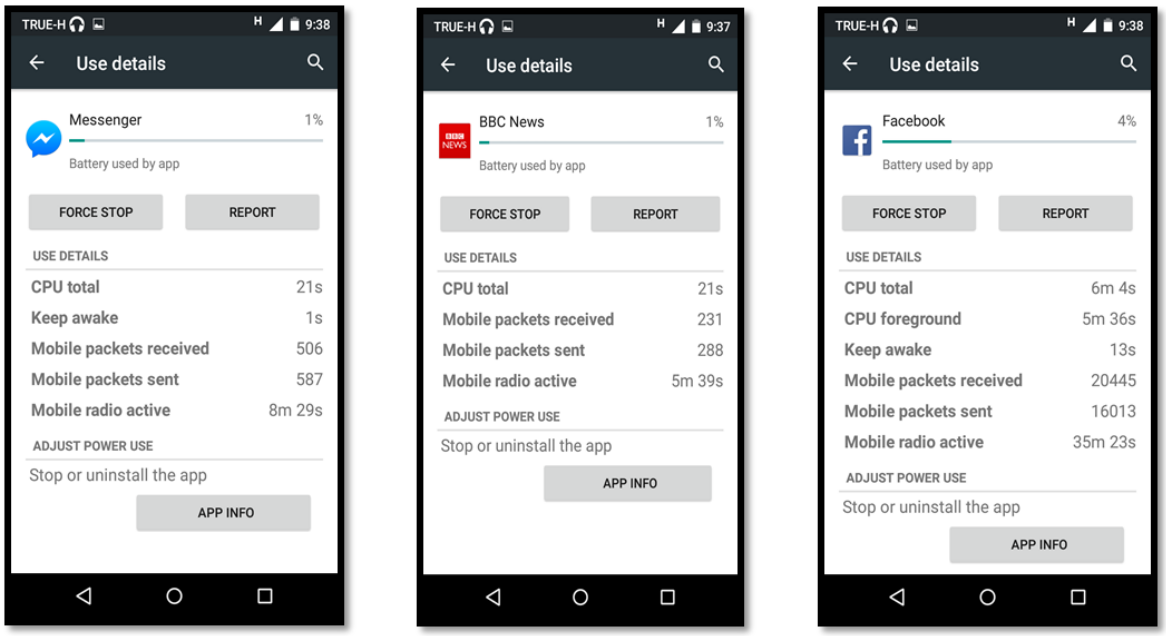
ภาพที่ 13. เริ่มต้นใช้งานแอปพลิเคชั่น BU Stats Collector

การทดสอบการใช้งานของ BU-Stats Collector ผ่านอุปกรณ์มือถือที่หลากหลายในเรื่องของ รุ่น ยี่ห้อ และเวอร์ชันของระบบปฏิบัติการ จนถึงปัจจุบันได้ทดสอบกับอุปกรณ์มือถือจำนวน 6 เครื่อง ดังนี้

1) Samsung Galaxy J5 --จำนวน 1 เครื่อง
2) Samsung Galaxy J7 --จำนวน 2 เครื่อง
3) Samsung Note 3 --จำนวน 1 เครื่อง
4) Samsung Tab 3 --จำนวน 1 เครื่อง
5) Asus ZenFone 2 --จำนวน 1 เครื่อง

การทดสอบเริ่มตั้งแต่ 23 กุมภาพันธ์ 2559 จนถึงปัจจุบัน ข้อมูลที่เก็บได้มีจำนวนมากกว่า 200,000 บรรทัด (records) อย่างไรก็ตามจำนวนอุปกรณ์ที่ใช้ทดสอบนั้นมีเพียง 3 เครื่องที่รันแอปพลิเคชั่นนี้ต่อเนื่อง สำหรับอุปกรณ์อีก 3 เครื่องนั้นต้องใช้ในการพัฒนาและทดสอบโมดูลอื่น

ตัวอย่างของการใช้งานแสดงในภาพที่ 14 แอปพลิเคชั่น BU Stat Collector สามารถแสดงรายละเอียดของการใช้ทรัพยากรของแต่ละแอปพลิเคชั่น เช่น Messenger, BBC News, Facebook เป็นต้น เมื่อผู้ใช้เห็นรายละเอียดการใช้งาน แอปพลิเคชั่นเพิ่มฟีเจอร์ในการสั่งหยุดทำงาน หรือส่งรายงานความผิดปกติไปยังระบบ



ภาพที่ 14. ตัวอย่างการแสดงผลของ BU Stats Collector

บรรณานุกรม

เว็บไซต์ทางการของระบบปฏิบัติการแอนดรอยด์, https://www.android.com/, สืบคืนเมื่อ 15 เมษายน 2559

Enck, William, Machigar Ongtang, and Patrick McDaniel. "Understanding android security." IEEE security & privacy 1 (2009): 50-57.

Annuzzi Jr, Joseph, Lauren Darcey, and Shane Conder. Advanced Android Application Development. Pearson Education, 2014.

Gandhewar, Nisarg, and Rahila Sheikh. "Google Android: An emerging software platform for mobile devices." International Journal on Computer Science and Engineering 1.1 (2010): 12-17.

บทที่ 4
การสร้างโมดูลประมวลผลฟีเจอร์ด้วยเทคนิคทางสถิติ

4.1 กรอบการทำงานโดยรวมของโมดูลประมวลผลฟีเจอร์

โมดูลประมวลผลฟีเจอร์เป็นเสมือนการเตรียมข้อมูลนำเข้าโมดูลตรวจจับแอปพลิเคชั่นโดยใช้เทคนิคการรู้จำของเครื่อง (Machine learning) ซึ่งการทำให้ฟีเจอรที่ได้จากโมดูลสกัดฟีเจอร์มีความหมายและสะท้อนพฤติกรรมได้ วิธีการเตรียมข้อมูลที่นิยมใช้คือการหาคำนวณค่าสถิติพื้นฐาน และนิยามฟีเจอร์ประมวลผลตัวใหม่ขึ้นมา

โมดูลประมวลผลฟีเจอร์นี้จะรันบนอุปกรณ์และเซิร์ฟเวอร์ โดยมีวัตถุประสงค์ที่ต่างกันดังนี้

1) ฝั่งเซิฟร์เวอร์: ใช้ในขั้นตอนการพัฒนาโมดูลตรวจจับแอปพลิเคชั่นอันตราย
2) ฝั่งอุปกรณ์: ใช้ในการเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตราย

4.2 การประมวลผลฟีเจอร์

การประมวลผลฟีเจอร์ คือ ขั้นตอนการคำนวณค่าทางสถิติของข้อมูลจราจรบนเครือข่าย ค่าทางสถิติที่ได้จะสามารถสื่อความหมายและสะท้อนถึงลักษณะของพฤติกรรมการใช้งานได้ ตัวอย่างการคำนวณที่นำมาใช้เป็นการหาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน ค่าน้อยสุด ค่ามากสุด เปอร์เซ็นต์ไทล์ เป็นต้น ด้านล่างเป็นฟีเจอร์ที่ได้จากการคำนวณทางสถิติ

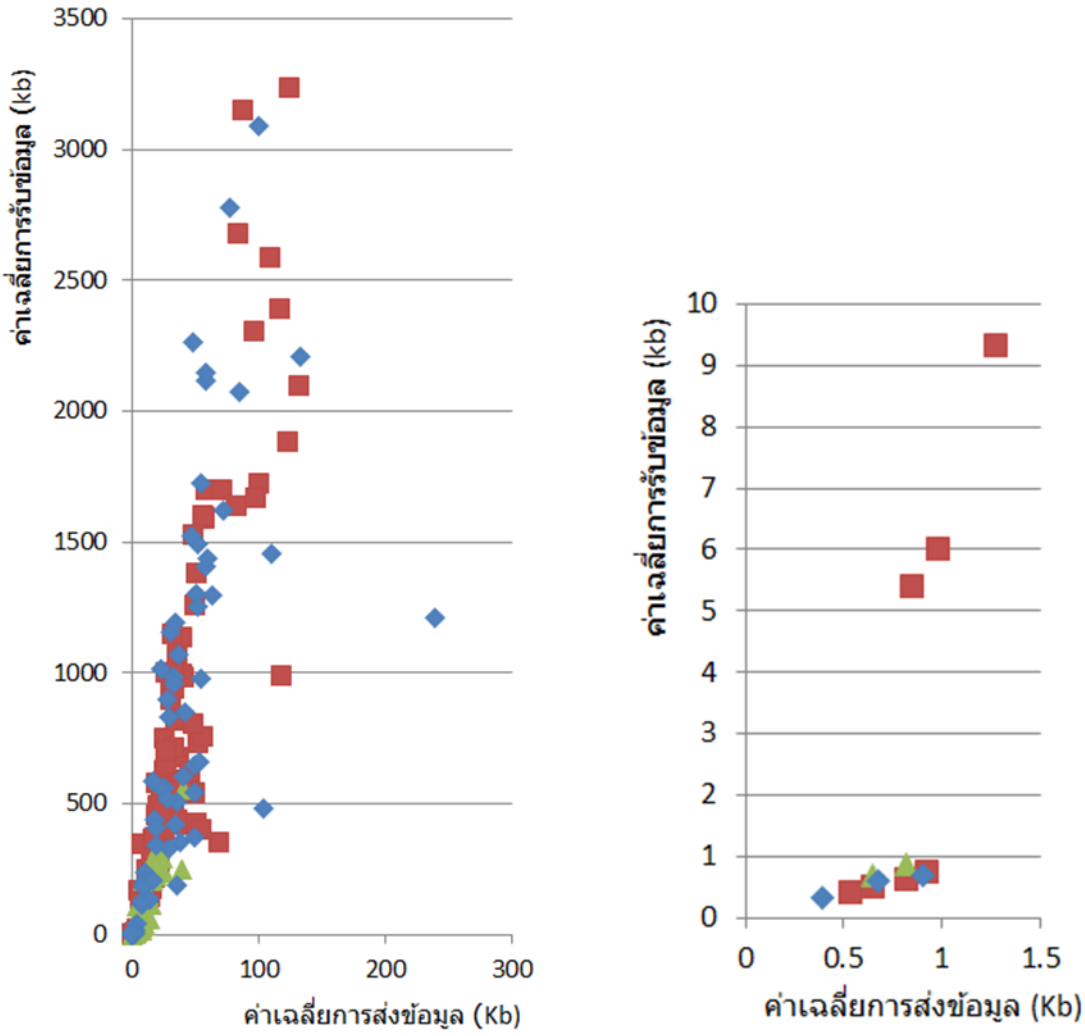
1) ค่าเฉลี่ย, ค่าเบี่ยงเบนมาตรฐาน, ค่าน้อยสุด และมากสุดของการรับ/ส่งข้อมูล
2) ค่าเฉลี่ย, ค่าเบี่ยงเบนมาตรฐาน, ค่าน้อยสุด และมากสุดของการรับ/ส่งข้อมูลที่เป็นอัตราส่วนของการรับส่งข้อมูลทั้งหมด
3) ร้อยละของการรับ/ส่งข้อมูล (ไบต์)
4) ช่วงห่างของเหตุการณ์การรับ/ส่งข้อมูล --- เน้นเหตุการณ์การรับ/ส่งข้อมูล แบ่งเป็น 2 แบบ
    ก) ค่าเฉลี่ยในช่วง (inner average) เหตุการณ์การรับ/ส่งข้อมูล ที่เกิดขึ้นในช่วง 30 วินาทีจากเหตุการณ์การรับ/ส่งข้อมูลก่อนหน้า จะทำการคำนวณค่าเฉลี่ยในช่วง
    ข) ค่าเฉลี่ยนอกช่วง (outer average) เหตุการณ์การรับ/ส่งข้อมูลที่เกิดขึ้นมากกว่าหรือเท่ากับ 30วินาทีจากเหตุการณ์ก่อนหน้า จะทำการคำนวณค่าเฉลี่ยนอกช่วง (outer average)

5) สถานะของเครือข่าย (Network state) – Cellular, Wifi ไม่มีสัญญาณ หรือแบบผสมผสาน สถานะผสมผสานหมายถึงกรณีที่การเชื่อมต่อเครือข่ายมีหลายสถานะ (เช่น ทั้ง cellular และ Wifi) ในช่วงเวลาที่รวมข้อมูล
6) ระยะเวลาเป็นนาทีนับตั้งแต่แอปพลิเคชั่นทำการรับส่งข้อมูลครั้งล่าสุด
7) สถานะของแอปพลิเคชั่นแบบที่ 1 – foreground, background หรือ แบบผสมผสาน สถานะแบบผสมผสานหมายถึงกรณีที่มีหลายสถานะเกิดในช่วงเวลาที่รวมข้อมูล
8) สถานะของแอปพลิเคชั่นแบบที่ 2 – active, non-active หรือ แบบผสมผสาน สถานะแบบผสมผสานหมายถึงกรณีที่มีหลายสถานะเกิดในช่วงเวลาที่รวมข้อมูล
9) ระยะเวลาโดยรวมและระยะเวลาที่เจาะจง (วินาที) ของสถานะของแอปพลิเคชั่น (foreground / background) ช่วงเวลาเป็น 2 ประเภท – เจาะจง (local) คือช่วงเวลาที่เจาะจงในการรวมข้อมูล และ โดยรวม (global) เป็นค่าเฉลี่ยของกระบวนการตรวจสอบทั้งหมด ทั้งนี้ ช่วงเวลา local จะครอบคลุมพฤติกรรมของแอปพลิเคชั่นในช่วงเวลาใดๆ ในขณะที่ ช่วงเวลา global อธิบายถึงพฤติกรรมโดยรวมของแอปพลิเคชั่นเริ่มตั้งแต่จุดเริ่มสังเกตการจนถึงปัจจุบัน
10) ทั้งนี้ระยะเวลา local อาจเป็นไปได้ตั้งแต่ 0 – 60 วินาที ซึ่งแสดงถึงค่าในการรวมข้อมูลในขณะที่ระยะเวลารวมคือเวลาทั้งหมดที่แอปพลิเคชั่นถูกใช้งาน
11) ระยะเวลาเป็นนาทีนับตั้งแต่แอปพลิเคชั่น active ครั้งล่าสุด
12) ระยะเวลาเป็นวันนับตั้งแต่แอปพลิเคชั่นมีการแก้ไขล่าสุด (application’s last modified time)

4.3 รูปแบบพฤติกรรมการใช้งานทรัพยากรของแอปพลิเคชั่น

เพื่อแสดงให้เห็นว่าแอปพลิเคชั่นแต่ละแอปพลิเคชั่นควรจะมีรูปแบบการจราจรบนเครือข่ายที่เฉพาะตัว ดังแสดงในภาพที่ 16 ซึ่งเป็นการแสดงรูปแบบของการรับส่งข้อมูลจากผู้ใช้หลายคน สีจุดข้อมูลแสดงถึงผู้ใช้ต่างคนกัน เพื่อให้เห็นภาพที่ชัดเจนกราฟที่นำเสนอจึงเลือกแสดงในรูปแบบ 2 มิติ โดยใช้ตัวแปรค่าเฉลี่ยการส่งข้อมูลและค่าเฉลี่ยการรับข้อมูล จุดที่ได้แต่ละจุดเป็นค่าเฉลี่ยคำนวณจากการรับส่งข้อมูลภายใน 5 นาที เมื่อใช้ข้อมูลของ Facebook เทียบกับ BBC News จะเห็นได้ว่า รูปแบบการใช้งานของ Facebook แตกต่างจาก BBC News อย่างชัดเจน หรืออาจกล่าวได้ว่า รูปแบบการใช้งานของ Facebook และ BBC News น่าจะมีลักษณะเฉพาะตัว

ถึงแม้ว่าตัวอย่างในภาพที่ 16 จะแสดงเพียง 2 ตัวแปร (มิติ) แต่การสร้างโมดูลตรวจจับจะนำตัวแปรทั้งหมด (ทุกมิติ) มาวิเคราะห์ร่วม



ภาพที่ 16. รูปแบบพฤติกรรมที่ใช้การประมวลผลฟีเจอร์เปรียบเทียบระหว่าง Facebook เทียบกับ BBC News

บทที่ 5
การออกแบบและพัฒนาอัลกอริทึมในการตรวจจับแอปพลิเคชั่นอันตราย

สำหรับแอปพลิเคชั่นอันตราย (มัลแวร์) การตรวจจับแอปพลิเคชั่นอันตรายแบ่งเป็น 2 กรณี (กล่าวโดยละเอียดในบทที่ 2) คือ กรณีที่ 1) ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์ และกรณีที่ 2) หลังจากที่ได้ติดตั้ง mCOP App ระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายที่แฝงตัวมากับแอปพลิเคชั่นปกติ

เพื่อให้สอดคล้องกับทั้งสองกรณี อัลกอริทึมในการตรวจจับแอปพลิเคชั่นอันตรายแบ่งออกเป็น 3 ส่วนหลัก ได้แก่ (1) การตรวจสอบแอปพลิเคชั่นอันตบนเครื่องโดยใช้แฮชโค้ด (ฐานข้อมูล) (2) การตรวจสอบการใช้งานที่ผิดปกติของ SMS และ (3) การประเมินพฤติกรรมของแต่ละแอปพลิเคชั่น เพื่อระบุแนวโน้มที่แต่ละแอปพลิเคชั่น จะกลายเป็นแอปพลิเคชั่นอันตราย

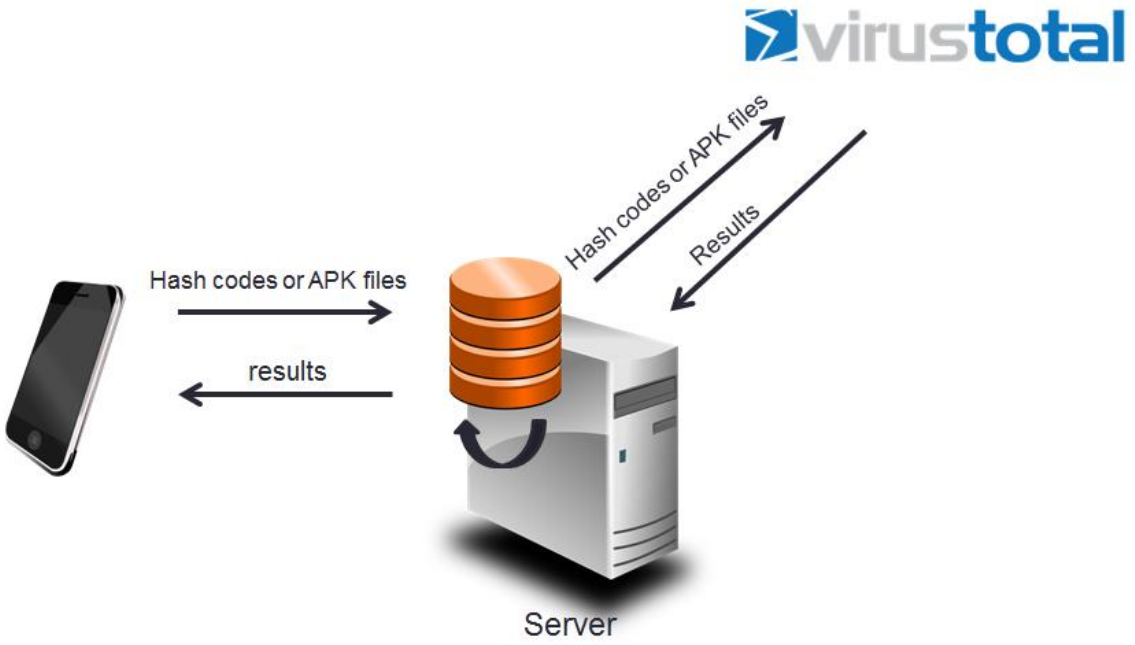
สำหรับส่วนแรก การตรวจสอบแอปพลิเคชั่นบนเครื่องกับฐานข้อมูลที่มีอยู่เพื่อให้ผู้ใช้มั่นใจว่าทุกแอปพลิเคชั่นบนเครื่องเป็นแอปพลิเคชั่นปกติไม่ใช่มัลแวร์ หลังจากนั้นส่วนที่ 2-3 เป็นการเฝ้าระวังว่าแอปพลิเคชั่นใดที่มีพฤติกรรมที่น่าสงสัย (แอปพลิเคชั่นปลอม) เพื่อตรวจจับและเตือนภัยให้กับผู้ใช้งาน

5.1 การตรวจสอบแอปพลิเคชั่นบนเครื่องกับฐานข้อมูลที่มีอยู่

หลังจากผู้ใช้ติดตั้งแอป mCOP แล้ว ระบบจะทำการตรวจสอบเบื้องต้นว่า แอปพลิเคชั่น ที่มีในเครื่องขณะนั้นมีความเสี่ยงในระดับใด[1] โดยตรวจสอบเทียบกับฐานข้อมูลบนฝั่งเซิร์ฟเวอร์ดังแสดงในภาพที่ 17

หลังการติดตั้งแฮชโค้ด (Hashcode) ของทุก แอปพลิเคชั่น ที่อยู่ในเครื่องจะถูกส่งไปยังเซิร์ฟเวอร์เพื่อตรวจสอบระดับความเสี่ยงของ แอปพลิเคชั่น ทั้งหมด ในกรณีที่ฐานข้อมูลบนเซิร์ฟเวอร์มีข้อมูลของแอปพลิเคชั่น ดังกล่าวอยู่แล้ว เซิร์ฟเวอร์จะทำการแจ้งผลลัพธ์กลับมาให้ผู้ใช้

[1] การตรวจสอบระดับความเสี่ยงของ แอปพลิเคชั่น ในเครื่องจะทำเพียงครั้งเดียวหลังติดตั้ง ซึ่งขั้นตอนนี้เป็นส่วนเพิ่มเติมและไม่ได้เป็นการประเมินพฤติกรรมของแต่ละแอปพลิเคชั่นแต่อย่างใด



ภาพที่ 17. กรอบการทำงานในด้านการตรวจสอบแอปพลิเคชั่นบนเครื่องกับฐานข้อมูล

ทั้งนี้ในกรณีที่ฐานข้อมูลบนเซิร์ฟเวอร์ยังไม่มีข้อมูลของแอปพลิเคชั่นนั้นๆ ทางเซิร์ฟเวอร์จะทำการส่งต่อแฮชโค้ดไปยัง Virus Total[2] ซึ่งฝั่ง Virus Total จะทำการตรวจดูว่าแฮชโค้ด ดังกล่าวเคยถูกสแกนแล้วหรือไม่ ถ้าปรากฏว่ามีข้อมูลอยู่แล้วก็ทำการส่งผลลัพธ์กลับมาให้ และทางเซิร์ฟเวอร์ของ mCOP ก็ทำการเพิ่มข้อมูลลงในฐานข้อมูลก่อนรายงานผลไปยังผู้ใช้

Virus Total เป็นเว็บไซต์ที่ Google เป็นเจ้าของ เปิดให้บริการออนไลน์ฟรีในการช่วยวิเคราะห์ไฟล์และ URL ที่เข้าข่ายว่าจะเป็นไวรัส เวิร์ม โทรจันและเนื้อหาที่เป็นอันตรายชนิดอื่นๆ โดยตรวจจับจาก antivirus engine มากกว่า 50 ตัว โดยภารกิจ Virus Total คือการช่วยในการปรับปรุงการป้องกันไวรัสและการรักษาความปลอดภัย รวมไปถึงการทำให้อินเทอร์เน็ตเป็นสถานที่ที่ปลอดภัยด้วยการพัฒนาเครื่องมือและบริการ

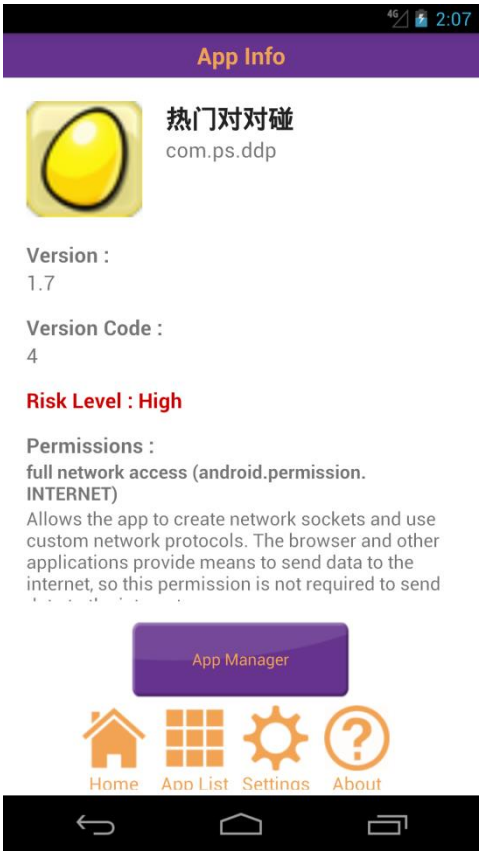
[2] Virus Total เป็นเว็บไซต์ที่ Google เป็นเจ้าของ เปิดให้บริการออนไลน์ฟรีในการช่วยวิเคราะห์ไฟล์

5.1.1 การวิเคราะห์ระดับความเสี่ยง

เนื่องจาก Virus Total ใช้ Antivirus Engines มากกว่า 50 ตัว ได้แก่ BitDeffender, McAfee, AVG, Norton, Dr.Web, AegisLab เป็นต้น หลักการที่ใช้วิเคราะห์ระดับความเสี่ยงของแอปพลิเคชันนั้นได้ใช้หลักการของการ vote โดยผลลัพธ์ของการ scan ของแต่ละ anti virus engine คือการ vote จากนั้นได้นำผลของการ vote มาคิดคะแนนรวม ทั้งนี้ เนื่อง engine ที่ virus total ใช้มีความแหลกหลายมากที่ประกอบด้วย enigines ที่เป็นที่รู้จักอย่าง McAfee และ BitDeffender และ enignes ที่ไม่เป็นที่นิยมมากนัก การให้น้ำหนักของการ vote จึงต้องแตกต่างกัน ทางทีมพัฒนาจึงได้ศึกษาข้อมูล Antivirus จากแหล่งในการจัดอันดับ 4 แหล่ง (Top Antivirus) ตัวอย่างเช่น pcmag.com toptenreviews.com และ tomsguide.com เพื่อมากำหนดค่าน้ำหนักของ Antivirus แต่ละตัว โดยค่าน้ำหนักจะมีค่าตั้งแต่ 0.4 ถึง 4 หมายความว่า Antivirus จากค่ายใหญ่จะมีค่าน้ำหนักมากกว่าค่ายเล็กมากสุดถึง 10 เท่า

คะแนนรวมของการ vote คิดเป็น 100คะแนน หมายถึง ถ้าคะแนนที่ได้เป็น 0 แสดงว่าไม่มี engine ใดเลยระบุว่าแอปพลิเคชันนั้นเป็น malware ในขณะที่คะแนนที่ได้เป็น 100 แสดงว่าทุก engine ระบุว่าแอปพลิเคชันนั้นเป็น malware โดยได้ทำการแบ่งระดับของ malware ออกเป็น 4 ระดับดังนี้

- คะแนน <= 5 ระดับ ปลอดภัย (Safe)
- 5 < คะแนน <= 25 ระดับ เสี่ยงน้อย (Low Risk)
- 25 < คะแนน <= 50 ระดับ เสี่ยง (Risk)
- 50 < คะแนน <= 100 ระดับ เสี่ยงมาก (High Risk)



ภาพที่ 18. ตัวอย่างการแจ้งเตือนระดับความเสี่ยง (Risk Level: High) ของระบบผ่านหน้าอินเทอร์เฟซกับผู้ใช้

5.2 การตรวจสอบการใช้งานที่ผิดปกติของ SMS

โมดูลตรวจจับการส่งข้อความ SMS นี้ มีจุดมุ่งหมายเพื่อตรวจจับ SMS ที่ส่งออกไปโดยแอปพลิเคชั่นอื่นบนแพลตฟอร์ม เพื่อตรวจจับการใช้งานเครือข่ายเซลลูล่าร์ ที่อาจก่อให้เกิดค่าใช้จ่ายแก่ผู้ใช้ ส่วนนี้ของงานวิจัยยังครอบคลุมถึงการศึกษารูปแบบการส่ง SMS โดยแอปพลิเคชั่น permission ที่จำเป็นในการทำงานกับ SMS และแนวทางการตรวจจับการส่งข้อความ SMS

แอปพลิเคชั่นสามารถส่งข้อความ SMS ได้ 2 วิธีคือ

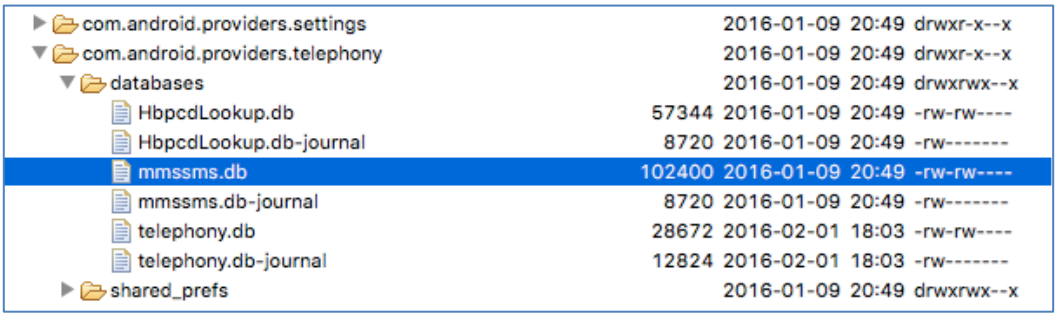
1) การส่งผ่าน Intent เพื่อให้ SMS application ของแพลตฟอร์มเป็นผู้ส่งให้ โดยจะเปิด SMS application พร้อมทั้งระบุหมายเลขปลายทางและข้อความไว้พร้อม เพื่อให้ผู้ใช้กดส่ง
2) การส่งผ่าน SmsManager ซึ่งรับผิดชอบการเตรียม SMS PDU และจัดส่ง SMS (Android-1, 2016)

ทั้งนี้ ตั้งแต่ Android Version 4.4 (API Level 19) เป็นต้นมา แพลตฟอร์มแอนดรอยด์รองรับให้แอปพลิเคชั่นใดๆ สามารถร้องขอให้ตนเองเป็น Default SMS Application แทนที่ SMS application ที่มาพร้อมกับแพลตฟอร์มได้ ดังที่จะนำเสนอในหัวข้อถัดไป

ทั้งนี้แพลต์ฟอร์มแอนดรอยด์ มีกลไกควบคุมสิทธิ์การส่ง SMS ทั้งตั้งแต่ขั้นตอนการติดตั้งแอปพลิเคชั่น (install time) และในระหว่างการทำงานของแอปพลิเคชั่น (runtime) ดังอธิบายในข้อถัดไป

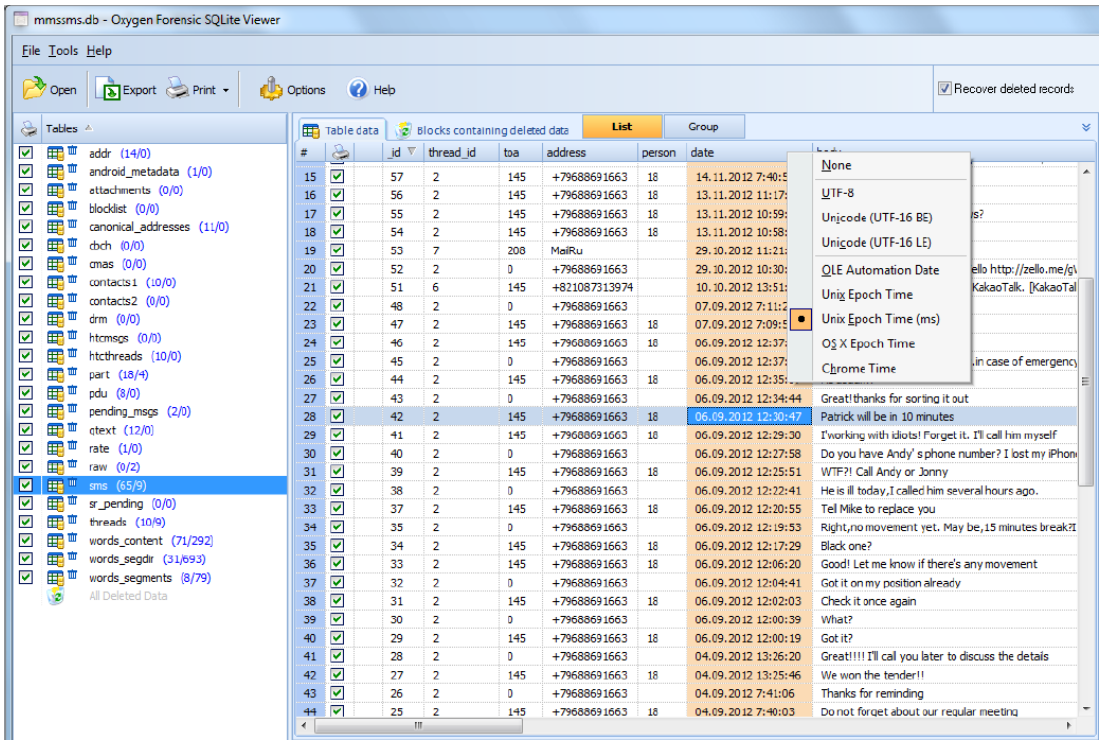
5.2.1 การบริหารจัดการ SMS โดยแพลตฟอร์มระบบปฏิบัติการแอนดรอยด์

ทุกครั้งที่มีการรับหรือส่ง SMS แพลต์ฟอร์มแอนดรอยด์จะทำการบันทึกข้อมูลเอาไว้ในฐานข้อมูล SQLite Database ที่ /data/data/com.android.providers.telephony/databases/mmssms.db ซึ่งสังเกตุได้ดังภาพที่ 19 ซึ่งไฟล์นี้เป็นแหล่งข้อมูล (Data Source) ให้กับ SMS Provider ซึ่งเป็นองค์ประกอบที่ทำหน้าที่เก็บและแบ่งปันข้อมูลให้กับแอปพลิเคชั่นและองค์ประกอบอื่นๆบนแพลตฟอร์มแอนดรอยด์



ภาพที่ 19 ไฟล์ mmssms.db

ไฟล์ mmssms.db บรรจุข้อมูลต่างๆของ SMS ได้แก่ หมายเลขปลายทาง (address), ข้อความ, วันเวลาการส่ง, ID ของ Thread ที่ส่ง เป็นต้น ดังภาพที่ 20 เมื่อใดก็ตามที่ผู้ใช้ลบ SMS ข้อมูลเรคคอร์ดนั้นๆจะถูกย้ายไปที่ /root/data/com.android.providers.telephony/databases/ ซึ่งแอปพลิเคชั่นทั่วไปไม่มีสิทธิ์การอ่านไดเร็กทอรี่ root นี้ ยกเว้นแต่เมื่ออุปกรณ์ผ่านการรูท (root) แล้ว



ภาพที่ 20 ตัวอย่างข้อมูลในไฟล์ mmssms.db

สิทธิ์การทำงานกับ SMS

แอปพลิเคชั่นต้องร้องขอ permission โดยระบุการร้องขอลงใน AndroidManifest.xml สำหรับการทำงานรูปแบบต่างๆกับ SMS ดังต่อไปนี้

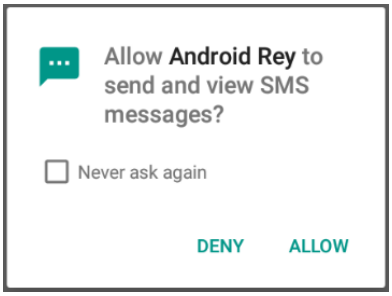
Permission	Protection Level	Description
SEND_SMS	dangerous	permission ในการส่ง SMS ผ่านวิธีใดๆ ดังที่กล่าวไว้ในหัวข้อแรก
RECEIVE_SMS	dangerous	permission ในการรับ SMS ขาเข้าที่ส่งเข้ามายังอุปกรณ์
READ_SMS	dangerous	permission ในการอ่านข้อความ SMS จาก SMS Provider

Runtime (Dynamic) Permission

ในแพลตฟอร์มแอนดรอยด์ดั้งเดิม ผู้ใช้ให้สิทธิ์ (grant permission) แก่แอปพลิเคชั่นตั้งแต่ขั้นตอนการติดตั้งแอปพลิเคชั่น แต่ตั้งแต่ Android 6.0 (API 23) เป็นต้นมา การขอสิทธิ์ (permission) ระดับ dangerous กับแอปพลิเคชั่น จะกระทำในระหว่างการทำงานของแอปพลิเคชั่น (runtime) (Android-2, 2016) เพื่อให้ขั้นตอนการติดตั้งยืดหยุ่นขึ้น กล่าวคือ ผู้ใช้ไม่จำเป็นต้องให้ทุกสิทธิ์การทำงานแก่แอปพลิเคชั่นก็

สามารถติดตั้งแอปพลิเคชั่นได้ นอกจากนั้นผู้ใช้ยังสามารถยกเลิก (revoke) การให้สิทธิ์การทำงานอย่างใดอย่างหนึ่งของแอปพลิเคชั่นได้ตลอดเวลา โดยการปรับการตั้งค่า (settings)

ดังนั้น ตั้งแต่ Android 6.0 เป็นต้นไป ผู้ใช้จะได้รับความปลอดภัยมายิ่งขึ้น เนื่องจากจะได้รับการเตือนในระหว่าง runtime ก่อนที่แอปพลิเคชั่นจะขอส่ง SMS ในครั้งแรก ดังตัวอย่างในภาพที่ 21



ภาพที่ 21 แอปพลิเคชั่นขอ permission ในการส่งและอ่าน SMS ในขณะ Runtime

Source: http://www.androidrey.com/run-time-permission-request-in-marshmallow/

จาก permission ที่กล่าวมาข้นต้น จะเห็นได้ว่าแอปพลิเคชั่นไม่มีสิทธิ์ในการเขียน SMS Provider แต่ประการใด ทำให้การส่ง SMS ของแอปพลิเคชั่นจะยังคงทิ้งร่องรอยข้อมูลเอาไว้ใน Outbox ของ SMS Provider แต่อย่างไรก็ตาม แพลตฟอร์มแอนดรอยด์ยังมี permission WRITE_SMS ซึ่งเป็น permission ซ่อน (hidden API) ที่อนุญาตให้แอปพลิเคชั่นเขียนลงใน SMS Provider ได้ กล่าวคือ แอปพลิเคชั่นอาจลบร่องรอยการส่ง SMS ขาออก ออกจาก SMS Provider ได้

Dangerous Hidden API

Permission WRITE_SMS อันเป็น permission ซ่อน (ไม่ระบุอยู่ในเอกสารสำหรับนักพัฒนา) นั้นถือว่าเป็น permission อันตรายที่อาจถูกใช้โดยแอปพลิเคชั่นอันตราย (malicious application) ได้ เนื่องจากแอปพลิเคชั่นอาจลบร่องรอยการส่ง SMS ขาออก ออกจาก SMS Provider ได้ อย่างไรก็ตาม จากการทดสอบพบว่า แอปพลิเคชั่น Anti-Virus เช่น Avast! (Avast Software, 2016) สามารถตรวจจับการขอ permission WRITE_SMS ของแอปพลิเคชั่นได้ และแจ้งเตือนในระหว่างการติดตั้งแอปพลิเคชั่น ว่าแอปพลิเคชั่นดังกล่าวเป็นแอปพลิเคชั่นต้องสงสัย

Default SMS Application

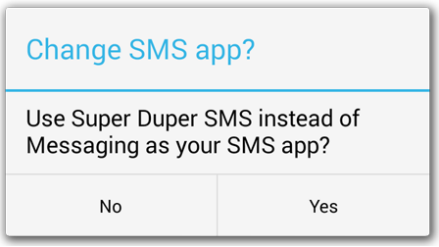
ตั้งแต่ Android 4.4 (API Level 19) เป็นต้นมา นักพัฒนาสามารถสร้างแอปพลิเคชั่นขึ้นมาเป็น Default SMS Application ทดแทน SMS Application ของแพลตฟอร์มแอนดรอยด์ได้ (Scott Main, David Braun, 2013) โดยแอปพลิเคชั่นดังกล่าวจะสามารถอ่านและเขียนลง SMS Provider ได้ โดยไม่ต้องใช้ Hidden API ที่นำเสนอในหัวข้อที่ผ่านมา ทั้งนี้ แอปพลิเคชั่นที่ต้องการเป็น default SMS application จะต้องขอ permission เพิ่มดังนี้

Permission	Protection Level	Description
BROADCAST_SMS	dangerous	permission ในการรับ SMS ขาเข้าโดยตรง และส่ง notification ไปแจ้งแอปพลิเคชั่นอื่นๆ
BROADCAST_WAP_PUSH	dangerous	permission ในการรับ MMS ขาเข้าโดยตรง และส่ง notification ไปแจ้งแอปพลิเคชั่นอื่นๆ

นอกจากนั้นแล้วแอปพลิเคชั่นยังต้องบรรจุ Broadcast Receiver ที่คอยตรวจจับ Intent ที่มี action เป็น SMS_DELIVER_ACTION เพื่อรับ SMS ขาเข้าโดยตรง และ Broadcast Receiver ที่คอยตรวจจับ Intent ที่มี action เป็น WAP_PUSH_DELIVER_ACTION เพื่อรับ MMS ขาเข้าโดยตรง

แอปพลิเคชั่นที่รองรับการตรวจจับ Intent ที่มี action เป็น SMS_DELIVER_ACTION เพื่อรับ SMS ขาเข้าโดยตรงนี้ จะมีความสามารถมากกว่าแอปพลิเคชั่นที่ได้รับ permission SEND_SMS ในหัวข้อที่ผ่านมา ตรงที่สามารถเขียนลง SMS Provider ได้ด้วย

อย่างไรก็ตาม แอปพลิเคชั่นจะสามารถเป็น Default SMS Application ได้ จะต้องได้รับการอนุญาตจากผู้ใช้ โดยผู้ใช้จะได้รับข้อความดังภาพที่ 22 นอกจากนั้นแล้ว แอปพลิเคชั่นอื่นๆ สามารถตรวจสอบ Default SMS Application ปัจจุบันได้ผ่านคำสั่ง Telephony.Sms.getDefaultSmsPackage()



ภาพที่ 22 ตัวอย่างแอปพลิเคชั่นขอเป็น Default SMS Application

Source: Scott Main, David Braun (2013)

5.3 การตรวจจับพฤติกรรมที่ผิดปกติของการรับส่งข้อมูลผ่านเครือข่าย

การตรวจจับพฤติกรรมที่ผิดปกติของแอปพลิเคชั่นขณะใช้งานบนอุปกรณ์จึงเป็นเรื่องจำเป็น เช่น ในช่วงห้าวันแรกหลังจากติดตั้งแอปพลิเคชั่นการอัพเดตเวอร์ชั่นเป็นการอัพเดตแบบปกติ แต่ในวันที่หกแอปพลิเคชั่นนี้มีพฤติกรรมอันตรายกลายเป็นมัลแวร์ด้วยการดาวน์โหลดโค้ดอันตรายมาที่อุปกรณ์ เป็นต้น พฤติกรรมที่ผิดปกติบนอุปกรณ์มือถือจะเป็นลักษณะของการรับส่งข้อมูลผ่านเครือข่ายที่ผิดสังเกต

การแยะแยะแอปพลิเคชั่นที่มีพฤติกรรมน่าสงสัยจากแอปพลิเคชั่นปกติทั่วไปจึงสามารถทำได้โดยใช้การเฝ้าระวังด้วยการเก็บข้อมูลการรับส่งข้อมูลผ่านเครือข่าย รวมถึงข้อมูลจราจรบนเครือข่าย ได้แก่

1) ปริมาณข้อมูลที่รับส่งผ่าน WiFi (Data usage over WiFi)
2) ปริมาณข้อมูลที่รับส่งผ่าน Cellular (Data usage over 3G/4G)

ตรวจสอบขณะที่แอปพลิเคชั่นทำงานบนอุปกรณ์ เพื่อตรวจจับพฤติกรรมของแอปพลิเคชั่นที่เปลี่ยนแปลงจากเดิมอย่างมีนัยสำคัญและแจ้งให้ผู้ใช้งานทราบถึงภัยที่กำลังคุกคาม โดยย่อการเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายบนโทรศัทพ์เคลื่อนที่มีขั้นตอนดังนี้ (ดังแสดงในภาพที่ 2)

1) การเฝ้าสังเกตพฤติกรรมการรับส่งข้อมูลผ่านอินเตอร์เน็ต
2) การตรวจพบพฤติกรรมการรับส่งข้อมูลผ่านอินเตอร์เน็ตที่ผิดปกติ โดยเปรียบเทียบกับรูปแบบมาตรฐานว่าแตกต่างออกไปอย่างมีนัยสำคัญทางสถิติ และใช้เทคนิคการเรียนรู้ของเครื่องในการแยกแยะพฤติกรรมที่ผิดปกติ
3) การแจ้งเตือนผู้ใช้ถึงความผิดปกติที่เกิดขึ้น โดยผู้ใช้สามารถสั่งระงับหรือหยุดการทำงานของแอปพลิเคชั่น และลบแอปพลิเคชั่นออกจากอุปกรณ์ (ถอนการติดตั้ง)

ข้อมูลการใช้เครือข่ายของแต่ละแอปพลิเคชั่นต้องนำมาคำนวณเป็นฟีเจอร์ หรือ ขั้นตอนการคำนวณค่าทางสถิติของข้อมูลจราจรบนเครือข่าย ค่าทางสถิติที่ได้จะสามารถสื่อความหมายและสะท้อนถึงลักษณะของพฤติกรรมการใช้งานได้ ตัวอย่างการคำนวณที่นำมาใช้ เป็นการหาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน ค่าน้อยสุด ค่ามากสุด เปอร์เซ็นต์ไทล์ เป็นต้น ด้านล่างเป็นฟีเจอร์ที่ได้จากการคำนวณทางสถิติและนำมาใช้ในการสร้างโมเดลการเรียนรู้

1) ค่าเฉลี่ย, ค่าเบี่ยงเบนมาตรฐาน, ค่าน้อยสุด และมากสุดของการรับ/ส่งข้อมูล
2) ค่าเฉลี่ย, ค่าเบี่ยงเบนมาตรฐาน, ค่าน้อยสุด และมากสุดของการรับ/ส่งข้อมูลที่เป็นอัตราส่วนของการรับส่งข้อมูลทั้งหมด
3) ร้อยละของการรับ/ส่งข้อมูล (ไบต์)

4) สถานะของเครือข่าย (Network state) – Cellular, Wifi ไม่มีสัญญาณ หรือแบบผสมผสาน สถานะผสมผสานหมายถึงกรณีที่การเชื่อมต่อเครือข่ายมีหลายสถานะ (เช่น ทั้ง cellular และ Wifi) ในช่วงเวลาที่เก็บข้อมูล
5) สถานะของแอปพลิเคชั่นแบบที่ 1 – foreground, background หรือ แบบผสมผสาน สถานะแบบผสมผสานหมายถึงกรณีที่มีหลายสถานะเกิดในช่วงเวลาที่เก็บข้อมูล

โมเดลการเรียนรู้ใช้หลักของ Classification Techniques มาช่วยในการแยกแยะแอปพลิเคชั่นน่าสงสัย เช่น การใช้ Support Vector Machine (SVM) หรือ Local Outlier Factor (LOF) เป็นต้น ในที่นี้เราเลือกใช้ SVM ซึ่งให้ค่าความแม่นยำสูงกว่า

บรรณานุกรม

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15..

Dai, Shuaifu, et al. "Behavior-based malware detection on mobile phone." Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on. IEEE, 2010.

Dini, Gianluca, et al. "MADAM: A Multi-level Anomaly Detector for Android Malware." MMM-ACNS. Vol. 12. 2012.

Garcia-Teodoro, Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security 28.1 (2009): 18-28.

Android-1 (2016), SMSManager

จาก https://developer.android.com/reference/android/telephony/SmsManager.html

Android-2 (2016), Requesting Permissions at Run Time

จาก https://developer.android.com/training/permissions/requesting.html

Avast Software (2016), Avast! Anti-Virus for Android จาก https://www.avast.com

Scott Main and David Braun (2013), Getting Your SMS Apps Ready for KitKat จาก http://android-developers.blogspot.com/2013/10/getting-your-sms-apps-ready-for-kitkat.html

บทที่ 6
การพัฒนาและผลการทดสอบโมดูลตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์มือถือโดยพัฒนา APIs จาก Android Software Development Kit (SKD)

การตรวจจับแอปพลิเคชั่นอันตรายแบ่งเป็น 2 กรณี (กล่าวโดยละเอียดในบทที่ 2) คือ กรณีที่ 1) ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์ และกรณีที่ 2) หลังจากที่ได้ติดตั้ง mCOP App ระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายที่แฝงตัวมากับแอปพลิเคชั่นปกติ

เพื่อให้สอดคล้องกับทั้งสองกรณี อัลกอริทึมในการตรวจจับแอปพลิเคชั่นปลอมของระบบแบ่งออกเป็น 3 ส่วนหลัก ได้แก่ (1) การตรวจจับแอปพลิเคชั่นบนเครื่องโดยใช้ฐานข้อมูลที่มีอยู่ (2) การตรวจจับการใช้งานที่ผิดปกติของ SMS และ (3) การประเมินพฤติกรรมของแต่ละแอปพลิเคชั่น เพื่อระบุแนวโน้มที่แต่ละแอปพลิเคชั่น จะกลายเป็นแอปพลิเคชั่นอันตราย

สำหรับส่วนแรก การตรวจจับ แอปพลิเคชั่น บนเครื่องกับฐานข้อมูลที่มีอยู่เพื่อให้ผู้ใช้มั่นใจว่าทุกแอปพลิเคชั่นบนเครื่องเป็นแอปพลิเคชั่นปกติไม่ใช่มัลแวร์ หลังจากนั้นส่วนที่ 2-3 เป็นการเฝ้าระวังว่าแอปพลิเคชั่นใดที่มีพฤติกรรมที่น่าสงสัย (แอปพลิเคชั่นปลอม) เพื่อตรวจจับและเตือนภัยให้กับผู้ใช้งาน

6.1 การพัฒนาและทดสอบการตรวจจับแอปพลิเคชั่นอันตรายโดยใช้ฐานข้อมูล

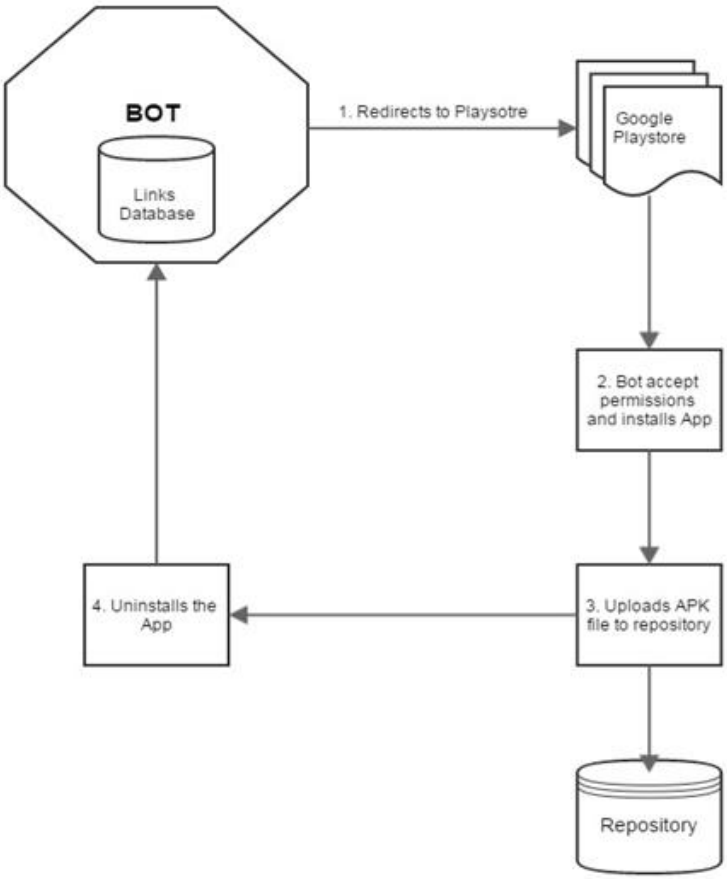
6.1.1 การสร้างฐานข้อมูลแอปพลิเคชั่นอันตราย

เพื่อเป็นการเพิ่มประสิทธิ์ภาพของการตรวจสอบระดับความเสี่ยง โครงงานนี้ได้สร้างฐานข้อมูลเพื่อใช้ในการวิเคราะห์ระดับความเสี่ยง โดยการรวบรวมแอปพลิเคชันใน market places ต่างๆ เช่น Google Play Store และ APK DL เป็นต้น การรวบรวมแอปพลิเคชันที่มีใน marlet places จำนวนมากสามารถได้โดยการสร้าง Bot ที่สามารถดาวน์โหลดแอปพลิเคชันมาเก็บไว้ก่อนแบบอัตโนมัส

ขั้นตอนการสร้าง Bot ประกอบด้วย 2 ส่วนหลักคือ

1) การใช้ web crawler (Scrapy) คือบอตอินเทอร์เน็ตที่ทำงานท่องไปบน Market Place ที่ต้องการ โดยมีจุดประสงค์เพื่อทำการรวบรวมลิงค์และข้อมูลของแอปพลิเคชัน

2) การสร้างบอตเพื่ออ่านไฟล์จากข้อที่ 1. และทำการดานว์โหลดแอปพลิเคชันมาเก็บไว้ที่เครื่อง ดังแสดงดารทำงานในภาพที่ 00



ภาพที่ 23 หลักการใช้ Web Crawler

```
SELECT TOP 1000 [md5]
      ,[sha1]
      ,[sha256]
      ,[detection_ratio]
      ,[detection_percentage]
      ,[scan_result]
      ,[scan_dt]
  FROM [BUAntivirusStats].[dbo].[api_report]
```

100 %

Results | Messages

	md5	sha1	sha256	detection_ratio	detection_percentage	scan_result	scan_dt
1	2d3e5ba01f1fcb50105...	19acafbee96bd210c0...	859a0ec1efc31dd3acfc6450...	39/56	79.29	Symantec:Android.Pjapps!Av...	2016-09-05 23:39:41.890
2	2d41dd9a8dd15d4e97...	972f85cec1aca9ee40...	f9448c4606f0eb5734aaf387...	36/52	81.35	Symantec:Android.Pjapps!Av...	2016-09-05 23:40:05.323
3	2ddcca2a490932c078...	2400cd4af2e7281b08...	7d71afe40f733eba98a9205b...	38/56	83.39	Symantec:Android.Pjapps!Av...	2016-09-05 23:40:11.877
4	3a24a80db4d4bacd3c...	6fc6f96ffb3aec1917ff...	bfa625cf6d6a19a3a8f474e6...	41/56	88.75	Symantec:Android.Pjapps!Av...	2016-09-05 23:41:05.273
5	3a7524115ffeed82505...	04d24624e98e7fc75f...	2a7d4aed796bbe92dce244b...	40/56	84.82	Symantec:Android.Pjapps!Av...	2016-09-05 23:41:25.473
6	2e65416b14725a8dd4...	436d08b5d607fd398a...	9896f337dd2114c98b0df64a...	40/56	84.82	McAfee:Artemis!2E65416B1...	2016-09-05 23:44:07.150
7	2ebda0d3bad1c19c9c...	9c61d5bd80a9c4514...	ecf0816685f8c00c6054bbd8...	39/56	84.11	McAfee:Artemis!2E65416B1...	2016-09-05 23:44:32.680
8	2ff6eb839fa275976b4b...	e28fedd06e1805fa7e...	863f25f36ea599a7f56c4bd5...	41/56	85.54	Symantec:Android.Adrd!Avas...	2016-09-05 23:45:59.737
9	3b9fd10de07bddc945f...	11be30ac4e33db99c...	419f5d6b2db869e9e405832...	39/56	84.11	Symantec:Android.Adrd!Avas...	2016-09-05 23:46:05.030
10	3cf8317a46b05daff42f...	781af82765a2716b74...	187485101d040737c24fbf30...	41/56	88.75	Symantec:Android.Adrd!Avas...	2016-09-05 23:46:21.960
11	05ae244c85f188ec6f4...	33534b5610c966388...	f8ad30f42a163036b4506415...	41/55	87.09	McAfee:Artemis!05AE244C8...	2016-09-05 23:49:21.300
12	4a5e1fc3535cf5535f93...	0ab00155ee61e961c...	981c519da5b40df4546e7ca...	39/55	85.64	McAfee:Artemis!4A5E1FC35...	2016-09-05 23:53:36.817
13	6aa2a914b6b60c4378...	7678793631b2453b9...	3c51b6745c5e27788b3acaa...	38/56	83.39	Symantec:Android.Pjapps!Av...	2016-09-05 23:54:03.493
14	01f5ad8046ff54b94997...	b8d7a8e94a7338d9b...	0c49a873f66c286996debf6c...	39/56	84.11	McAfee:Artemis!01F5AD804...	2016-09-05 23:34:32.610
15	0aa7ac27d5ab7c937e...	23cbb6aa61b2ff1959...	a3b0e6f2557921ab39729eb...	40/55	86.36	McAfee:Artemis!01F5AD804...	2016-09-05 23:35:02.380
16	0d8fb22653caec013f4...	c912f95e68047d0765...	42f5549d638243926135a93...	41/56	85.54	McAfee:Artemis!01F5AD804...	2016-09-05 23:35:27.807

ภาพที่ 24 ฐานข้อมูลแอปพลิเคชั่นที่เป็นอันตรายและแฮชโค้ด

6.1.2 ฐานข้อมูลแอปพลิเคชั่นอันตรายที่ได้

แอปพลิเคชันที่เป็นอันตรายที่สามารถรวบรวมได้มีทั้งหมด 2,000 โค้ดที่แตกต่างกัน (แสงดในภาพที่ 24) และต่างประเภทกันประเภทของโค้ดอันตรายสามารถจัดได้ดังแสดงในภาคผนวก ข ซึ่งมีคำอธิบายถึงพฤติกรรมของโค้ดอันตรายแต่ละตระกูลซึ่งใช้ยุทธวิธีในการเจาะเข้าระบบรักษาความปลอดภัยที่แตกต่างกัน โดยที่แอปพลิเคชันอันตรายจำนวน 155 แอปพลิเคชันสามารถระบุตระกูลได้ (ภาคผนวก ข.)

6.2 การพัฒนาและทดสอบการตรวจจับการส่งข้อความ SMS

6.2.1 การพัฒนาส่วนการตรวจจับการส่งข้อความ SMS

การส่ง SMS ขาออก ไม่ว่าจะเป็นการส่งผ่านวิธีใดๆ ก็ตาม จะก่อให้เกิดการเขียนข้อมูลลง Outbox ใน SMS Provider ดังนั้นวิธีที่สะดวกวิธีหนึ่งในการตรวจจับการส่ง SMS คือการอ่านข้อมูลจาก SMS Provider โดยข้อมูลที่อ่านได้จะประกอบด้วย

- ข้อมูลแอปพลิเคชั่นผู้ส่ง SMS ได้แก่ Package Name, Application ID และ Thread ID
- ข้อมูล address (เช่น เบอร์โทรศัพท์) ปลายทาง
- ข้อมูลเนื้อหา SMS และเวลาที่ส่งออก

ทั้งนี้ การอ่านข้อมูลเรคคอร์ดล่าสุดใน Outbox ของ SMS Provider ทำได้โดยเข้าถึง content URI content://sms/sent แล้วอ่านเรคคอร์ดแรก ดังนี้ตัวอย่างโค้ดในภาพที่ 25

```
private SMS readFromOutgoingSMS(Context context) {
   Cursor cursor = context.getContentResolver().query(
         Uri.parse("content://sms/sent"), null, null, null, null);
   SMS sms = null;
   if (cursor.moveToNext()) {
      String protocol =
cursor.getString(cursor.getColumnIndex("protocol"));
      int type = cursor.getInt(cursor.getColumnIndex("type"));

      // Only processing outgoing sms event & only when it
      // is sent successfully (available in SENT box).
      if (protocol != null || type != MESSAGE_TYPE_SENT) {
         return sms;
      }

      // Get Column Index
      int bodyColumn = cursor.getColumnIndex("body");
      int addressColumn = cursor.getColumnIndex("address");
      int thread_id = cursor.getColumnIndex("thread_id");
      int creator = cursor.getColumnIndex("creator");
      int seen = cursor.getColumnIndex("seen");

      // Get Value
      String str = "SMS From: " + cursor.getString(addressColumn) +
            "\n" + cursor.getString(bodyColumn) +
            "\ncreator : " + cursor.getString(creator) +
            "\nthread_id : " + cursor.getString(thread_id) +
            "\napp_id : " + cursor.getString(app_id) +
            "\nseen : " + cursor.getString(seen) +
            "\n";

      Log.d("Lattapol", str);

      String from = "0";
      String to = cursor.getString(addressColumn);
      Date now = new Date(cursor.getLong(dateColumn));
      String message = cursor.getString(bodyColumn);
      sms = new SMS(from, to, message, now);
   }
   cursor.close();
   return sms;
}
```

ภาพที่ 25 เมธอดที่ใช้ในการอ่านข้อมูล SMS เรคคอร์ดล่าสุดใน SMS Provider

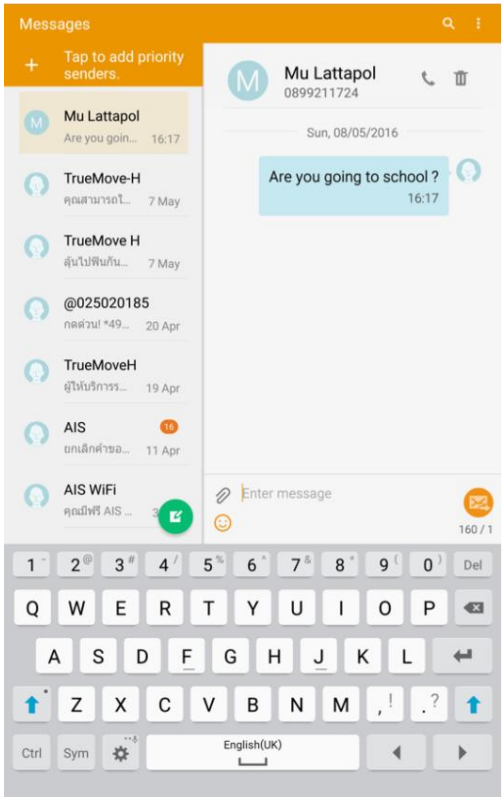
นอกจากนั้น การตรวจจับ SMS ขาออกแบบ Real-Time จำเป็นต้องสร้าง Content Observer ซึ่งคอยฟังและได้รับ callbacks ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูล ระบบ mCop ได้สร้าง Content Observer ไว้ที่ Content URI content://sms ดังภาพที่ 26 โดยจะเห็นได้ว่าระบบสามารถตรวจจับได้ และตรวจสอบเพื่อหา SMS ขาออก ทุกครั้งที่มีการเปลี่ยนแปลงของ Content ของ SMS Provider ได้

```
private void registerContentObserver(final AndroidEvent dc) {
   if (observer != null) {
      return;
   }
   final Context context = dc.getContext();
   observer = new ContentObserver(null) {
      public void onChange(boolean selfChange) {
         SMS sms = readFromOutgoingSMS(context);
         if (sms != null) {
            getReporter().report(dc, sms);
         }
      }
   };
   context.getContentResolver().registerContentObserver(
      Uri.parse(CONTENT_SMS), true, observer);
}
```

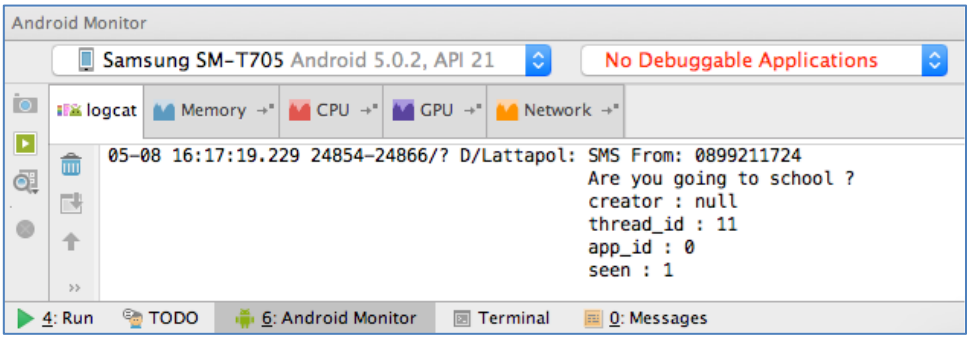
ภาพที่ 26 เมธอดที่ลงทะเบียน Content Observer เพื่อฟังการเปลี่ยนแปลงของเนื้อหาใน SMS Provider

6.2.2 ผลการตรวจสอบการส่ง SMS ขาออก

โดยปกติแล้ว หาก SMS ขาออกเป็น SMS ที่ส่งโดยผู้ใช้ ดังภาพที่ 27 แอปพลิเคชั่นตรวจจับ SMS ขาออกจะได้ข้อมูล SMS ที่ถูกส่งออกไปทันที ดังภาพที่ 28

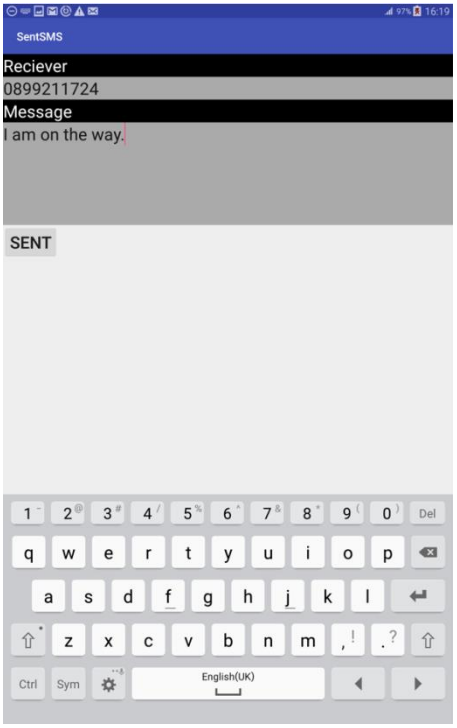


ภาพที่ 27 การส่ง SMS ขาออกโดยผู้ใช้

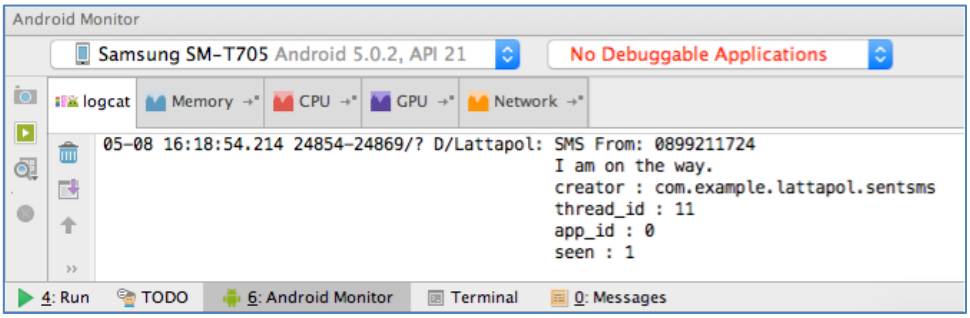


ภาพที่ 28 ข้อมูล SMS ขาออกที่ส่งโดยผู้ใช้ ซึ่งแอปพลิเคชั่นตรวจจับได้ทันที

จากภาพที่ 28. จะเห็นได้ว่า SMS ที่ส่งออกโดยผู้ใช้ จะมีค่าในคอลัมน์ creator เป็น null และหากเราจำลองการส่ง SMS ด้วยแอปพลิเคชั่น ดังภาพที่ 29 เราจะได้ข้อมูล SMS ขาออกแบบ real-time เช่นกัน แต่ในทางตรงกันข้าม ค่าในคอลัมน์ creator เป็น package name ของแอปพลิเคชั่นที่ส่งออก ดังภาพที่ 30



ภาพที่ 29 การส่ง SMS ขาออกโดยแอปพลิเคชั่น



ภาพที่ 30 ข้อมูล SMS ขาออกที่ส่งโดยแอปพลิเคชั่น ซึ่งแอปพลิเคชั่นตรวจจับของเราสามารถตรวจจับได้ทันที

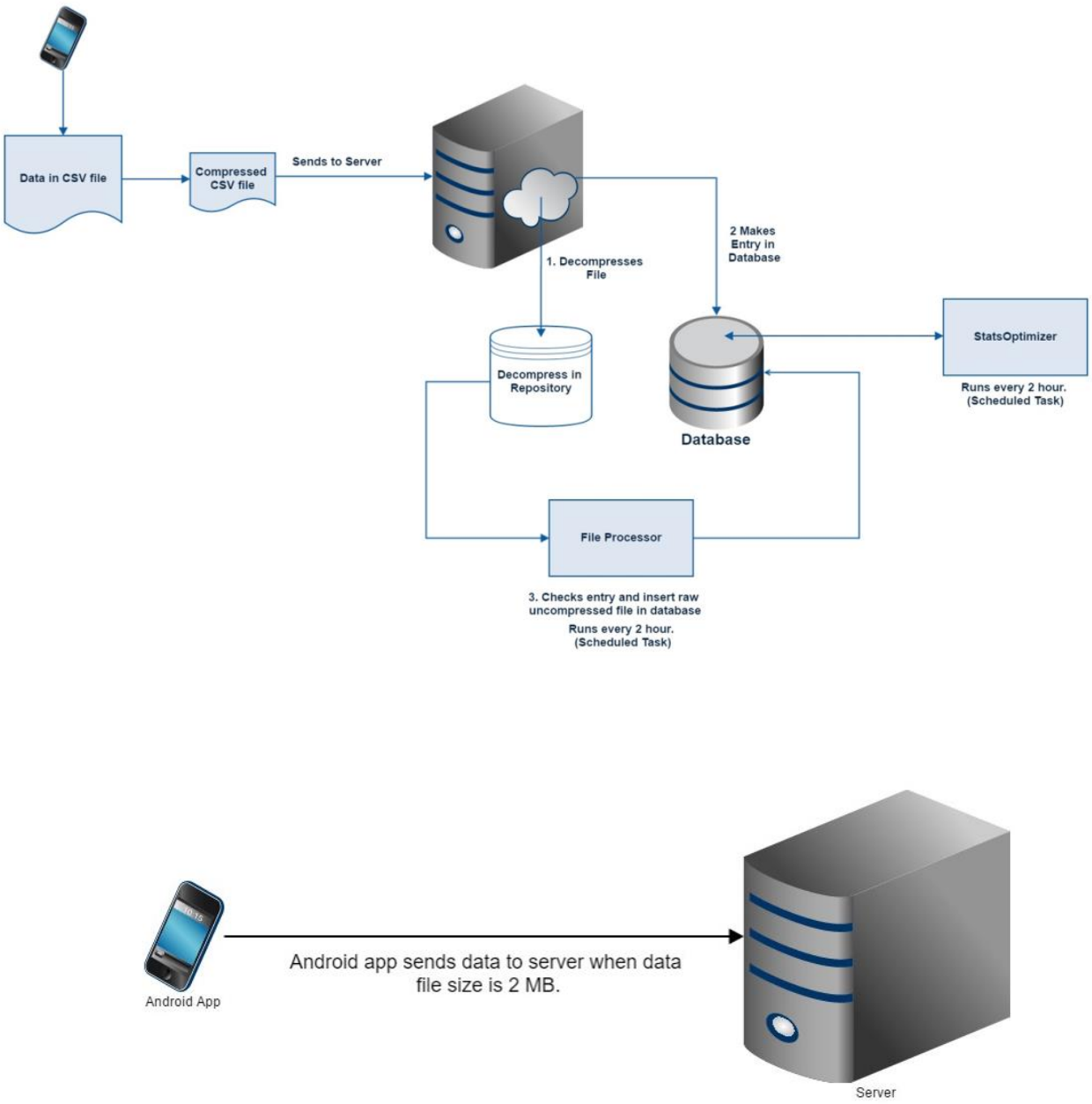
นอกจากการตรวจจับ SMS ขาออกแล้ว ระบบยังสามารถตรวจสอบสิทธิ์การจัดการ SMS ของแอปพลิเคชั่นอื่นได้ด้วย โดยแบ่งตามกรณีได้ดังนี้

1. ตรวจวิเคราะห์ หากแอปพลิเคชั่นใด ร้องขอ permission WRITE_SMS เราอาจตั้งข้อสงสัยการแอบเขียน SMS Provider ผ่าน Hidden API ดังนั้น อาจระบุไว้ว่าเป็นแอปพลิเคชั่นที่อาจเป็นอันตราย โดยแนวทางนี้คล้ายกับการตรวจจับของแอปพลิเคชั่น Anti-Virus
2. ตรวจวิเคราะห์ว่ามีการเปลี่ยนแปลง Default SMS Application ซึ่งจะทำให้แอปพลิเคชั่นดังกล่าวสามารถเขียนลง SMS Provider ได้ ทั้งนี้ สามารถดึงข้อมูล Default SMS Application ได้ผ่านคำสั่ง Telephony.Sms.getDefaultSmsPackage()

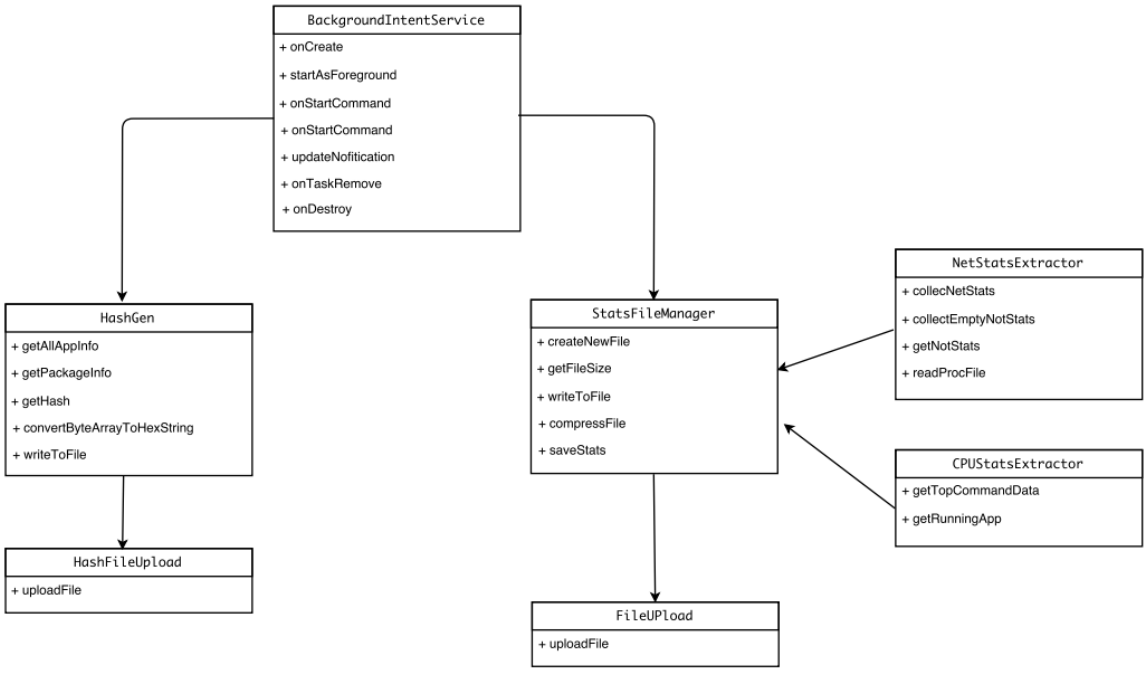
อย่างไรก็ตาม แม้ว่า Default SMS Application จะสามารถก่อให้เกิดการเปลี่ยนแปลงของเนื้อหา SMS Provider ได้ ระบบก็สามารถตรวจจับการเปลี่ยนแปลง (ทั้งใส่ข้อมูล และ (แอบ) ลบข้อมูล) ได้ ผ่านการใช้งาน Content Observer

6.3 การพัฒนาและทดสอบการตรวจจับพฤติกรรมที่ผิดปกติของการรับส่งข้อมูลผ่านเครือข่าย

โมดูลการรวบรวมข้อมูลการใช้งานนั้นถือเป็นหัวใจหลักของแอปพลิเคชั่น mCOP โดยมีเฟรมเวิร์คตามภาพที่ 31 โดยแอปพลิเคชั่น mCOP จะรวบรวมข้อมูลการใช้งานทุกๆหน่วยเวลา (ค่า default คือ 5 วินาที) บันทึกไว้ในหน่วยความจำของเครื่องมือถือจนครบ เมื่อไฟล์ที่บันทึกถึงขนาด 2 MB ไฟล์ดังกล่าวจะถูกบีบอัดเพื่อลดขนาดไฟล์ก่อนส่งไปยังเซิร์ฟเวอร์ ทางฝั่งเซิร์ฟเวอร์จะทำการแตกไฟล์และบันทึกลงฐานข้อมูล ภาพที่ 32 แสดงคลาสไดอะแกรมของโมดูลการรวบรวมข้อมูลการใช้งาน



ภาพที่ 31 เฟรมเวิร์คของการส่งข้อมูลการรับส่งจากเครื่องผู้ใช้ไปเซิร์ฟเวอร์เพื่อวิเคราะห์พฤติกรรม



ภาพที่ 32 คลาสไดอะแกรมของการส่งข้อมูลการรับส่งจากเครื่องผู้ใช้ไปเซิร์ฟเวอร์เพื่อวิเคราะห์พฤติกรรม

6.3.1 การทดสอบโมดูลการรวบรวมข้อมูลการใช้งาน

ทางทีมพัฒนาได้ทำการทดสอบโมดูลการรวบรวมข้อมูลการใช้งานเรียกว่า “BU Stats collector” โดยแบ่งการทดสอบเป็น 1) ตรวจสอบความถูกต้องของการค้นหาและติดตั้ง BU-Stats Collector Application และ 2) ตรวจสอบการทำงานของ BU-Stats Collector Application ซึ่งวิธีและผลของการทดสอบเป็นตามตารางที่ 3-4

ตารางที่ 3 ตรวจสอบความถูกต้องของการค้นหาและติดตั้ง BU-Stats Collector Application

No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
1. ผู้ใช้งานทำการค้นหาและติดตั้ง Application BU-Stats Collector ผ่าน Browser (On PC)					
1.1	เปิด Web Browser บน PC เพื่อทำการทดสอบ	1) ไปที่ Url: www.google.com 2) ใส่ Keyword ในการค้นหาเป็น "BU-Stats Collector" 3) กดปุ่มค้นหา	พบ Link ของ BU-Stats Collector Application ในผลลัพธ์ของการค้นหา	PASS	
		ทำการกด Link ของ Application "BU-Stats Collector" จากหน้าผลลัพธ์การค้นหา	สามารถไปที่หน้าของ Application BU-Stats Collector บน Google play store ได้ถูกต้อง	PASS	
		ทำการติดตั้ง Application โดย 1) กดปุ่มติดตั้ง 2) ทำการเลือกอุปกรณ์ที่จะทำการติดตั้ง Application	Application ถูกติดตั้งลงอุปกรณ์ที่เลือกได้สำเร็จ	PASS	
1.2	เปิด Web Browser บน PC เพื่อทำการทดสอบ	ไปที่ Url: https://play.google.com/store/apps/details?id=th.ac.bu.science.mit.allappstatscollector	สามารถไปที่หน้าของ Application BU-Stats Collector ได้ถูกต้อง	PASS	
		ทำการติดตั้ง Application โดย 1) กดปุ่มติดตั้ง 2) ทำการเลือกอุปกรณ์ที่จะทำการติดตั้ง Application	Application ถูกติดตั้งลงอุปกรณ์ที่เลือกได้สำเร็จ	PASS	
2. ผู้ใช้งานทำการค้นหาและติดตั้ง Application BU-Stats Collector ผ่าน Browser (On Android Smartphone)					
2.1	เปิด Application Google Chrome Browser บน Smartphone Android เพื่อทำการทดสอบ	1) ไปที่ Url: www.google.com 2) ใส่ Keyword ในการค้นหาเป็น "BU-Stats Collector" 3) กดปุ่มค้นหา	พบ Link Google play store ของ Application BU-Stats Collector ในผลลัพธ์ของการค้นหา	PASS	
		1) ทำการกด Link ของ Application "BU-Stats Collector" จากหน้าผลลัพธ์การค้นหา 2) เลือก Application ที่ใช้ในการเปิด Link เป็น Google Chrome Browser	สามารถไปที่หน้าของ Application Bu บน Google play store ได้ถูกต้อง	PASS	

No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
		ทำการติดตั้ง Application โดย 1) กดปุ่มติดตั้ง 2) ทำการเลือกอุปกรณ์ที่จะทำการติดตั้ง Application	Application ถูกติดตั้งลงอุปกรณ์ที่เลือกได้สำเร็จ	PASS	
2.2		1) ไปที่ Url: www.google.com 2) ใส่ Keyword ในการค้นหาเป็น "BU-Stats Collector" 3) กดปุ่มค้นหา	พบ Link Google play store ของ Application BU-Stats Collector ในผลลัพธ์ของการค้นหา	PASS	
		1) ทำการกด Link ของ Application "BU-Stats Collector" จากหน้าผลลัพธ์การค้นหา 2) เลือก Application ที่ใช้ในการเปิด Link เป็น Google play store	สามารถไปที่หน้าของ Application Bu บน Google play store ได้ถูกต้อง	PASS	
		ทำการติดตั้ง Application โดย กดปุ่ม ติดตั้ง	Application ถูกติดตั้งลงอุปกรณ์ที่เลือกได้สำเร็จ	PASS	
2.3	เปิด Application Google Chrome Browser บน Smartphone Android เพื่อทำการทดสอบ	ไปที่ Url: https://play.google.com/store/apps/details?id=th.ac.bu.science.mit.allappstatscollector	สามารถไปที่หน้าของ Application Bu บน Google play store ได้ถูกต้อง	PASS	
		ทำการติดตั้ง Application โดย 1) กดปุ่มติดตั้ง 2) ทำการเลือกอุปกรณ์ที่จะทำการติดตั้ง Application	Application ถูกติดตั้งลงอุปกรณ์ที่เลือกได้สำเร็จ	PASS	
3. ผู้ใช้งานทำการค้นหาและติดตั้ง Application BU Antivirus ผ่าน Application Google Play Store (On Smartphone Android)					
3.1	เปิด Application Google Play Store เพื่อทำการทดสอบ	1) ทำการค้นหา Application Bu โดยใส่ Keyword "BU-Stats Collector" 2) กดปุ่ม ค้นหา	พบ Application BU-Stats Collector ในผลลัพธ์ของการค้นหา	PASS	
		กดเลือก Application BU-Stats Collector จากหน้าผลลัพธ์ของการค้นหา	สามารถไปที่หน้าของ Application BU-Stats Collector บน Google play store ได้ถูกต้อง	PASS	
		ทำการติดตั้ง Application โดย กดปุ่ม ติดตั้ง	Application BU-Stats Collector ถูกติดตั้งลงอุปกรณ์ได้สำเร็จ	PASS	

ตารางที่ 4 ตรวจสอบการทำงานของ BU-Stats Collector Application

No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
1. การทำงานของ Application BU-Stats Collector					
1.1	เปิดใช้งาน Application BU-Stats Collector	กดที่ไอคอนของ Application BU-Stats Collector - กดปุ่ม START COLLECTING STATS	เปิดใช้งาน Application BU-Stats Collector ได้สำเร็จ Application สามารถเก็บข้อมูลการใช้งานได้	PASS	
1.2	ทดสอบการทำงานของ Application BU-Stats Collector - การสะสมข้อมูลการใช้งาน CPU ของอุปกรณ์	1) ปิดการทำงานของ Mobile Data หรือ Wi-Fi 2) เปิด Application BU 3) กดปุ่ม Home เพื่อกลับสู่หน้าจอหลัก 4) เปิดใช้งาน Application (Games, Camera, Music Player, etc.) โดยเปิด มากกว่า 3 Application 5) กลับไปที่ Application BU-Stats Collector เพื่อตรวจสอบผล	Application BU-Stats Collector สามารถสะสมข้อมูลการใช้งาน CPU ของอุปกรณ์ได้สำเร็จ โดย - Status แสดงเป็น "Collecting data...." - File Size มีขนาดเพิ่มขึ้นทุก 1 วินาที	PASS	
1.3	ทดสอบการทำงานของ Application BU-Stats Collector - การสะสมข้อมูลการใช้งาน CPU และ Network ของอุปกรณ์	1) เปิดการทำงานของ Mobile Data หรือ Wi-Fi 2) เปิด Application BU-Stats Collector 3) กดปุ่ม Home เพื่อกลับสู่หน้าจอหลัก 4) เปิดใช้งาน Application ที่เชื่อมต่อ Internet ในการใช้งาน (Facebook, YouTube, Twitter, Browser, etc.)	Application Bu Antivirus สามารถสะสมข้อมูลการใช้งาน CPU และ Network ของอุปกรณ์ได้สำเร็จ โดย - Status แสดงเป็น "Collecting data...." - File Size ต้องมีขนาดเพิ่มขึ้นทุก 1 วินาที	PASS	
1.4		กดปุ่ม STOP COLLECTING STATS	Application Bu Antivirus หยุดการสะสมข้อมูลการใช้งาน CPU และ Network	PASS	

No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
1.5	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - ไม่มีการเชื่อมต่อกับอินเตอร์เน็ต (WI-FI, MOBILE DATA) - ขนาดของข้อมูลมีมากกว่า 2 MB	- ปิดการใช้งานของ Mobile Data หรือ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ไม่มีการส่งไฟล์ข้อมูลการใช้งานไปยัง Server	PASS	
1.6	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (Wi-Fi) - ขนาดของข้อมูลน้อยกว่า 2 MB	- เปิดการใช้งานของ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ไม่มีการส่งไฟล์ข้อมูลการใช้งานไปยัง Server	PASS	
1.7	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (Wi-Fi) - ขนาดของข้อมูลเท่ากับ 2 MB	- เปิดการใช้งานของ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ทำการ Compresses ไฟล์ และส่งข้อมูลไปยัง Server	PASS	
1.8	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (Wi-Fi) - ขนาดของข้อมูลมีมากกว่า 2 MB	- เปิดการใช้งานของ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ทำการแบ่งไฟล์ออกเป็นไฟล์ย่อยที่มีขนาด 2 MB ก่อน Compresses ไฟล์ และส่งข้อมูลไปยัง Server	PASS	

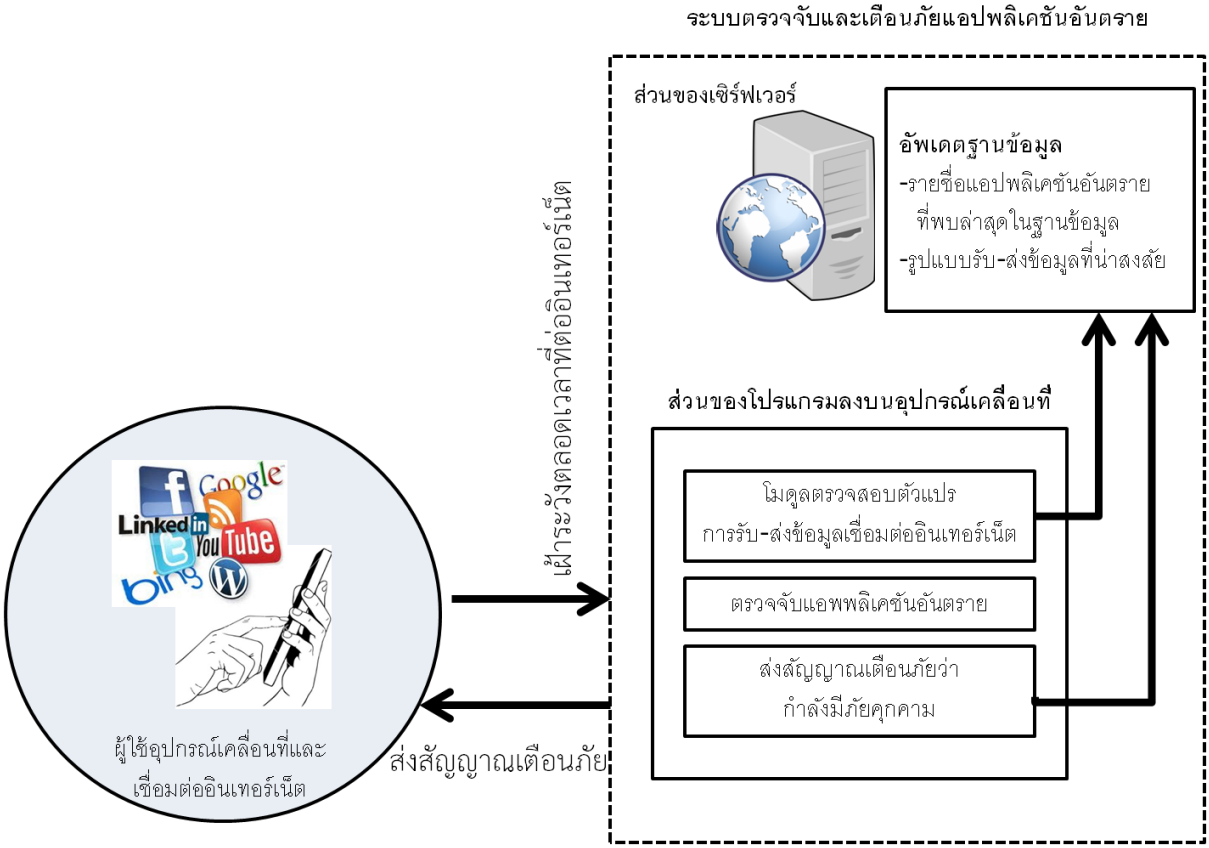
No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
1.9	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (Wi-Fi) แต่ในระหว่างการส่งไฟล์ การเชื่อมต่ออินเตอร์เน็ตสูญหาย - ขนาดของข้อมูลมากกว่าหรือเท่ากับ 2 MB	- เปิดการใช้งานของ Wi-Fi - ระหว่างที่ Application ทำการส่งไฟล์ ปิดการใช้งานของ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector หยุดทำการส่งไฟล์ไปยัง Server	PASS	
1.10	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (Wi-Fi) แต่ในระหว่างการส่งไฟล์ การเชื่อมต่ออินเตอร์เน็ตสูญหาย และกลับมาเชื่อมต่ออีกครั้ง - ขนาดของข้อมูลมากกว่าหรือเท่ากับ 2 MB	- เปิดการใช้งานของ Wi-Fi - ระหว่างที่ Application BU-Stats Collector ทำการส่งไฟล์ ปิดการใช้งานของ Wi-Fi เป็นเวลา 1 นาที - เปิดการใช้งานของ Wi-Fi - ตรวจสอบ Log การส่งข้อมูลของ App BU ไปยัง Server	Application BU-Stats Collector สามารถทำการส่งไฟล์ไปยัง Server ต่อเนื่องจากการส่งก่อนหน้าได้สำเร็จ	PASS	
1.11	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (MOBILE DATA) - ขนาดของข้อมูลน้อยกว่า 2 MB	- เปิดการใช้งานของ MOBILE DATA - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ไม่มีการส่งไฟล์ข้อมูลการใช้งานไปยัง Server	PASS	

No.	Test Steps	Input Data / Special Information	Expected Results/ Output Data	Pass / Failed (P/F)	Concerns
1.12	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (MOBILE DATA) - ขนาดของข้อมูลเท่ากับ 2 MB	- เปิดการใช้งานของ MOBILE DATA - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ทำการ Compresses ไฟล์ และส่งข้อมูลไปยัง Server	PASS	
1.13	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (MOBILE DATA) - ขนาดของข้อมูลมีมากกว่า 2 MB	- เปิดการใช้งานของ MOBILE DATA - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector ทำการแบ่งไฟล์ออกเป็นไฟล์ย่อยที่มีขนาด 2 MB ก่อน Compresses ไฟล์ และส่งข้อมูลไปยัง Server	PASS	
1.14	ทดสอบการส่งข้อมูลการใช้งาน ของ Application BU-Stats Collector - มีการเชื่อมต่อกับอินเตอร์เน็ต (MOBILE DATA) แต่ในระหว่างการส่งไฟล์ การเชื่อมต่ออินเตอร์เน็ตสูญหาย - ขนาดของข้อมูลมากกว่าหรือเท่ากับ 2 MB	- เปิดการใช้งานของ MOBILE DATA - ระหว่างที่ Application BU-Stats Collector ทำการส่งไฟล์ ปิดการใช้งานของ MOBILE DATA - ตรวจสอบ Log การส่งข้อมูลของ App BU ไปยัง Server	Application BU-Stats Collector หยุดทำการส่งไฟล์ไปยัง Server	PASS	
1.15	ทดสอบการส่งข้อมูลการใช้งาน - มีการเชื่อมต่อกับอินเตอร์เน็ต MOBILE DATA แต่ในระหว่างการส่งไฟล์ การเชื่อมต่ออินเตอร์เน็ตสูญหาย และกลับมาเชื่อมต่ออีกครั้ง - ขนาดของข้อมูลมากกว่าหรือเท่ากับ 2 MB	- เปิดการใช้งานของ MOBILE DATA - ระหว่างที่ Application BU-Stats Collector ทำการส่งไฟล์ ปิดการใช้งานของ MOBILE DATA เป็นเวลา 1 นาที - เปิดการใช้งานของ MOBILE DATA - ตรวจสอบ Log การส่งข้อมูลของ Application BU-Stats Collector ไปยัง Server	Application BU-Stats Collector สามารถทำการส่งไฟล์ไปยัง Server ต่อเนื่องจากการส่งก่อนหน้าได้สำเร็จ	PASS	

บทที่ 7
การประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายในห้องปฏิบัติการ

ระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายประกอบด้วยโมดูลหลักๆ ได้แก่ โมดูลสกัดฟีเจอร์ โมดูลประมวลผลฟีเจอร์ด้วยเทคนิคทางสถิติ และโมดูลการตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์เคลื่อนที่ โมดูลหลัก ที่ได้กล่าวไว้ในบทที่ 3-6 โดยที่บทที่จะกล่าวถึงการพัฒนาโมดูลเหล่านี้ให้เป็นระบบที่ใช้งานได้ และประเมินความสำเร็จของระบบนี้ในห้องปฏิบัติการ (เวอร์ชั่นแอลฟ่า)

โครงสร้างของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย แสดงดังภาพที่ 33 แยกการทำงานในส่วนของเซิร์ฟเวอร์ซึ่งเป็นที่เก็บฐานข้อมูลทั้งหมดที่ได้รับรายงานจากอุปกรณ์เคลื่อนที่ จากส่วนของโปรแกรมบนอุปกรณ์เคลื่อนที่ โดยที่โมดูลหลักทั้งหมดทำงานอยู่บนอุปกรณ์เคลื่อนที่



ภาพที่ 33 โครงสร้างของระบบตรวจจับและเตือนภัยแอพพลิเคชันอันตราย

7.1 ระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเวอร์ชั่นแอลฟ่า

ผังการทำงานของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย แสดงดังภาพที่ 34 ซึ่งครอบคลุมการทำงานทั้ง 2 กรณี

1) ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App ระบบตรวจสอบทุกแอปพลิเคชั่นที่ติดตั้งบนอุปกรณ์ (คำอธิบายในตารางที่ 1 ในบทที่ 1)

2) หลังจากที่ได้ติดตั้ง mCOP App ระบบเฝ้าระวังและตรวจจับแอปพลิเคชั่นอันตรายที่แฝงตัวมากับแอปพลิเคชั่นปกติ (คำอธิบายในตารางที่ 2 ในบทที่ 1)

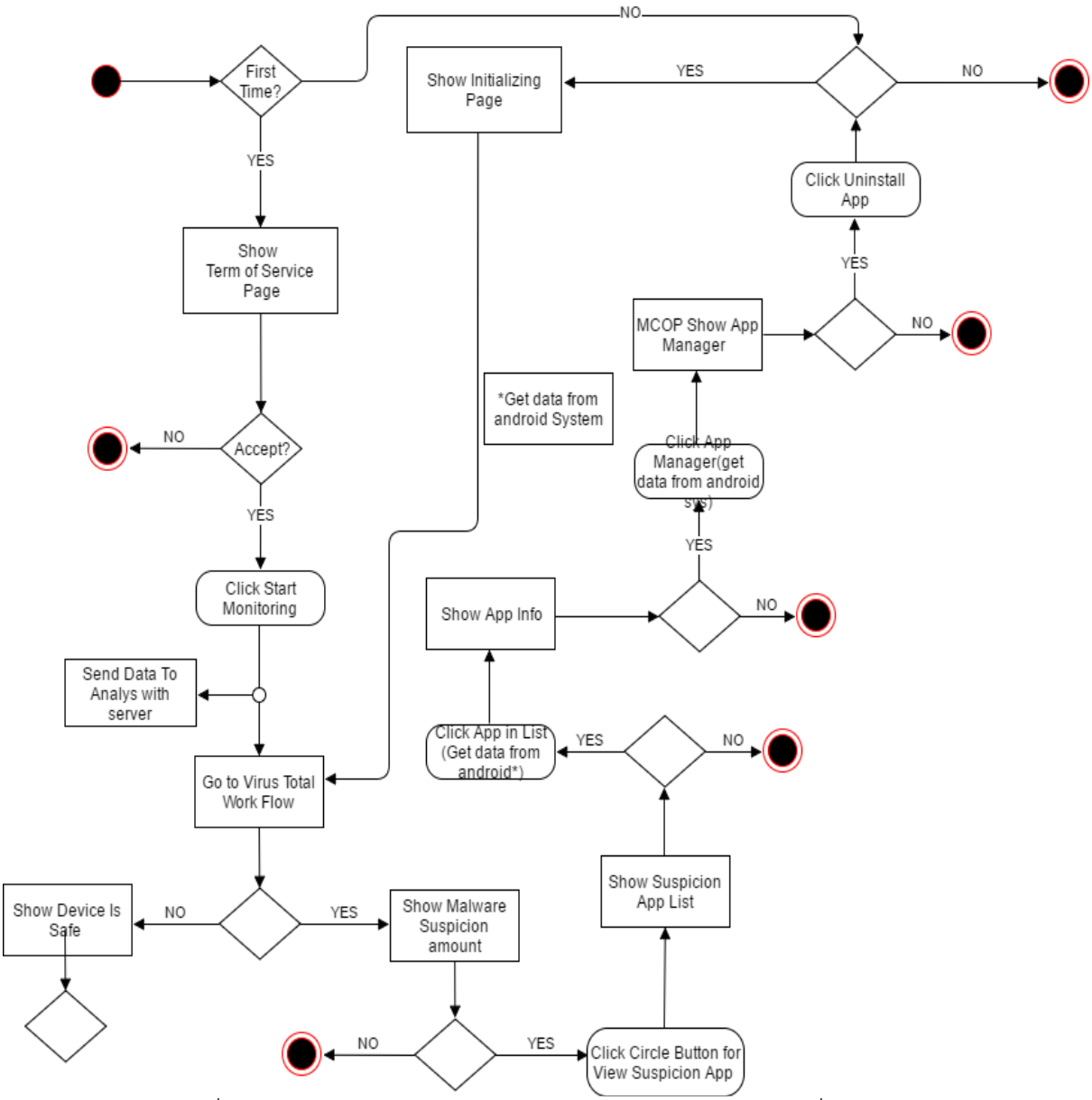
สำหรับกรณีที่ 1 ทันทีที่ผู้ใช้ติดตั้งและใช้งาน mCOP App หน้าอินเทอร์เฟซ Show Term of Service จะแสดงสอบถามผู้ใช้ หลังจากผู้ใช้ตอบตกลง ระบบจะเริ่มตรวจสอบว่ามีแอปพลิเคชั่นอันตรายในอุปกรณ์มือถือหรือไม่ (หน้าที่การทำงานของ VirusTotal-base Module) การแสดงผลจะแบ่งเป็น 2 หน้าอินเทอร์เฟซ คือ หน้าแสดงแอปพลิเคชั่นปลอดภัย และหน้าแสดงแอปพลิเคชั่นที่น่าสงสัย

ในหน้าแสดงแอปพลิเคชั่นที่น่าสงสัย ผู้ใช้สามารถคลิกดูรายละเอียดของแต่ละแอปพลิเคชั่นที่น่าสงสัยได้ และผู้ใช้สามารถสามารถที่จะเลือกที่จะถอดการติดตั้งแอปพลิเคชั่นที่น่าสงสัยได้ หรือเลือกที่จะยังเก็บแอปพลิเคชั่นเหล่านั้นไว้ (กรณีนี้ถ้าผู้ใช้กด Ignore แอปพลิชั่นที่น่าสงสัยเหล่านี้จะไม่แสดงเตือนอีกต่อไป)

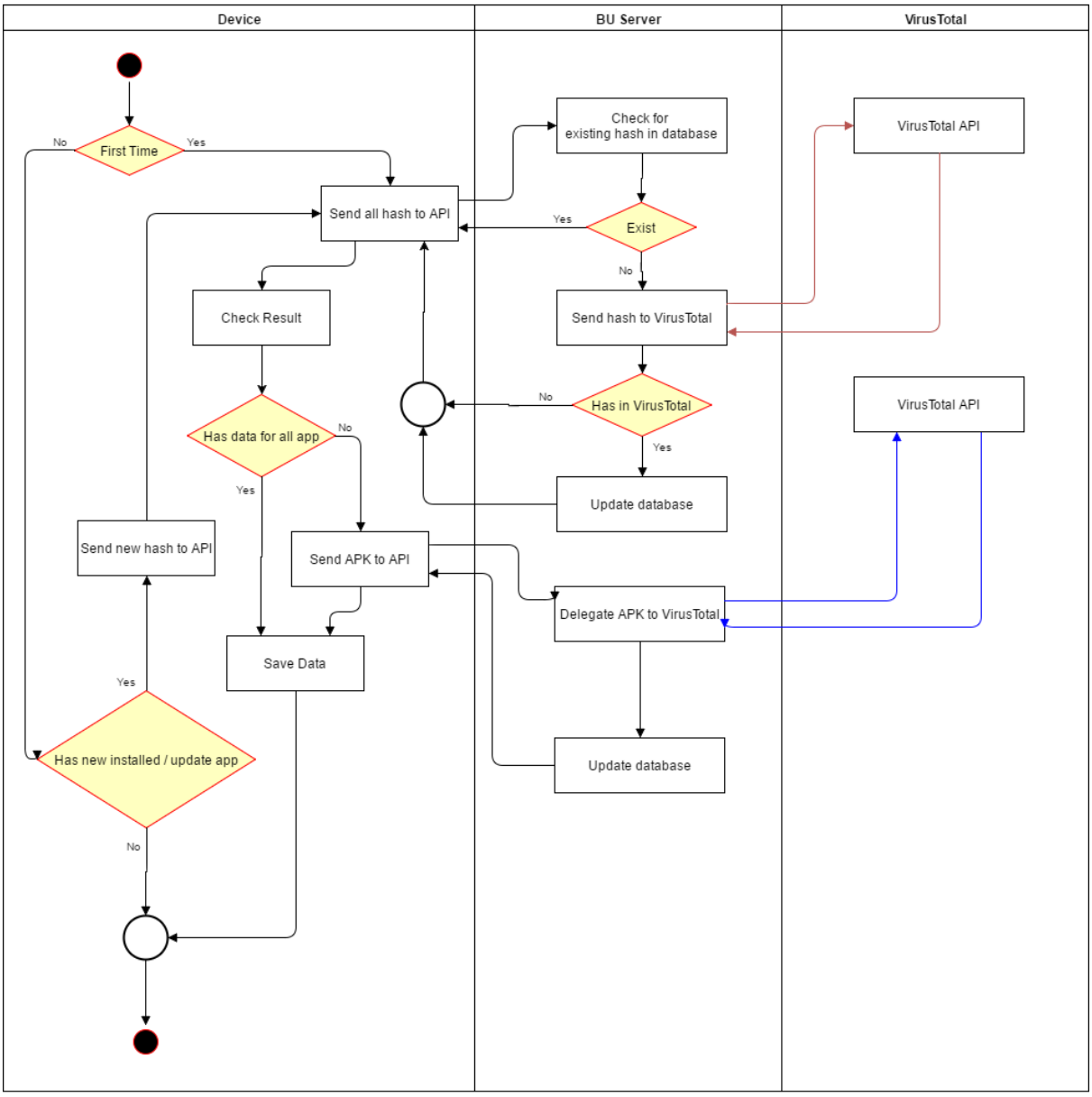
สำหรับกรณีที่ 2 ระบบเฝ้าระวังตลอดเวลาว่ามีแอปพลิเคชั่นน่าสงสัยหรือไม่ (หลังจากที่ติดตั้งและสแกนทุกแอปพลิเคชั่นที่อยู่บนอุปกรณ์มือถือแล้ว) ระบบจะตรวจสอบว่า 1) มีแอปพลิเคชั่นใดที่อัพเดตใหม่ ถ้าใช่ต้องตรวจสอบแอปพลิเคชั่นนั้นใหม่ ซึ่งอธิบายในภาพที่ 35 2) มีพฤติกรรมการส่ง sms ที่ผิดปกติ 3) มีพฤติกรรมการรับส่งข้อมูลอินเทอร์เน็ตที่ผิดปกติ (อธิบายไว้ในบทที่ 6)

แอปพลิเคชั่นที่เคยตรวสอบแล้วถ้ามีการอัพเดต แอปพลิเคชั่นที่อัพเดตจะกลายเป็นแอปพลิเคชั่นตัวใหม่ทันที เนื่องจากทุกแอปพลิเคชั่นในระบบจะใช้แฮชโค้ดในการตรวจสอบว่าเป็นแอปพลิเคชั่นเดิมหรือไม่ เมื่อแฮชโค้ดเปลี่ยน แอปพลิเคชั่นนั้นจะต้องตรวจสอบในฐานข้อมูลอีกครั้งว่าเป็นอันตรายหรือไม่ กรณีไม่พบในฐานข้อมูล APK นั้นจะส่งต่อไปที่ mCOP server และผ่านไปที่ VirusTotal-based Module ตามผังการทำงานที่ 35

นอกจากนี้ การทำงานที่ต้องมีการส่งต่อข้อมูลและโต้ตอบระหว่าง ระบบ อุปกรณ์ และผู้ใช้ ผังการทำงานแสดงในภาพที่ 36

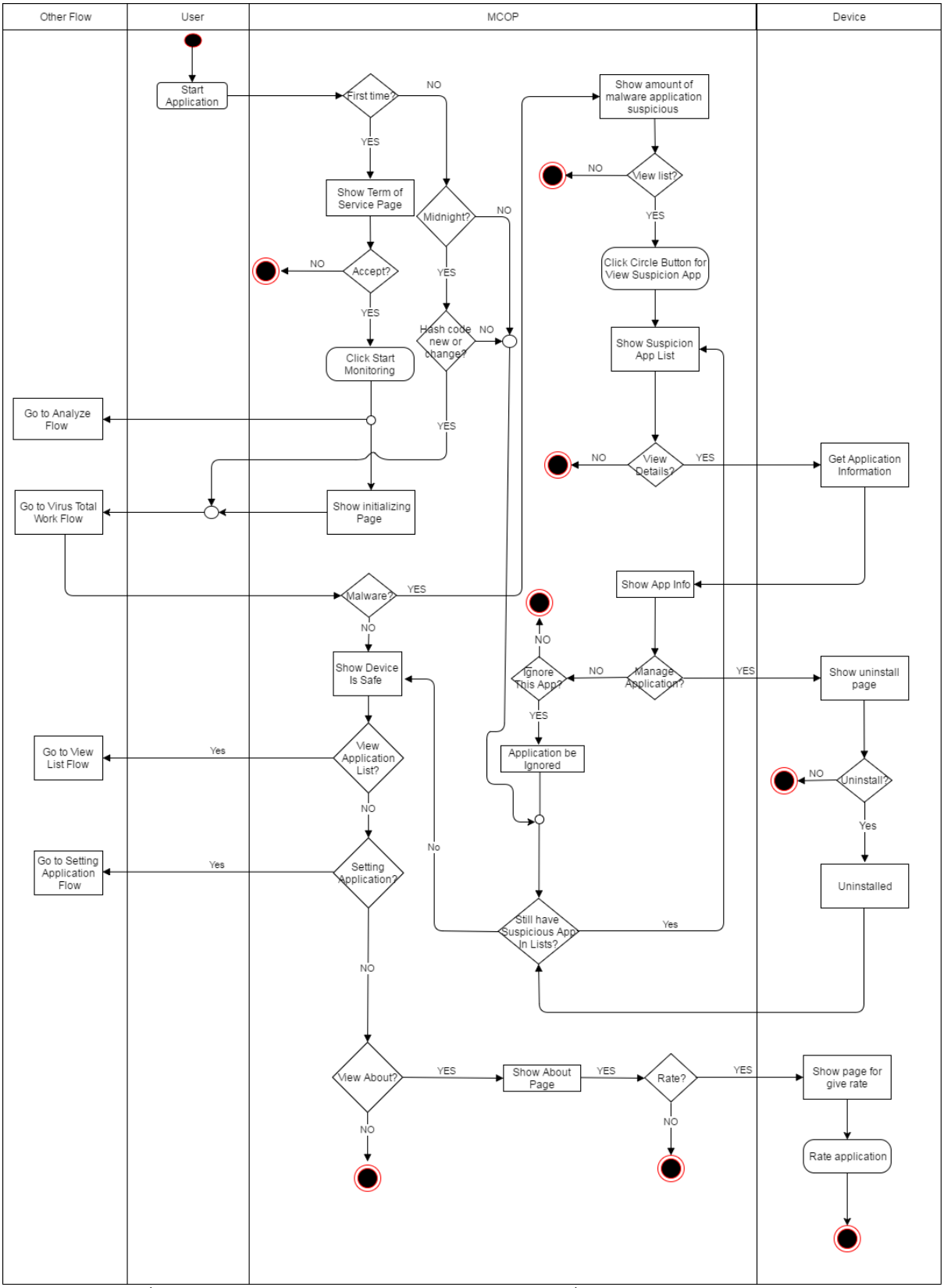


ภาพที่ 34 ผังการทำงานของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย



Text

ภาพที่ 35 ผังการทำงานของการตรวจสอบแอปพลิเคชั่นโดยใช้ Virustotal-based Module และอัพเดตฐานข้อมูลแอปพลิเคชั่นอันตรายอัตโนมัติ (ฐานข้อมูลของระบบ)



ภาพที่ 36 ผังการส่งต่อข้อมูลของระบบ mCOP ระหว่างฝั่งของ ระบบ ผู้ใช้ และอุปกรณ์

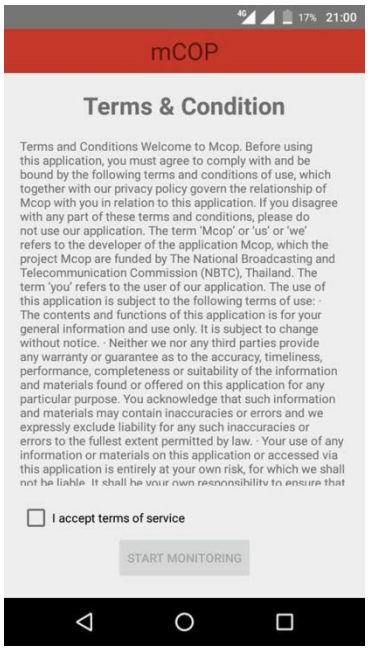
7.2 หน้าอินเทอร์เฟซของแอปพลิเคชั่นตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเวอร์ชั่นแอลฟ่า

แอปพลิเคชั่นในที่มีฟังก์ชั่นการทำงานในลักษณะนี้ในตลาดนั้น รู้จักกันในนาม แอนติไวรัส แอปพลิเคชั่นที่เป็นแอนติไวรัสนั้น หน้าอินเทอร์เฟซของแอนติไวรัสส่วนใหญ่จะมีดังนี้

(1) หน้า Term & Condition เป็นหน้าที่ชี้แจงกฎข้อบังคับและข้อควรปฏิบัติซึ่งผู้ใช้จะต้องตกลงยินยอม ผู้ใช้จึงสามารถใช้แอปพลิเคชั่นนี้ได้

(2) หน้าอินเทอร์เฟซเพื่อตรวจสอบแอปพลิเคชั่นบนอุปกรณ์ ซึ่งในปัจจุบันหน้าที่ได้ออกแบบมาให้กระชับและสื่อสารกับผู้ใช้เพียงกดหนึ่งคลิก

(3) หน้าอินเทอร์เฟซเพื่อรายงานผลการตรวจสอบ พร้อมระบุระดับความเสี่ยง (มีเฉพาะบางแอนติไวรัส)



ภาพที่ 37 หน้าอินเทอร์เฟซ Terms & Condition

สำหรับแอปพลิเคชั่น mCOP นั้น หน้าอินเทอร์เฟซต่างๆ แสดงในภาพที่ 37-41 โดยเน้นการออกแบบที่เรียบง่ายและใช้งานง่าย สำหรับโทนสีที่เลือกเป็นพื้นสีเทาตัดแดง ซึ่ง สีแดง เป็นสีตราสัญลักษณ์ของ กสทช. เริ่มต้นด้วยหน้า Term & Condition ของแอปพลิเคชั่น mCOP แสดงในภาพที่ 37 ซึ่งมีเนื้อหารายละเอียดดังนี้

" Terms and Conditions

Welcome to Mcop. Before using this application, you must agree to comply with and be bound by the following terms and conditions of use, which together with our privacy policy govern the relationship of Mcop with you in relation to this application. If you disagree with any part of these terms and conditions, please do not use our application.

The term 'Mcop' or 'us' or 'we' refers to the developer of the application Mcop, which the project Mcop are funded by The National Broadcasting and Telecommunication Commission (NBTC), Thailand. The term 'you' refers to the user of our application.

The use of this application is subject to the following terms of use:

· The contents and functions of this application is for your general information and use only. It is subject to change without notice.

· Neither we nor any third parties provide any warranty or guarantee as to the accuracy, timeliness, performance, completeness or suitability of the information and materials found or offered on this application for any particular purpose. You acknowledge that such information and materials may contain inaccuracies or errors and we expressly exclude liability for any such inaccuracies or errors to the fullest extent permitted by law.

· Your use of any information or materials on this application or accessed via this application is entirely at your own risk, for which we shall not be liable. It shall be your own responsibility to ensure that any products, services or information available through this application meet your specific requirements.

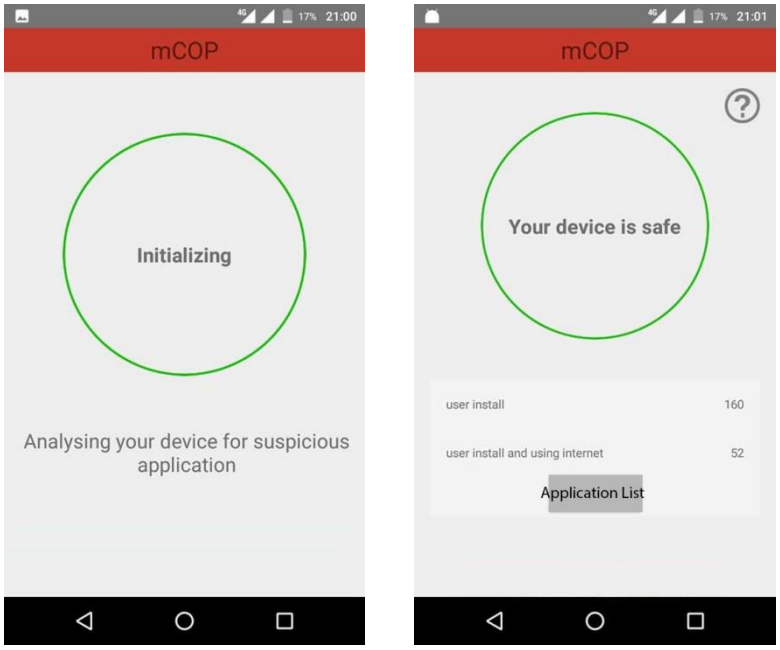
· All content on the application is protected by copyright, and owned or controlled by Mcop, or the party accredited as the owner or provider of the content. You must abide by all copyright notices, information, or restrictions contained in any content accessed through the application.

· When you download the Mcop application, you are bound by the terms of the GNU General Public License. Full GNU GPL license details here: (http://www.gnu.org/licenses/licenses.html#GPL)

· Unauthorised use of this application may give rise to a claim for damages and/or be a criminal offence.

· Your use of this application and any dispute arising out of such use of the application is subject to the laws of Thailand."

ต่อมาหน้าอินเทอร์เฟซสำหรับตรวจสอบแอปพลิเคชั่น ดังแสดงในภาพที่ 38 ประกอบด้วยหน้าตรวจสอบแอปพลิเคชั่นบนอุปกรณ์มือถือซึ่งเป็นวงกลมสีเขียวกระพริบเพื่อแสดงว่าแอปพลิเคชั่นกำลังทำงาน และเมื่อตรวจสอบเสร็จสิ้นแล้ว รายงานผลจะขึ้นมาว่า อุปกรณ์มือถือของผู้ใช้ปลอดภัยหรือไม่ ถ้าปลอดภัยจะแสดงคำว่า "Your device is safe."



ก) ข)

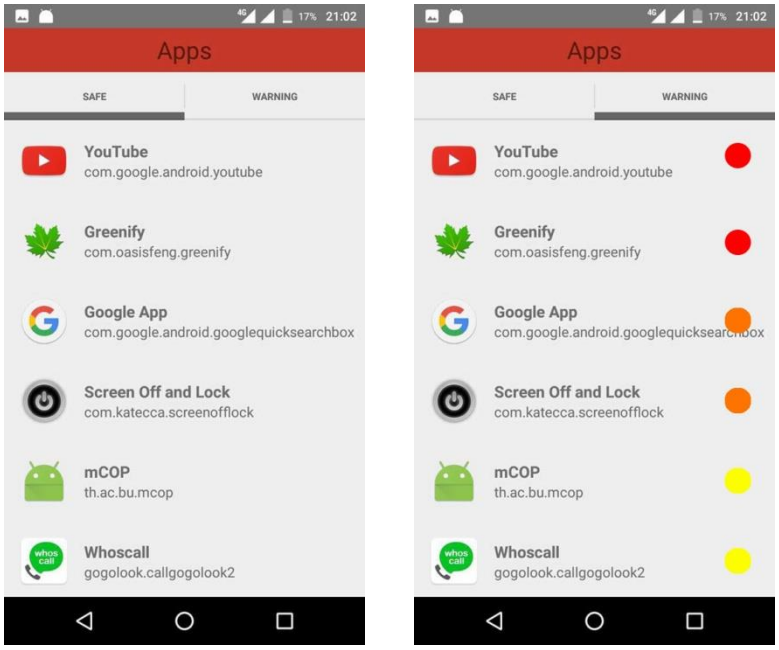
ภาพที่ 38 หน้าอินเทอร์เฟซ ก) การตรวจสอบแอปพลิเคชั่นบนอุปกรณ์มือถือ ข) รายงานผลดำเนินการ

ต่อมาหน้ารายการแอปพลิเคชั่นที่ปลอดภัย และหน้ารายการแอปพลิเคชั่นที่น่าสงสัยพร้อมระดับความเสี่ยง ดังแสดงในภาพที่ 39 (ก) และ 39 (ข) (แอปพลิเคชั่นในภาพนี้เป็นเพียงตัวอย่างที่สมมติขึ้นเท่านั้น) mCOP จะตรวจสอบและแยกแอปพลิเคชั่นที่น่าสงสัยมารายงาน โดยระดับความเสี่ยงมีรายงานมี 3 ระดับ คือ ระดับความเสี่ยง (การวิเคราะห์ระดับความเสี่ยงอธิบายในหัวข้อที่ 5.1.1)

(1) ระดับ เสี่ยงน้อย (Low Risk) แสดงด้วย จุดสีเหลือง

(2) ระดับ เสี่ยง (Risk) แสดงด้วย จุดสีส้ม

(3) ระดับ เสี่ยงมาก (High Risk) แสดงด้วย จุดสีแดง



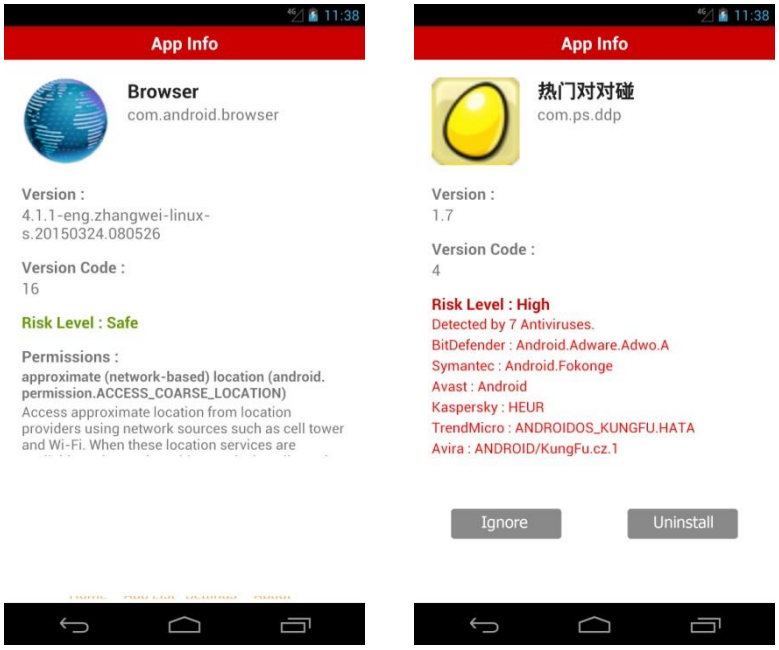
ก) ข)

ภาพที่ 39 หน้าอินเทอร์เฟซ ก) รายงานผลแอปพลิเคชั่นปลอดภัย
ข) รายงานผลแอปพลิเคชั่นน่าสงสัยพร้อมระดับความเสี่ยง (จุดสี)

หลังจากแสดงหน้ารายการแอปพลิเคชั่นที่น่าสงสัยแล้ว ผู้ใช้สามารถคลิกแต่ที่ชื่อแอปพลิเคชั่นเพื่อเปิดดูรายละเอียดตามแสดงในภาพที่ 40(ข) ขณะที่ภาพที่ 40(ก) แสดงรายละเอียดแอปพลิเคชั่นปลอดภัย กรณีที่เป็นแอปพลิเคชั่นที่น่าสงสัยผู้ใช้สามารถเลือกที่จะ "ignore" หรือ "uninstall"ได้

กรณีเลือก "ignore" หมายถึง ผู้ใช้ไม่ต้องการให้แอปพลิเคชั่นนั้นถูกแสดงว่าน่าสงสัยในครั้งถัดไป กล่าวอีกนัยหนึ่ง ระบบจะย้ายแอปพลิเคชั่นนั้นไปในฝั่งที่เป็นแอปพลิเคชั่นปลอดภัย

กรณีเลือก "uninstall" หมายถึง ผู้ใช้ต้องการถอดการติดตั้งแอปพลิเคชั่นออกจากอุปกรณ์ ระบบจะผ่านไปยังหน้าถอดการติดตั้งซึ่งอยู่ในส่วนระบบปฏิบัติการของอุปกรณ์



ก) ข)

ภาพที่ 40 หน้าอินเทอร์เฟซแสดงรายละเอียดของแต่ละแอปพลิเคชั่น ก) รายละเอียดของแอปพลิเคชั่นปลอดภัย ก) รายละเอียดของแอปพลิเคชั่นน่าสงสัยพร้อมระดับความเสี่ยง

ภาพที่ 41 หน้าอินเทอร์เฟซ เกี่ยวกับแอปพลิเคชั่น

ภาพที่ 42 ไอคอนของแอปพลิเคชั่น mCOP

สำหรับภาพที่ 41 หน้า "about" แสดงข้อมูลเกี่ยวกับแอปพลิเคชั่น ในที่นี้ mCOP จะใส่ผู้ให้ทุนสนับสนุนการจัดทำแอปพลิเคชั่นนี้ ได้แก่ กทปส และ กสทช โดยใส่โลโก้และชื่อองค์กร และโลโก้หรือไอคอนของแอปพลิเคชั่น mCOP แสดงในภาพที่ 42

7.3 การทดสอบระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเวอร์ชั่นแอลฟ่า

การทดสอบระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเวอร์ชั่นแอลฟ่านั้น ทีมนักวิจัยได้ออกแบบชุดทดสอบในรูปแบบของ User Scenario ทั้งหมด 74 กรณี โดยที่ในเวอร์ชั่นแอลฟ่าสามารถทดสอบได้ สามารถทดสอบได้ 34 กรณี ที่เหลือ 38 กรณีใช้สำหรับทดสอบเวอร์ชั่นเบต้า

สำหรับ user scenario ทั้งหมด 74 กรณี จัดเป็นชุดได้เป็น 4 ชุด ดังนี้

1. ตรวจสอบความถูกต้องของการใช้งาน mCOP Application กรณีเปิดใช้งาน Application ในครั้งแรกได้สำเร็จ (กรณีพบ Application ที่เป็น Malware)
2. ตรวจสอบความถูกต้องของการใช้งาน mCOP Application กรณีเปิดใช้งาน Application ในครั้งแรก (กรณีไม่พบ Application ที่เป็น Malware)

3. ตรวจสอบความถูกต้องของการใช้งาน mCOP Application กรณีไม่ใช่การใช้งานครั้งแรกและพบ Application ที่เป็น Malware

4. ตรวจสอบความถูกต้องของการใช้งาน mCOP Application กรณีไม่ใช่การใช้งานครั้งแรกและไม่พบ Application ที่เป็น Malware

ในแต่ละชุดทดสอบคำถามในการทดสอบจะมีประเด็นต่อไปนี้

(1) ผู้ใช้งานทำการค้นหาและติดตั้ง Application mCOP ผ่าน Browser (On PC)
(2) ผู้ใช้งานทำการค้นหาและติดตั้ง Application mCOP ผ่าน Browser (On Android Smartphone)
(3) ผู้ใช้งานทำการค้นหาและติดตั้ง Application BU Antivirus ผ่าน Application Google Play Store (On Smartphone Android)
(4) ผู้ใช้เปิดใช้งาน Application mCOP ในครั้งแรกแล้วพบ Application ที่เป็น Malware
(5) ผู้ใช้งานดูรายละเอียดของ Application ที่เข้าข่ายเป็น Malware
(6) ผู้ใช้ทำการถอนการติดตั้ง Application ที่เข้าข่ายเป็น Malware
(7) ผู้ใช้ทำการยกเว้น Application ที่เข้าข่ายเป็น Malware เพื่อให้ระบบไม่ทำการตรวจสอบอีกครั้ง
(8) ผู้ใช้ทำการตั้งค่า Application mCOP
(9) ผู้ใช้ทำการดูข้อมูลเกี่ยวกับ Application mCOP

ในประเด็นทั้ง 9 แบ่งเป็นที่ใช้ทดสอบกับเวอร์ชั่นแอลฟ่า 5 ประเด็น (ส่วนที่เหลืออีก 4 ประเด็นนำไปใช้ทดสอบในเวอร์ชั่นเบต้า) ดังนี้

(1) ผู้ใช้เปิดใช้งาน Application mCOP ในครั้งแรกแล้วพบ Application ที่เป็น Malware
(2) ผู้ใช้งานดูรายละเอียดของ Application ที่เข้าข่ายเป็น Malware
(3) ผู้ใช้ทำการถอนการติดตั้ง Application ที่เข้าข่ายเป็น Malware
(4) ผู้ใช้ทำการยกเว้น Application ที่เข้าข่ายเป็น Malware เพื่อให้ระบบไม่ทำการตรวจสอบอีกครั้ง
(5) ผู้ใช้ทำการดูข้อมูลเกี่ยวกับ Application mCOP

การรายงานผลการทดสอบแสดงในภาคผนวก ค. ทั้ง 4 ชุดทดสอบ ซึ่งมีทั้งหมด 72 กรณี แต่ในการทดสอบกับเวอร์ชั่นแอลฟ่าครอบคลุมทั้งหมด 34 กรณี เท่านั้น ซึ่งผ่านทั้งหมด 34 กรณี

7.4 รายงานผลการประเมินที่ได้จากการประเมินความสำเร็จของแอลฟ่าเวอร์ชั่น มาปรับปรุงระบบเป็นเบต้าเวอร์ชั่น

หลังจากเวอร์ชั่นแอลฟ่าผ่านการทดสอบทั้งหมด 34 กรณี (ทดสอบในห้องปฏิบัติการ) แล้วนั้น มีประเด็นในการพัฒนา ดังนี้

(1) ผู้ใช้งานทำการค้นหาและติดตั้ง Application mCOP ผ่าน Browser (On PC)
(2) ผู้ใช้งานทำการค้นหาและติดตั้ง Application mCOP ผ่าน Browser (On Android Smartphone)
(3) ผู้ใช้งานทำการค้นหาและติดตั้ง Application BU Antivirus ผ่าน Application Google Play Store (On Smartphone Android)
(4) ผู้ใช้ทำการตั้งค่า Application mCOP

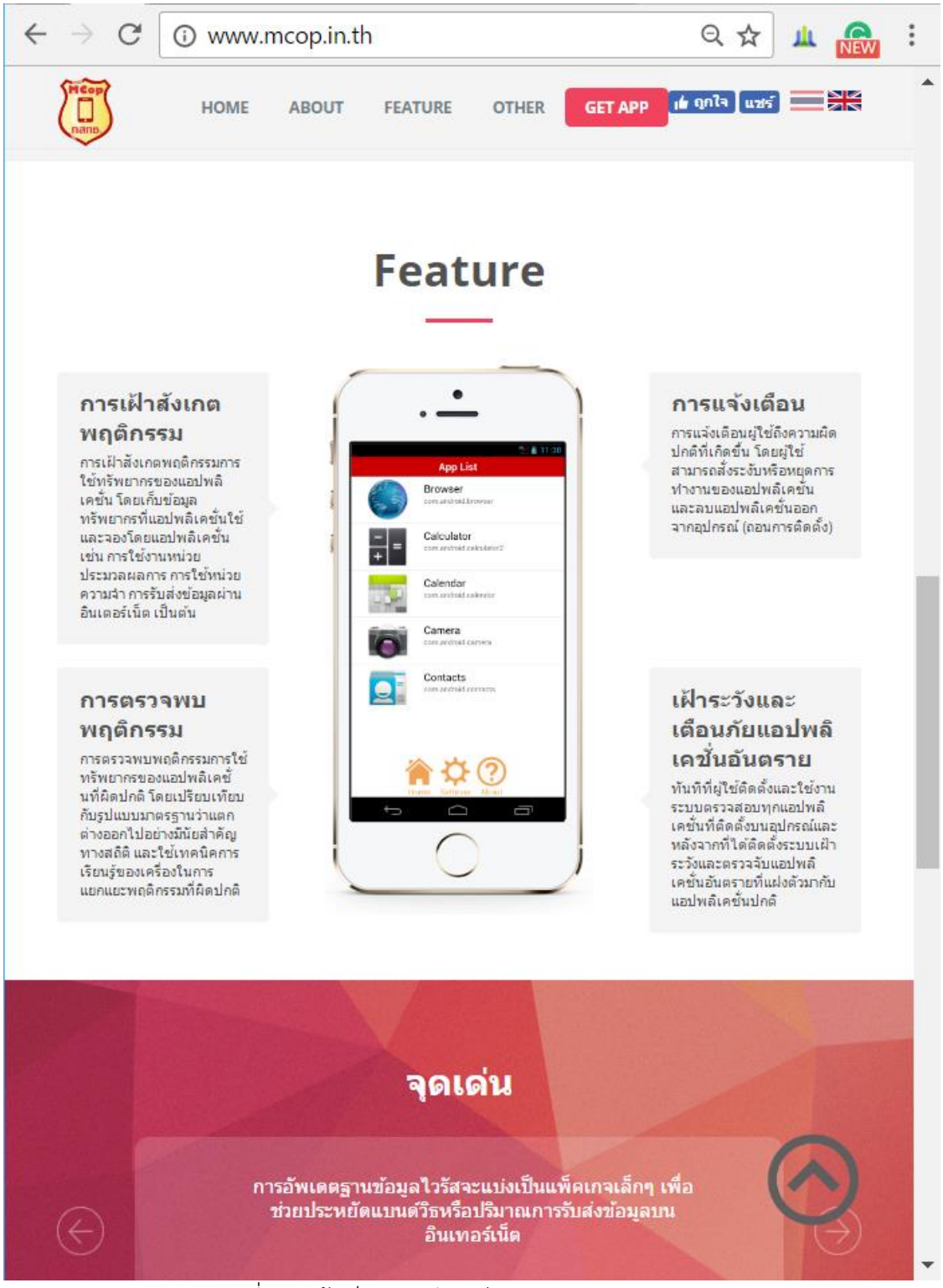
บทที่ 8
การพัฒนาเว็บไซต์ของระบบเพื่อเป็นช่องทางในการสื่อสารกับผู้ใช้

การพัฒนาเว็บไซต์เป็นการเพิ่มช่องทางในการสื่อสารกับผู้ใช้ ทั้งในแง่ของการประชาสัมพันธ์ อัพเดตข่าวสาร ช่องทางติดต่อกับทีมพัฒนาในกรณีที่มีคำถามหรือปัญหา และยังเป็นช่องทางที่ให้ผู้พัฒนาได้รับ feedback จากผู้ใช้อีกด้วย เบื้องต้นทางผู้พัฒนาได้สร้างจัดทำเว็บไซต์ โดยใช้โดเมนเนม คือ www.mcop.in.th โดยการออกแบบเว็บไซต์ของระบบเป็นในลักษณะ responsive ซึ่งรองรับใช้งานทั้งผ่านคอมพิวเตอร์ แท็ปเล็ต และอุปกรณ์มือถือ

ข้อมูลหลักที่แสดงในเว็บไซต์ประกอบด้วย ฟีเจอร์สำคัญของ mCOP app ทั้งการเปรียบเทียบ ข้อดีและข้อจำกัด หลักการทำงานของแอปพลิเคชั่น โดยที่หน้าหลักของเว็บไซต์ ประกอบโดยเมนูหลักที่เชื่อมต่อไปยังหน้าต่างๆ ของเว็บไซต์ซึ่งประกอบด้วย การติดตั้งแอปพลิคชั่น เว็บบอร์ดสำหรับถามตอบ ข้อมูลเกี่ยวกับแอปพลิเคชั่น แบบสำรวจ ผู้สนับสนุน (กสทช กทปส และมหาวิทยาลัยกรุงเทพ) และข้อมูลสำหรับติดต่อทีมพัฒนา ดังแสดงในภาพที่ 43

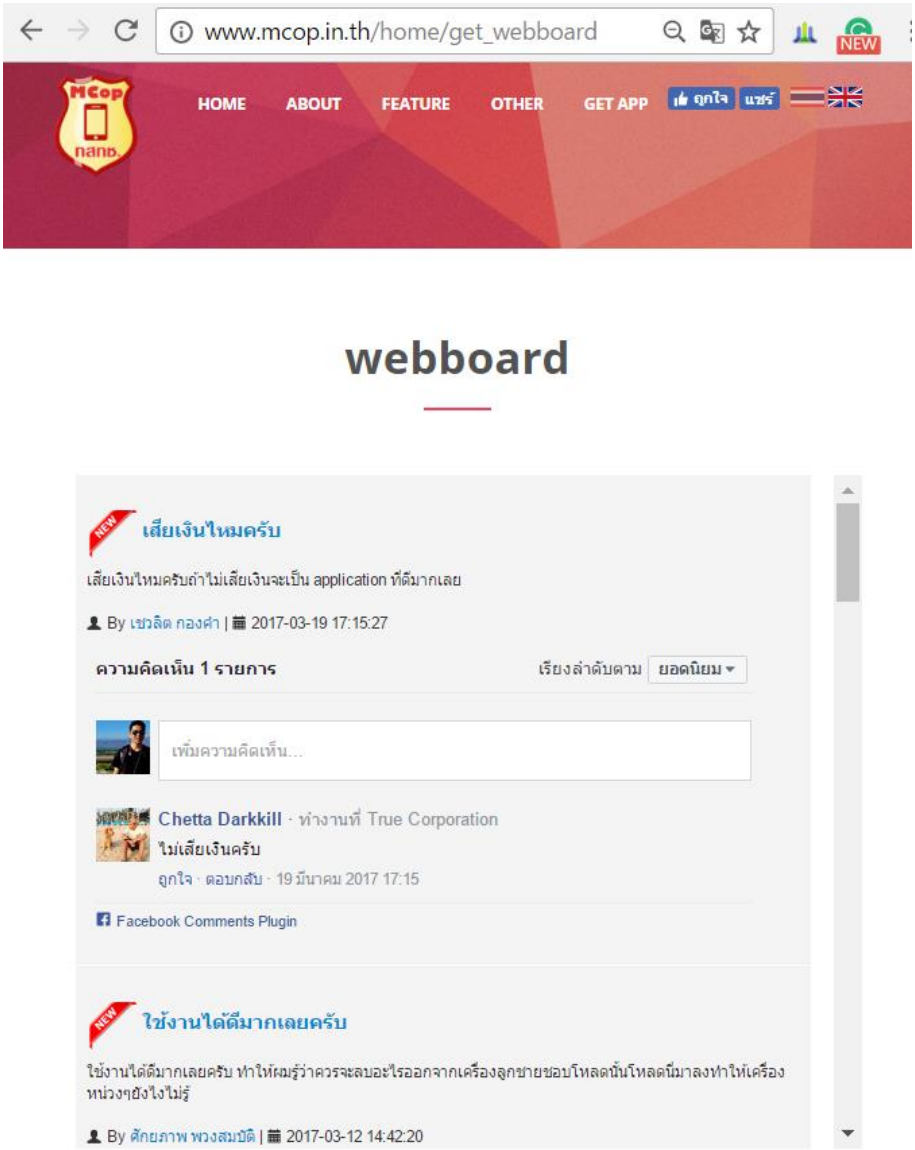
เนื่องจากหลักการของแอปพลิเคชั่น mCOP ต่างจากแอปพลิเคชั่นอื่นๆ ที่มักใช้ malware signature ในการตรวจจับโค้ดที่เป็นอันตราย ในขณะที่ mCOP ใช้การตรวจพบพฤติกรรมการการรับส่งข้อมูลของแอปพลิเคชั่นที่ผิดปกติ โดยเปรียบเทียบกับรูปแบบมาตรฐานว่าแตกต่างออกไปอย่างมีนัยสำคัญทางสถิติ และใช้เทคนิคการเรียนรู้ของเครื่องในการแยกแยะพฤติกรรมที่ผิดปกติ ในเว็บไซต์จึงได้นำเสนอหลักการเบื้องต้นในส่วนของฟีเจอร์สำคัญของ mCOP ดังภาพที่ 44

ภาพที่ 43 หน้าหลักของเว็บไซต์ www.mcop.in.th



ภาพที่ 44 หน้าเว็บแสดงฟีเจอร์สำคัญของ mCOP

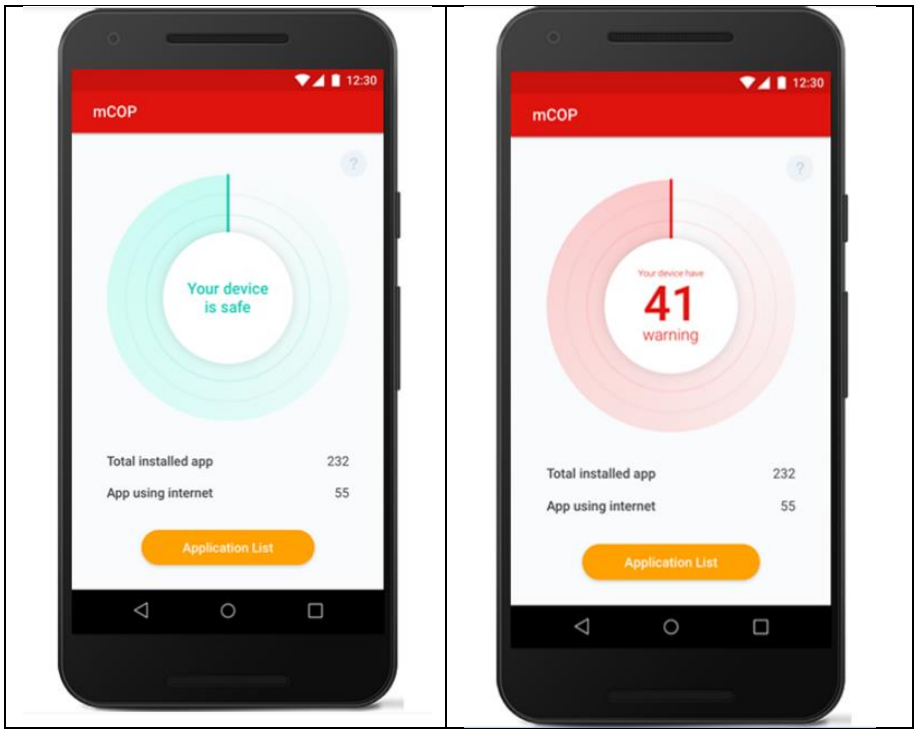
เนื่องจากในปัจจุบัน ผู้ใช้ส่วนมากมักมีบัญชีของสื่อสังคมออนไลน์อยู่แล้ว ทั้งนี้ในประเทศไทย facebook เป็นสื่อสังคมออนไลน์ได้รับความนิยมเป็นอันดับหนึ่ง ดังนั้นเว็บไซต์นี้จึงออกแบบให้สามารถเชื่อมกับ facebook ได้ ทั้งในการกด like และ share ซึ่งเป็นการประชาสัมพันธ์ได้อย่างดี นอกจากนั้นผู้ใช้สามารถใช้ login ของ facebook ในการแสดงความคิดเห็นในเว็บบอร์ดสำหรับถามตอบและแลกเปลี่ยนประสบการณ์ในการใช้งานดังภาพที่ 45 ซึ่งแสดงหน้าของเว็บบอร์ดที่ผู้ใช้ได้ใช้บัญชีของ facebook ในการแสดงความคิดเห็นและถามตอบ



ภาพที่ 45 เว็บบอร์ดสำหรับถามตอบและแลกเปลี่ยนประสบการณ์ในการใช้งาน

บทที่ 9
การประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเมื่อนำไปใช้จริง

ระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายได้ผ่านการทดสอบการใช้งานในห้องแลปตามรายงานในบทที่ 7 เมื่อนำไปใช้จริง เริ่มตั้งแต่ 1 เมษายน 2560 และได้มีการอัพเดตเวอร์ชั่นล่าสุดเมื่อ 1 สิงหาคม 2560 โดยได้ปรับชื่อ แอปพลิเคชั่นนี้ใช้ชื่อว่า mCOP by NBTC (เรียกย่อๆ ว่า mCOP) และปรับปรุงในส่วนของอินเทอร์เฟซให้มีกราฟิกที่สวยงามน่าใช้มากขึ้นดังภาพด้านล่าง (ภาพที่ 46)

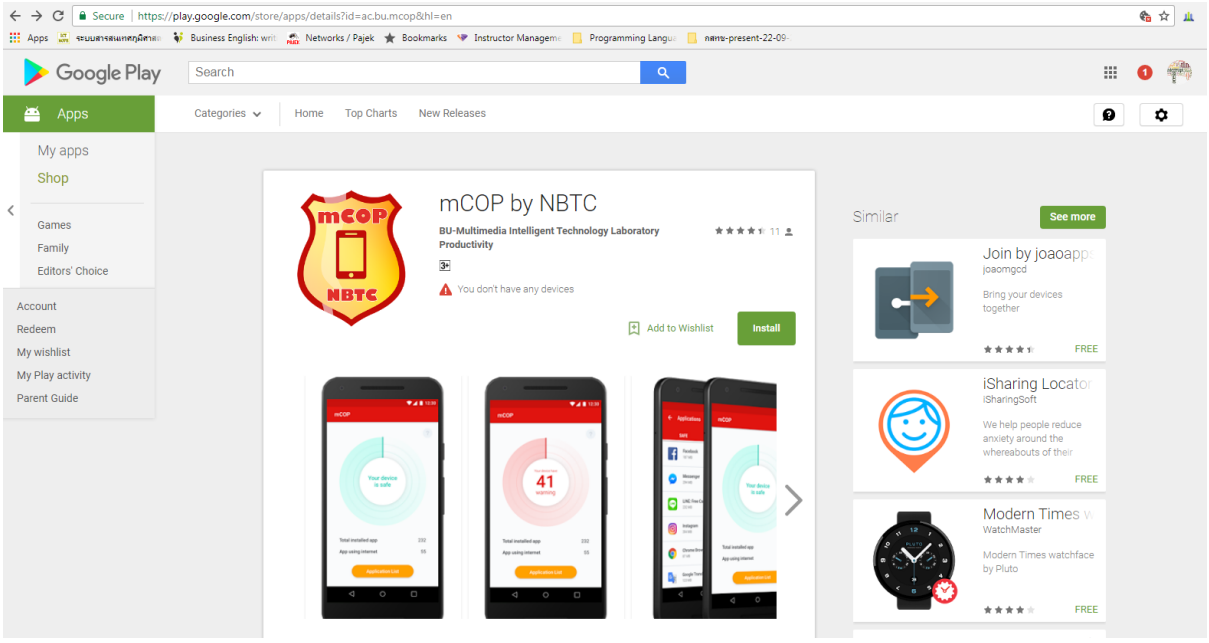


ภาพที่ 46 หน้าอินเทอร์เฟซหลักของ mCOP by NBTC

ในบทนี้ทีมผู้พัฒนาจะกล่าวถึงหัวข้อต่อไปนี้ การอัพโหลดแอปพลิเคชั่นใน Google Play การทดสอบกับผู้ใช้ทั่วไปโดยใช้แบบประเมินแบบออนไลน์ และผลตอบรับที่ได้จากการประเมินความสำเร็จของระบบเมื่อนำไปใช้จริงและประมวลผลข้อมูล

9.1 การอัพโหลดแอปพลิเคชั่นใน Google Play

แอปพลิเคชั่น mCOP by NBTC เวอร์ชั่นล่าสุด 0.0.4 นี้มีหน้าแนะนำแอปพลิเคชั่นใน Google Play ตามรูปด้านล่าง (ภาพที่ 47)



ภาพที่ 47 หน้าแนะนำแอปพลิเคชั่น mCOP by NBTC ใน Google Play

ในหน้าแนะนำแอปพลิเคชั่น ได้ระบุทีมผู้พัฒนา คือ BU-Multimedia Intelligent Technology Laboratory พร้อมคำอธิบายแอปพลิเคชั่นไว้ดังต่อไปนี้

"The application "mCop" is a new behavior-based anomaly detection application that detecting suspicious apps from application's network behavior and SMS usage.
The main goal of the application is to protect mobile device users from malicious applications by identification of malicious applications installed on a mobile device.
More specifically, MCop attempts to monitor for a new type of mobile malware with self-updating capabilities which cannot be detected using the standard signatures approach.
Mcop collects network data of each application, then extract its' features (or patterns) and send to Mcop server for the analysis using machine-learning methods.
If the patterns of the data usages deviations from the application's expected behavior, mCop will warns the user.

For the SMS part, if a third party application tries to send an SMS in the background, mCop will send a notification to warn the user.

Note that, mcop collects network data (in term of amount usages) as well as monitoring user SMS usages for the detection. Mcop will not collect personal contents.

For more information, please go to www.mcop.in.th"

ฟีเจอร์สำคัญของแอปพลิเคชั่นนี้ที่ทีมพัฒนาได้แจ้งกับผู้ใช้ ได้แก่

- Monitoring and Warning of Malicious Applications
- Monitoring your application usage
- Warning when you have a suspect SMS

การขอสิทธิ์ในการเข้าถึงของแอปพลิเคชั่นนี้ ได้ระบุไว้ใน Google Play ดังแสดงตารางด้านล่าง

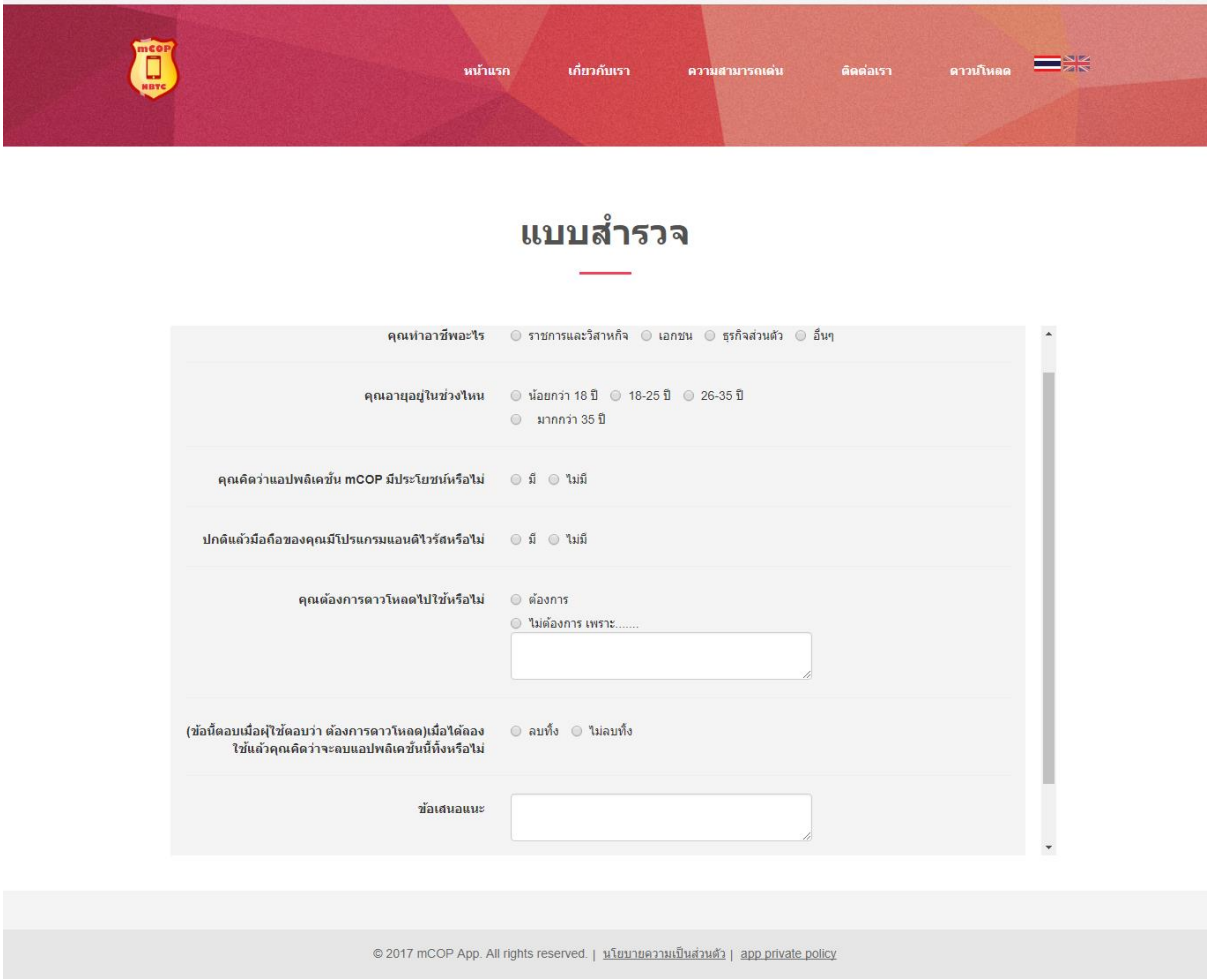
ประเภท	สิทธิ์ที่แอปพลิเคชั่น mCOP ขอจากผู้ใช้
Location	precise location (GPS and network-based)
SMS	read your text messages (SMS or MMS) receive text messages (SMS)
Phone	read call log read phone status and identity write call log
Photos/Media/Files	read the contents of your USB storage modify or delete the contents of your USB storage
Storage	read the contents of your USB storage modify or delete the contents of your USB storage
Wi-Fi connection information	view Wi-Fi connections
Device ID & call information	read phone status and identity
Others	update component usage statistics view network connections full network access run at startup control vibration prevent device from sleeping

9.2 การทดสอบกับผู้ใช้ทั่วไปโดยใช้แบบประเมินแบบออนไลน์

แบบประเมินออนไลน์นั้นมีในเว็บไซต์ www.mcop.in.th ซึ่งทางทีมผู้พัฒนาได้ออกแบบคำถามให้กระชับและตรงเป้าประสงค์ในเรื่องการยอมรับและความต้องการของผู้ใช้ คำถามทั้งหมดในแบบประเมินมีทั้งหมด 6 คำถาม ดังแสดงในตารางที่ 2 และหน้าเว็บดังแสดงในภาพที่ 48

ตารางที่ 2: คำถามในแบบประเมินออนไลน์

ข้อ	คำถาม	ตัวเลือก
1	คุณอายุอยู่ในช่วงไหน	☐ น้อยกว่า 18 ปี ☐ 18 - 25 ปี ☐ 26 - 35 ปี ☐ มากกว่า 35 ปี
2	คุณทำอาชีพอะไร	☐ ราชการ, รัฐวิสาหกิจ ☐ เอกชน ☐ ธุรกิจส่วนตัว ☐ อื่นๆ
3	ปกติแล้วมือถือของคุณมีโปรแกรมแอนติไวรัสหรือไม่	☐ มี ☐ ไม่มี
4	คุณคิดว่าแอปพลิเคชั่น mCOP มีประโยชน์หรือไม่	☐ มี ☐ ไม่มี
5	คุณต้องการดาวโหลดไปใช้หรือไม่	☐ ต้องการ ☐ ไม่ต้องการ เพราะ .....
6	เมื่อได้ลองใช้แล้วคุณคิดว่าจะลบแอปพลิเคชั่นนี้ทิ้งหรือไม่ (ตอบคำถามนี้ก็ต่อเมื่อคุณได้ดาวโหลดไปใช้แล้ว)	☐ ลบทิ้ง ☐ ไม่ลบทิ้ง เพราะ .....



ภาพที่ 48 แบบประเมินออนไลน์ ในเว็บไซต์ www.mcop.in.th

หลังจากมีผู้ตอบรับเข้าประเมินออนไลน์นั้นมีในเว็บไซต์ www.mcop.in.th ซึ่งทางทีมผู้พัฒนาได้ประชาสัมพันธ์เบื้องต้น กับนักศึกษาและบุคลากรของคณะเทคโนโลยีสารสนเทศและนวัตกรรม มหาวิทยาลัยกรุงเทพ จำนวน 100 คน โดยที่นักศึกษาที่ตอบรับนั้นได้อ่านข้อมูลเกี่ยวกับแอปพลิเคชั่นจากใน Google Play หรือ เว็บไซต์ mCOP แล้ว โดยได้ผลประเมินดังแสดงในตารางที่ 3

ตารางที่ 3: ผลประเมินออนไลน์ต่อแอปพลิเคชั่น mCOP

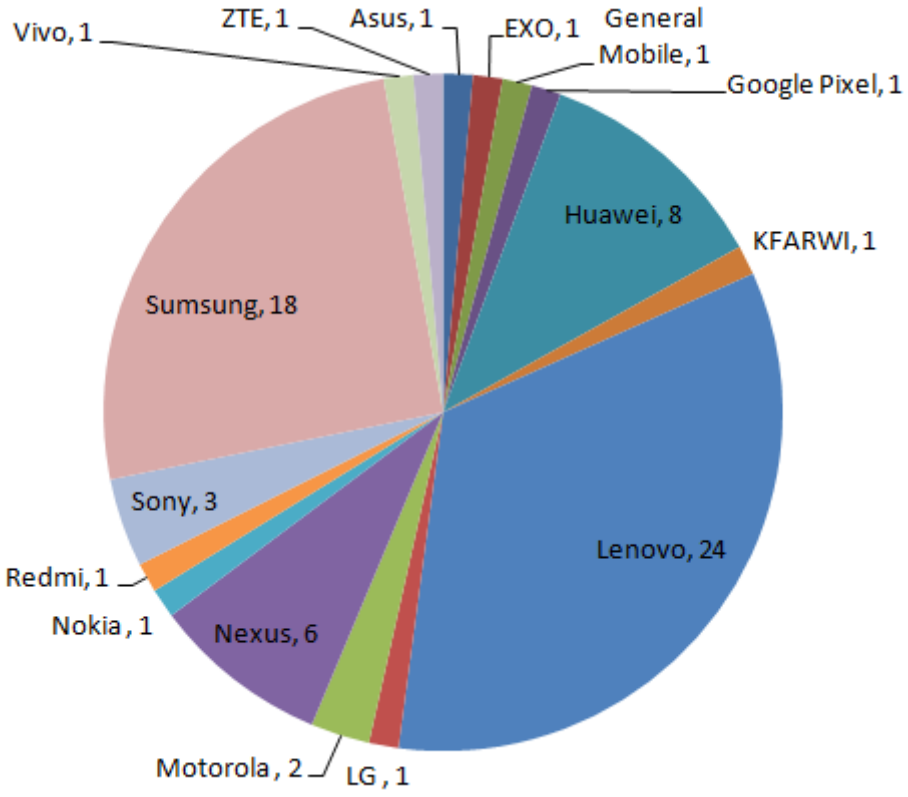
ข้อ	คำถาม	ตัวเลือก	ร้อยละ
1	**คุณอายุอยู่ในช่วงไหน**	น้อยกว่า 18 ปี 18 - 25 ปี 26 - 35 ปี มากกว่า 35 ปี	**3** **79** **11** **7**
2	**คุณทำอาชีพอะไร**	ราชการ, รัฐวิสาหกิจ เอกชน ธุรกิจส่วนตัว อื่นๆ	0 16 0 84
3	**ปกติแล้วมือถือของคุณมีโปรแกรมแอนติไวรัสหรือไม่**	มี ไม่มี	8 92
4	**คุณคิดว่าแอปพลิเคชั่น mCOP มีประโยชน์หรือไม่**	มี ไม่มี	99 1
5	**คุณต้องการดาวโหลดไปใช้หรือไม่**	ต้องการ ไม่ต้องการ	37 63
6	**เมื่อได้ลองใช้แล้วคุณคิดว่าจะลบแอปพลิเคชั่นนี้ทิ้งหรือไม่** **(ตอบคำถามนี้ก็ต่อเมื่อคุณได้ดาวโหลดไปใช้แล้ว)**	ลบทิ้ง ไม่ลบทิ้ง ยังไม่ตัดสินใจ	5 17 15

ตามตารางที่ 3 ผลประเมินจากผู้ตอบรับเข้าร่วม 100 คนแบ่งเป็นนักศึกษา 84 คน และเป็นบุคคลากร 16 คน ซึ่งในจำนวนนี้ผู้เข้าร่วมส่วนใหญ่บนอุปกรณ์เคลื่อนที่ของตนเองไม่มีโปรแกรมแอนติไวรัส (92%) ในจำนวนนี้ส่วนครึ่งหนึ่งเป็นผู้ใช้มือถือค่าย iOS อย่างไรก็ตามมีเพียง 1% ของผู้เข้าร่วมที่ไม่เห็นว่าแอปพลิเคชั่น mCOP มีประโยชน์ เนื่องจากในจำนวนวผู้ตอบรับเข้าร่วมนั้นประกอบด้วยผู้ใช้มือถือทุกค่ายไม่ใช่เฉพาะแอนดรอยด์ ผู้ที่ตัดสินใจดาวโหลดจึงมีเพียง 37% นอกจากนั้นเหตุผลรองลงมาที่ผู้เข้าร่วมไม่ดาว

โหลดเพราะบนเครื่องของเขามีโปรแกรมแอนติไวรัสอยู่แล้ว และผู้เข้าร่วมกังวลใจว่าโปรแกรมแอนติไวรัสจะทำให้เครื่องมือถือช้าลง หลังจากผู้เข้าร่วมลงแอปพลิเคชั่น mCOP แล้ว ตัดสินใจที่จะ uninstall 5% เก็บไว้ 32% (ในจำนวนนี้ยังไม่ตัดสินใจ 15%)

9.3 ผลตอบรับที่ได้จากการประเมินความสำเร็จของระบบเมื่อนำไปใช้จริงและประมวลผลข้อมูล

แอปพลิเคชั่น mCOP by NBTC เวอร์ชั่นล่าสุดเมื่อ 1 สิงหาคม 2560 จนถึง 20 กันยายน 2560 นั้น มีผู้ใช้ทั่วไปที่ดาวโหลดไปใช้และใช้โปรแกรมต่อเนื่องนานเกิน 30 วันทั้งหมด จำนวน 71 เครื่อง ซึ่งประกอบด้วยยี่ห้อต่างๆ ดังแสดงในภาพที่ 49



ภาพที่ 49 ยี่ห้อของอุปกรณ์เคลื่อนของผู้ใช้แอปพลิเคชั่น mCOP

แอปพลิเคชั่นที่มีในอุปกรณ์ของผู้ใช้ 71 เครื่องนั้น ประกอบด้วยแพ็คเก็ตหรือแอปพลิเคชั่นที่แตกต่างกันทั้งสิ้น 1,407 รายการ แต่จำนวนแฮชโค้ด (เวอร์ชั่นทีต่างกัน) จำนวน 3,232 รายการ ส่วนความคิดเห็นที่ผู้ใช้ทั่วไปได้เขียนไว้ใน Google Play มีดังแสดงตารางที่ 4

ตารางที่ 4: สรุปผลการติชมต่อแอปพลิเคชั่น mCOP ใน Google Play

คำชม	**คำเสนอแนะ**
การใช้งานง่าย และสะดวก ผู้ใช้เครื่องมีความมั่นใจทุกครั้งที่มีแอปพลิเคชั่นมีการอัพเดต	การสแกนแอปพลิเคชั่นใช้เวลานาน การแจ้งเกี่ยวกับ Access Permission ให้ชัดเจนในตัวแอปพลิเคชั่นก่อนที่จะเข้าถึง การชี้ให้เห็นถึงประโยชน์ของแอปพลิเคชั่น

ผู้ใช้เห็นประโยชน์ของแอปพลิเคชั่นโดยเฉพาะจุดเด่นที่ แอปพลิเคชั่นนี้ใช้งานง่ายและคอยเฝ้าระวังการอัพเดตของแอปพลิเคชั่นที่มีอยู่ว่าได้กลายเป็นมัลแวร์หรือไม่ โดยที่ผู้ใช้ไม่ต้องคอยอัพเดตแพต (Patch) เหมือนโปรแกรมแอนติไวรัสทั่วไป สำหรับคำเสนอแนะจากผู้ใช้นั้น ทีมผู้พัฒนาได้วางแผนการปรับปรุงโดยอัพเดตเวอร์ชั่นใหม่ในเดือนตุลาคมนี้

บทที่ 10
สรุปผลการดำเนินงาน

ตั้งแต่เริ่มโครงการจนถึงวันรายงานความก้าวหน้าครั้งที่ 2 (22 กันยายน 2558 – 22 กันยายน 2559) การดำเนินงานได้สำเร็จลุล่วงจนถึงขั้นตอนที่ 5.4 โดยในขั้นตอนที่ 1 -4 ได้เขียนรายงานเบื้องต้นส่งแล้วในวันที่ 22 ตุลาคม 2558 และ 22 เมษายน 2559 โดยขั้นตอนที่ 5.1-5.4 อยู่ในรายงานความก้าวหน้าเล่มนี้

ในระหว่างวันที่ 23 ตุลาคม 2558 ถึง 22 เมษายน 2559 (รวมระยะเวลา 6 เดือน) คณะผู้ดำเนินโครงการได้ออกแบบและพัฒนาโมดูลสกัดฟีเจอร์ใช้งานแอปพลิชั่นโดยพัฒนา APIs จาก Android Software Development Kit (SKD) เพื่อเก็บข้อมูลการเชื่อมต่อเครือข่าย เช่น ชนิดของเครือข่ายที่เชื่อมต่อ จำนวนข้อมูลที่รับ-ส่ง โหมดที่ใช้ในการรับ-ส่งเป็นต้น ต่อจากนั้นสร้างโมดูลประมวลผลฟีเจอร์ด้วยเทคนิคทางสถิติโดยพัฒนา APIs จาก Android Software Development Kit (SKD) สุดท้ายติดตั้งโมดูลสกัดฟีเจอร์ลงบนอุปกรณ์มือถือเพื่อทดสอบผลลัพธ์จากการสกัดข้อมูล

ในระหว่างวันที่ 23 เมษายน 2559 ถึง 22 กันยายน 2559 (รวมระยะเวลา 6 เดือน) คณะผู้ดำเนินโครงการได้ออกแบบและพัฒนาอัลกอริทึมในการตรวจจับแอปพลิเคชั่นอันตราย พัฒนาโมดูลตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์มือถือโดยพัฒนา APIs จาก Android Software Development Kit (SKD) และทดสอบโมดูลตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์มือถือเพื่อทดสอบการทำงาน

ในระหว่างวันที่ 23 กันยายน 2559 ถึง 22 มีนาคม 2560 (รวมระยะเวลา 6 เดือน) คณะผู้ดำเนินโครงการได้ประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชันอันตราย(แอลฟ่าเวอร์ชัน)ในห้องปฏิบัติการ พัฒนาเว็บไซต์ของระบบเพื่อเป็นช่องทางในการสื่อสารกับผู้ใช้ และประเมินผลที่ได้จากการประเมินความสำเร็จของแอลฟ่าเวอร์ชัน มาปรับปรุงระบบเป็นเบต้าเวอร์ชัน

ในระหว่างวันที่ 23 มีนาคม 2560 ถึง 21 กันยายน 2560 (รวมระยะเวลา 6 เดือน) คณะผู้ดำเนินโครงการได้ประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชันอันตรายของระบบเมื่อนำไปใช้จริง ซึ่งประกอบด้วยผลการอัพโหลดแอปพลิเคชั่นใน Google Play เพื่อให้บุคคลทั่วไปสามารถดาวโหลดไปใช้งานได้ ผลการทดสอบกับผู้ใช้ทั่วไปโดยใช้แบบประเมินแบบออนไลน์ และประมวลผลตอบรับที่ได้จากการนำไปใช้จริง

โครงการนี้เน้นระเบียบวิธีการวิจัยและปฏิบัติทดลองเพื่อพัฒนาระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย ได้ใช้เทคโนโลยีอุปกรณ์เคลื่อนที่ เทคโนโลยีเครือข่ายคอมพิวเตอร์และหลักการปัญญาประดิษฐ์ เป็นแนวทางในการวิจัยและพัฒนาโดยมุ่งเน้นให้สามารถนำไปใช้งานได้จริง มีวิธีการและขั้นตอนดังนี้ (ตัวเลขในวงเล็บด้านหลังใช้อ้างอิงในแผนดำเนินงาน)

1. ศึกษาและรวบรวมข้อมูลเกี่ยวกับมัลแวร์บนอุปกรณ์เคลื่อนที่ที่ตรวจพบและมีรายงานจนถึงปัจจุบัน (1)

2. ศึกษาและรวบรวมแอปพลิเคชั่นปกติที่นิยมใช้บนระบบปฏิบัติการแอนดรอยด์ (2)

3. ศึกษามัลแวร์ที่มีคุณสมบัติดาวน์โหลดอัตโนมัติได้ และจำลองสถานการณ์ขึ้นเพื่อสร้างมัลแวร์ประเภทนี้เพื่อใช้ในการศึกษาในขั้นตอนถัดไป (3)

4. ขั้นตอนการศึกษาและวิเคราะห์พฤติกรรมของแอปพลิเคชั่นขณะทำงานอยู่บนอุปกรณ์เคลื่อนที่

4.1 สร้างโมดูลสกัดฟีเจอร์ขณะใช้งานแอปพลิชั่นโดยพัฒนา APIs จาก Android Software Development Kit (SKD) เพื่อเก็บข้อมูลการเชื่อมต่อเครือข่าย เช่น ชนิดของเครือข่ายที่เชื่อมต่อ จำนวนข้อมูลที่รับ-ส่ง โหมดที่ใช้ในการรับ-ส่งเป็นต้น (4)

4.2 สร้างโมดูลประมวลผลฟีเจอร์ด้วยเทคนิคทางสถิติโดยพัฒนา APIs จาก Android Software Development Kit (SKD) (5)

4.3 ติดตั้งโมดูลสกัดฟีเจอร์ลงบนอุปกรณ์มือถือเพื่อทดสอบผลลัพธ์จากการสกัดข้อมูล (6)

4.4 เก็บข้อมูลที่ได้จากโมดูลสกัดฟีเจอร์โดยทำการติดตั้งบนอุปกรณ์มือถือจริง และใช้เวลาในการเก็บข้อมูลเริ่มตั้งแต่ 4 สัปดาห์ (รวมระยะเวลาทั้งหมดประมาณ 4 เดือน) (7)

5. ขั้นตอนการสร้างอัลกอริทึมในการแยกแยะแอปพลิเคชั่นอันตรายจากแอปพลิเคชั่นปกติ

5.1 ตั้งและทดสอบสมมติฐานเกี่ยวกับตัวแปรสำคัญที่มีผลต่อการแยกแยะแอปพลิเคชั่นอันตรายจากแอปพลิเคชั่นปกติ (8)

5.2 ออกแบบและพัฒนาอัลกอริทึมในการตรวจจับแอปพลิเคชั่นอันตราย (9)

5.3 พัฒนาโมดูลตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์มือถือโดยพัฒนา APIs จาก Android Software Development Kit (SKD) (10)

5.4 ทดสอบโมดูลตรวจจับแอปพลิเคชั่นอันตรายบนอุปกรณ์มือถือเพื่อทดสอบการทำงาน (11)

6. ประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตราย (แอลฟ่าเวอร์ชั่น)

6.1 ออกแบบการทดสอบและแบบประเมินความพึงพอใจของผู้ใช้ (12)

6.2 ทำการทดสอบกับผู้ใช้เฉพาะกลุ่มและส่งแบบประเมินแบบออนไลน์ (13)

6.3 รวบรวมผลตอบรับที่ได้และประมวลผลข้อมูล (14)

7. พัฒนาเว็บไซต์ของระบบเพื่อเป็นช่องทางการสื่อสารกับผู้ใช้ (15)

8. นำผลการประเมินที่ได้จากการประเมินความสำเร็จของแอลฟ่าเวอร์ชั่น มาปรับปรุงระบบเป็นเบต้าเวอร์ชั่น ในด้านต่างๆ ดังต่อไปนี้

8.1 อินเทอร์เฟซกับผู้ใช้ (16)

8.2 ประสิทธิภาพของระบบ (17)

9. ประเมินความสำเร็จของระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเมื่อนำไปใช้จริง (เบต้าเวอร์ชั่น)

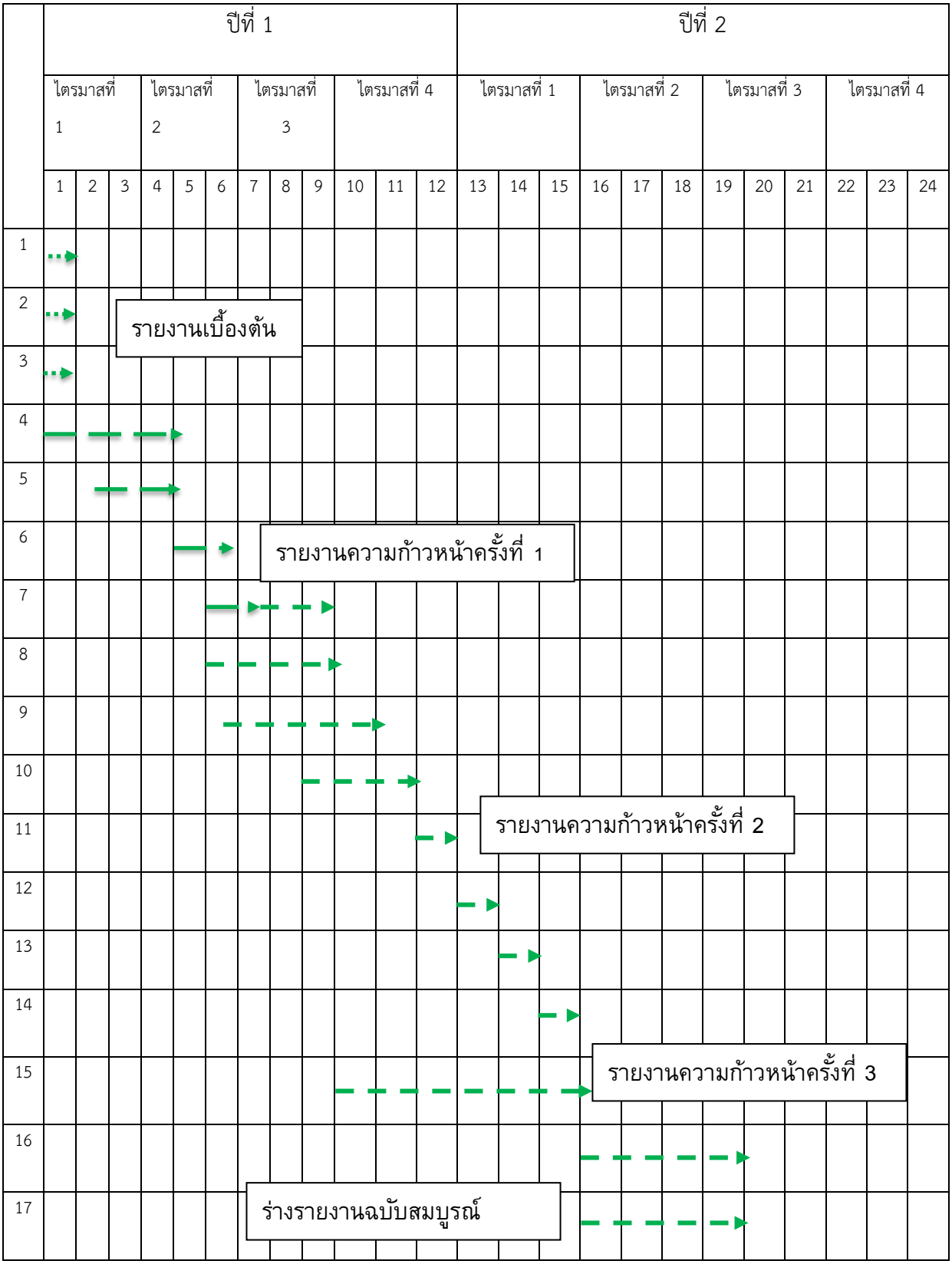
9.1 อัพโหลดแอปพลิเคชั่นใน Google Play เพื่อให้บุคคลทั่วไปสามารถดาวโหลดไปใช้งานได้ (18)

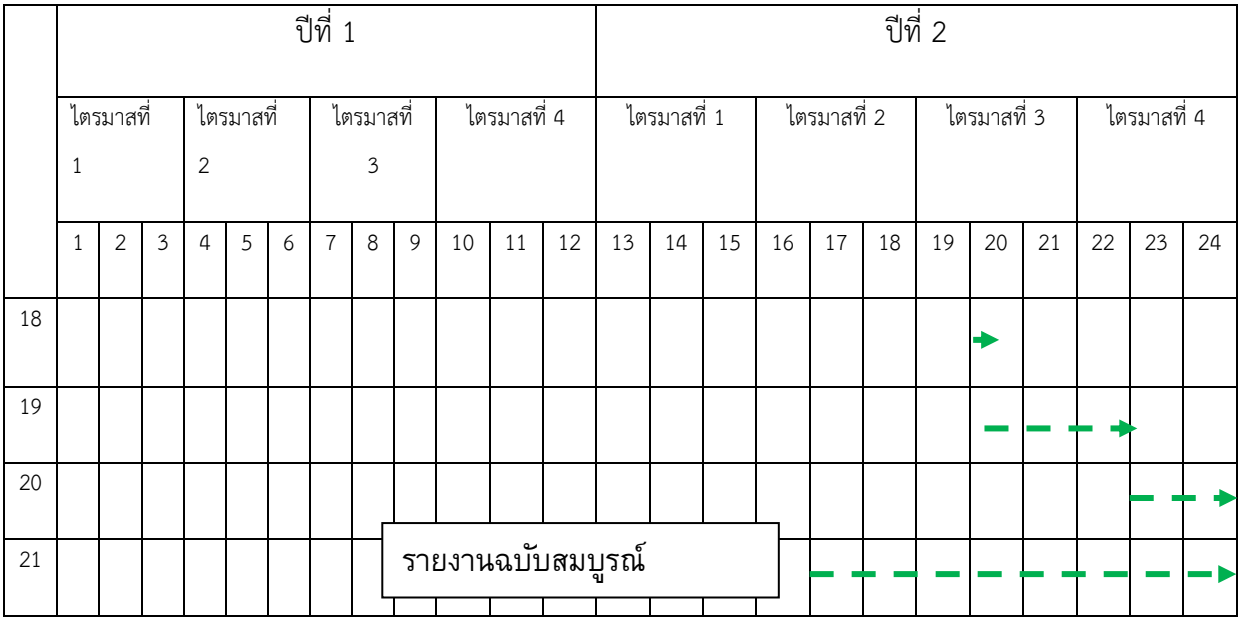
9.2 ทำการทดสอบกับผู้ใช้ทั่วไปและส่งแบบประเมินแบบออนไลน์ (19)

9.3 รวบรวมผลตอบรับที่ได้และประมวลผลข้อมูล (20)

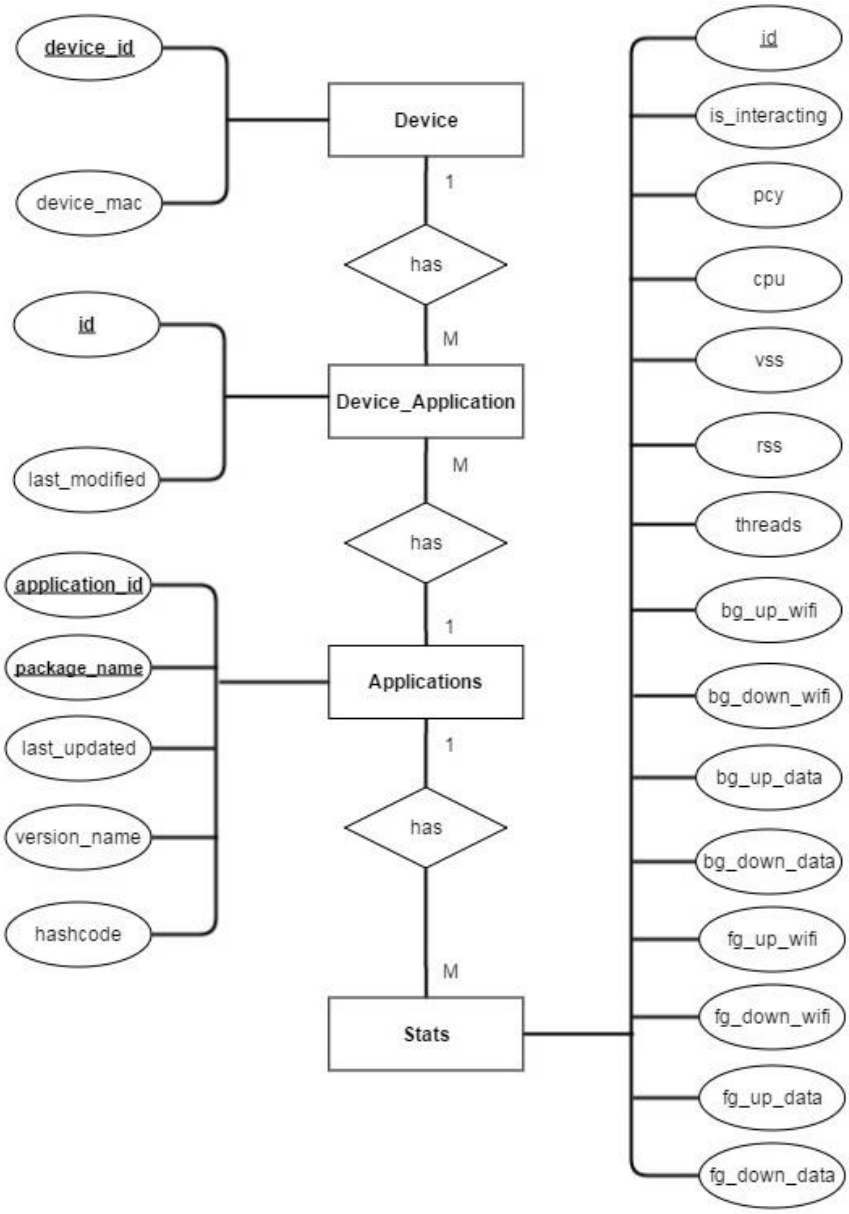
10. จัดทำเอกสารประกอบการส่งผลงานโครงการ (21)

ตารางที่ 5: การดำเนินงานที่ได้ดำเนินการแล้วและแผนดำเนินการตลอดทั้งโครงการ




ปีที่ 1
ปีที่ 2
ไตรมาสที่ 1
ไตรมาสที่ 2
ไตรมาสที่ 3
ไตรมาสที่ 4
ไตรมาสที่ 1
ไตรมาสที่ 2
ไตรมาสที่ 3
ไตรมาสที่ 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
18
19
20
21
รายงานฉบับสมบูรณ์

ภาคผนวกที่ ก การออกแบบฐานข้อมูลและพจนานุกรมข้อมูล



ภาคผนวกที่ ข ตระกูลของโค้ดอันตรายและคำอธิบายพฤติกรรม

#	ตระกูล	จำนวนของแอปพลิเคชัน	ประเภทความอันตรายและคำอธิบาย
1	Basebridge	19	- ส่งข้อความ SMS ในอัตราค่าบริการพิเศษอัตโนมัติโดยที่ผู้ใช้ไม่รู้ตัว (premium-rate SMS messages).
2	GEINIMI	23	- ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้
3	Kmin	18	- ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้
4	PJApps	14	- เปิดช่องโหว่บนระบบปฏิบัติการเพื่อเปิดโอกาสให้ผู้ไม่ประสงค์ดีมาขโมยข้อมูลจากอุปกรณ์เคลื่อนที่
5	ZergRush	6	- เป็นอันตรายต่อระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 2.2 และ 2.3 ที่มีช่องโหว่ในส่วนของการให้สิทธิซุปเปอร์ยูเซอร์ (Root)
6	YZHC	2	- ส่งข้อความ SMS ในอัตราค่าบริการพิเศษอัตโนมัติโดยที่ผู้ใช้ไม่รู้ตัว (premium-rate SMS messages) และบล็อกข้อความแจ้งเตือนผู้ใช้เกี่ยวกับเหตุการณ์ที่เกิดขึ้น - ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้

#	ประเภท	จำนวนของแอปพลิเคชัน	ประเภทความอันตรายและคำอธิบาย
7	Adware	15	- สืบค้นข้อมูลของผู้ใช้บนอุปกรณ์เพื่อใช้ในการแสดงโฆษณาบนเว็บบราวเซอร์
8	Steek	16	- ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้ - สืบค้นข้อมูลของผู้ใช้บนอุปกรณ์เพื่อใช้ในการแสดงโฆษณาบนเว็บบราวเซอร์
9	TrojanFakeNotifier	20	- ส่งข้อความ SMS ในอัตราค่าบริการพิเศษอัตโนมัติโดยที่ผู้ใช้ไม่รู้ตัว (premium-rate SMS messages).
10	Zitmo	7	- ขโมยข้อมูลส่วนตัวที่เกี่ยวข้องกับธุรกรรมทางการเงิน เช่น รหัสลับที่ธนาคารส่งมาที่โทรศัพท์ของผู้ใช้ เป็นต้น .
11	VdLoader	2	- เปิดช่องโหว่บนระบบปฏิบัติการเพื่อเปิดโอกาสให้ผู้ไม่ประสงค์ดีมาขโมยข้อมูลจากอุปกรณ์เคลื่อนที่
12	DroidKungFu	2	- ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้ .
13	AndroidGamex	1	- เปิดช่องโหว่บนระบบปฏิบัติการเพื่อเปิดโอกาสให้ผู้ไม่ประสงค์ดีมาขโมยข้อมูลจากอุปกรณ์เคลื่อนที่
14	LoozFon	2	- ขโมยข้อมูลส่วนตัวของผู้ใช้อุปกรณ์เคลื่อนที่และส่งไปยังเซิร์ฟเวอร์ที่กำหนดไว้
15	MMarketPay	1	- เปิดช่องโหว่เพื่อทำการจ่ายเงินซื้อแอปพลิเคชันจากตลาดแอนดรอยด์ของจีน

#	ประเภท	จำนวนของแอปพลิเคชัน	ประเภทความอันตรายและคำอธิบาย
16	Moghava	2	- เปิดช่องโหว่เพื่อทำตัดต่อรูปภาพทุกรูปที่อยู่ในอุปกรณ์กับภาพของนาย Ayatollah Khomeini (นักการเมืองอิหร่าน)
17	OPFake	4	- ส่งข้อความ SMS ในอัตราค่าบริการพิเศษอัตโนมัติโดยที่ผู้ใช้ไม่รู้ตัว (premium-rate SMS messages).
18	TigerBot	1	- ดาวโหลดและติดตั้งแอปพลิเคชันอื่นๆ ได้อัตโนมัติโดยผู้ใช้ไม่รู้ตัว - เปิดเว็บบราวเซอร์และดำเนินธุรกรรมบางอย่างได้ - อัพเดตไฟล์ไบนารีที่ติดตั้งตัวเองได้

ภาคผนวกที่ ค รายงานผลการทดสอบระบบตรวจจับและเตือนภัยแอปพลิเคชั่นอันตรายเวอร์ชั่นแอลฟ่า