การพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

Software Development of Intrusion Detection for Web Cache

โดย

นางสาวภัทราวดี เหมทานนท์

รหัส 44067013

อาจารย์ที่ปรึกษา

อาจารย์ อัครินทร์ คุณกิตติ

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2545
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช
นักศึกษา	นางสาวภัทราวดี เหมทานนท์
อาจารย์ที่ปรึกษา	อาจารย์อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

โปรแกรมเว็บพร็อกซี/แคชช่วยลดความคับคั่งของการสื่อสารและลดเวลาการร้องขอออบเจ็กต์ให้น้อยลงได้ แต่ปัญหาของโปรแกรมเว็บพร็อกซี/แคชอาจเกิดมาจาก client ภายในเครือข่ายที่มีหนอนอินเตอร์เน็ตทำงานแอบแฝงอยู่แล้วส่งการร้องขอออบเจ็กต์จน log file ของระบบเต็มทำให้โปรแกรมเว็บพร็อกซี/แคชหยุดทำงานได้ ดังนั้นจึงพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชโดยนำหลักการของการตรวจจับการโจมตี (Intrusion detection) มาช่วยตรวจสอบ log file ของโปรแกรมเว็บพร็อกซี/แคช โดยนำการร้องขอข้อมูลที่เก็บใน log file มาพิจารณาตามรูปแบบที่ได้กำหนดไว้ใน Pattern file โดยจะพิจารณา URL, IP address และMethod ในการร้องขอข้อมูลหากพบการร้องขอที่เข้าคู่กับรูปแบบใน Pattern file จะถือว่ามีโอกาสถูกโจมตีและจะเฝ้าพิจารณาจำนวนและเวลาของการร้องขอดังกล่าว หากพบจำนวนการร้องขอมากกว่าที่ได้กำหนดไว้และพิจารณาเวลาแล้วตรงกับรูปแบบที่กำหนดไว้จะส่งคำสั่งไปให้ไฟร์วอลล์เพื่อปฏิเสธการร้องขอนี้ นอกจากนี้โปรแกรมจะเฝ้าตรวจสอบการโจมตีดังกล่าวหากไม่พบการโจมตีจะส่งคำสั่งเพื่อยกเลิกการปฏิเสธการโจมตีไปยังไฟร์วอลล์เพื่อให้เครื่อง client นั้นสามารถทำงานได้ตามปกติ โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชพัฒนาบนระบบฟรีบีเอสดีและใช้ภาษาซีในการเขียนโปรแกรม ผลการทำงานของโปรแกรมสามารถตรวจจับการโจมตีและส่งคำสั่งเพื่อปฏิเสธการร้องขอข้อมูลไปยังไฟร์วอลล์ได้และสั่งยกเลิกได้เมื่อการโจมตีสิ้นสุดลง โปรแกรมนี้สามารถนำมาใช้ป้องกันการโจมตีจากภายในเครือข่ายได้โดยใช้งานร่วมกับไฟร์วอลล์และโปรแกรม Squid

Title	Software Development of Intrusion Detection for Web Cache
Student	Miss Pattharawadee Hamtanon
Advisor	Mr. Akharin Khunkitti
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

Abstract

Web proxy / cache program help us to reduce transmission congestion and reduce respond of object requesting .But the problem of web proxy / cache programs may be occur if there are some client in the network that infected worm. The infection cause system log file become full and make the programs terminated. This project is implementation of a intrusion detection program for web cache have been develop by using intrusion detection principle to check and test log file of web proxy / cache program. The program will gather URL, IP, and Method of request that store in log file then considerate them with format that determined in pattern file. If request information match with format in pattern file, it can refer that attacking can be occur. Then this request will become a number and time monitored request. If number of request more than the number that determined in pattern file and time equal to determined format this program will send commands to firewall for deny the request. And if the program can't detect any attacking from the monitoring, it will send commands to firewall for cache requesting deny for make that client working normally. Intrusion detection program for web cache developed on FreeBSD and use C language for implementation. The program can detect attacking and send request denying command to firewall then discard the when the attacking disappear. This detection program can be implement to detect internal attacking by using together with firewall and squid.

กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ อาจารย์อัครินทร์ คุณกิตติ ซึ่งได้ให้คำปรึกษาทั้งแนวคิดและข้อเสนอต่างๆจนกระทั่งโครงการพัฒนาระบบงานนี้เสร็จสมบูรณ์ และขอขอบเพื่อนๆทุกท่านที่ให้ทั้งกำลังใจและความช่วยเหลือโดยตลอดมา

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

บทที่

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญภาพ

หน้า

ภาพที่

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการพัฒนาระบบ

ปัจจุบันความต้องการในการใช้งานเครือข่ายอินเตอร์เน็ตเพิ่มสูงมากขึ้นเนื่องจากอินเตอร์เน็ตเป็นแหล่งข้อมูลข่าวสารขนาดใหญ่ที่สามารถใช้ประโยชน์ทั้งทางด้านการสื่อสารและการค้นหาข้อมูล ซึ่งการใช้งานถึง 80 เปอร์เซ็นต์บนอินเตอร์เน็ตอยู่ในรูปแบบของ world wide web ซึ่งมีการส่งข้อมูลข่าวสารมักจะอยู่ในรูปของ ข้อมูลภาพ ข้อมูลเสียง เอกสาร HTML หรือแม้กระทั่งข้อมูลประเภทมัลติมีเดีย เป็นต้น ผลของการใช้งานของผู้ใช้เครือข่ายอินเตอร์เน็ตจำนวนมากทำให้เกิดการรับและการส่งข้อมูลเป็นจำนวนมากทำให้เกิดปัญหาความคับคั่งของการสื่อสาร เพื่อแก้ปัญหาดังกล่าวจึงมีผู้คิดค้นและนำโปรแกรมประเภทเว็บ Web Proxy/Cache Server มาติดตั้ง ใช้งานกับเครือข่ายขององค์กร นอกจากโปรแกรมประเภทเว็บ Web Proxy/Cache Server สามารถแก้ปัญหาความคับคั่งของการสื่อสารแล้วยังช่วยเพิ่มประสิทธิภาพในการใช้ช่องทางการสื่อสารที่มีจำนวนน้อยให้คุ้มค่าอีกด้วย

ถึงแม้ว่าจะนำเอาโปรแกรมประเภท Web Proxy/Cache Server มาใช้ในการแก้ปัญหาการคับคั่งของการสื่อสารได้ แต่อาจจะเกิดปัญหาเกี่ยวกับโปรแกรม Web Proxy/Cache ทางอ้อมได้เช่นการโจมตีของหนอนอินเตอร์เน็ตทำให้ log file ของ โปรแกรม Web Proxy/Cache ได้รับผลกระทบ กล่าวคือ หนอนอินเตอร์เน็ตที่แอบแฝงเข้าสู่เครือข่ายภายในขององค์กรทำให้เครื่อง client ภายในที่มีหนอนอินเตอร์เน็ตแฝงทำงานอยู่ส่งการร้องขอข้อมูลไปยังโปรแกรม Web Proxy/Cache มากจน log file เต็ม ส่งผลให้โปรแกรม Web Proxy/Cache หยุดการทำงานได้ในที่สุด

จากปัญหาการโจมตี log file ดังกล่าวซึ่งไม่สามารถป้องกันโดยใช้การกรองแพ็กเกตจาก ไฟร์วอลล์ (Firewall) เพียงอย่างเดียว ควรเพิ่มการรักษาความปลอดภัยโดยใช้งานไฟร์วอล์ร่วมกับระบบตรวจจับการบุกรุก (Intrusion Detection System : IDS) ระบบตรวจจับการบุกรุกที่นำมาใช้นี้จะเป็นประเภทที่นำข่าวสารในระดับแอพพลิเคชัน (Application-based) มาใช้ในการพิจารณา เนื่องจากโปรแกรม Web Proxy/Cache เป็นโปรแกรมในระดับชั้นแอพพลิเคชัน ดังนั้นการเลือกใช้ระบบตรวจจับการบุกรุกจึงควรเป็นการป้องกันในระดับชั้นแอพพลิเคชันด้วย หลักการของการตรวจจับการโจมตีดังกล่าวทำได้โดยการเฝ้าดูการร้องขอข้อมูลจาก log file โดยนำเอาข้อมูลการร้องขอมาเปรียบเทียบกับจำนวนและ

เวลาของการร้องขอกับข้อมูลการโจมตีที่ได้กำหนดไว้ หากพบว่าข้อมูลทั้งสองอย่างสอดคล้องกันจะถือว่าเป็นการโจมตีและจะทำการตอบสนองกับการโจมตีนั้นโดยการส่งสัญญาณไปให้ไฟร์วอลล์เพื่อปฏิเสธ(deny) การร้องขอข้อมูลของ client ที่เป็นต้นเหตุของการโจมตี และหน้าที่อีกประการหนึ่งของโปรแกรมตรวจจับการโจมตีคือหลังจากปฏิเสธการร้องขอแล้วจะต้องมีการตรวจสอบการโจมตีที่ไฟร์วอลล์หากไม่พบการโจมตีจะส่งคำสั่งให้ไฟร์วอลล์ยกเลิกการปฏิเสธการร้องขอข้อมูล เพื่อให้ client ต้นเหตุสามารถร้องขอข้อมูลได้ตามปกติ

1.2 เป้าหมายของการพัฒนาระบบงาน

โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชทำงานร่วมกับโปรแกรม Squid ซึ่งเป็นโปรแกรม Web Proxy/Cache ที่มีประสิทธิภาพ ทำงานโดยเฝ้าดูการร้องขอข้อมูลจาก access.log ซึ่งจะทำให้ทราบถึงความผิดปกติที่เกิดขึ้น และสามารถป้องกันการโจมตีจากการร้องขอข้อมูลที่มากเกินปกติอันเป็นเหตุให้โปรแกรม Web Proxy/Cache หยุดทำงาน

โปรแกรมการตรวจจับการโจมตีสำหรับเว็บแคชมีความสามารถในการทำงานเกี่ยวกับการป้องกันการโจมตีดังนี้

- เฝ้าตรวจสอบการร้องขอข้อมูลของ client จาก log file ของ โปรแกรม Squid ชื่อ access.log และตรวจจับการโจมตีโดยเปรียบเทียบกับเงื่อนไขที่กำหนดไว้ในแฟ้มข้อมูล
- เมื่อทราบว่ามีการโจมตีจาก client ใดจะปฏิเสธการร้องขอข้อมูลโดยส่งคำสั่งเพื่อปฏิเสธการร้องขอข้อมูลไปที่ ไฟร์วอลล์ หรือคำสั่งอื่นๆในลักษณะเดียวกัน
- โปรแกรมสามารถตรวจสอบการสิ้นสุดการโจมตีและคำสั่งเพื่อสั่งให้ ไฟร์วอลล์ยกเลิกปฏิเสธการร้องขอให้เป็นไปตามปกติ
- ผู้ใช้โปรแกรมสามารถทำการกำหนดเงื่อนไขการโจมตีต่างๆได้ในแฟ้มข้อมูลสำหรับเก็บข้อมูลเพื่อใช้ในการเปรียบเทียบกับการร้องขอที่ผิดปกติได้
- โปรแกรมสามารถเก็บผลการทำงานไว้ในแฟ้มข้อมูล (history file)

1.3 วัตถุประสงค์ของการพัฒนาระบบงาน

- เพื่อศึกษาหลักการทำงานของการตรวจจับการบุกรุกและโปรแกรม Web Proxy/Cache และส่วนที่เกี่ยวข้องกับโครงงาน
- เพื่อวิเคราะห์และออกแบบระบบการตรวจจับการโจมตีของโปรแกรม Web Proxy/Cache

- เพื่อลดปัญหาการโจมตีที่ทำให้ log file ของโปรแกรม Web Proxy/Cache เต็มเนื่องจากการร้องขอที่มากเกินจำเป็น

1.4 ขอบเขตของการพัฒนาโปรแกรม

การพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชจะแบ่งการทำงานสามส่วนหลักๆ คือ

- ส่วนของการเฝ้าดูการโจมตีซึ่งจะตรวจสอบจาก log file ของโปรแกรม Squid โดยทำการเปรียบเทียบเงื่อนไขการโจมตีจากแฟ้มข้อมูลรูปแบบการโจมตี (pattern file) ซึ่งผู้ใช้โปรแกรมเป็นผู้กำหนดและเมื่อพบการโจมตีจะทำการปฏิเสธการโจมตีโดยส่งคำสั่งไปยังไฟร์วอลล์เพื่อปฏิเสธการร้องขอข้อมูล
- ส่วนของการยกเลิกการปฏิเสธการร้องขอข้อมูลเพื่อจำกัดการโจมตี เมื่อตรวจสอบตามเวลาที่ได้กำหนดไว้ในแฟ้มข้อมูล รูปแบบการโจมตี (pattern file) และไม่พบการโจมตีจะส่งสัญญาณยกเลิกการปฏิเสธการร้องขอข้อมูลไปยังไฟร์วอลล์
- ส่วนของการตรวจสอบข้อมูลที่หมดเวลา (Time out) ระหว่างโปรแกรมทำงานจะมีการเก็บข้อมูลต่างๆเพื่อใช้ในการทำการตรวจสอบการโจมตีและการยกเลิกจำกัดการโจมตี ซึ่งอาจจะมีข้อมูลบางส่วนที่ไม่จำเป็นต่อการใช้งานเนื่องจากข้อมูลดังกล่าวสิ้นสุดเวลาการตรวจสอบการทำงาน ในโปรแกรมส่วนนี้จะทำการปรับปรุงข้อมูลเพื่อลดการใช้ทรัพยากร

การพัฒนาจะพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชจะพัฒนาภายใต้ระบบปฏิบัติการ freeBSD 4.7 และใช้โปรแกรม Squid ซึ่งเป็นโปรแกรม Web Proxy/Cache ส่วนการเขียนโปรแกรมจะพัฒนาโดยใช้ภาษาซี

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ระบบที่ใช้โปรแกรม Squid มีความปลอดภัยจากการโจมตี log file ซึ่งเกิดจากการร้องขอข้อมูลจากเครือข่ายภายใน นอกจากนี้ยังมีประโยชน์โดยตรงต่อผู้ดูแลระบบเนื่องจากได้รับความสะดวกจากโปรแกรมคือลดภาระการเพิ่มหรือลบกฎเพื่อจำกัดการร้องขอข้อมูลที่ผิดปกติด้วยตัวเองได้ และทำให้ป้องกันการโจมตีได้ทันท่วงทีเนื่องจากไม่ขึ้นอยู่กับผู้ดูแลระบบ

1.6 ขั้นตอนในการพัฒนาระบบงาน

- ศึกษาหลักการของการตรวจจับการโจมตีและศึกษารูปแบบของ log file ของ โปรแกรม Squid

- ศึกษาปัญหาที่เกิดขึ้นจากการโจมตี log file ของโปรแกรม Squid
- ศึกษาแนวทางในการแก้ไขปัญหา และทฤษฎีต่างๆ ที่เกี่ยวข้อง
- ทำการวิเคราะห์ และออกแบบโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช
- ศึกษาเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบงาน
- พัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช
- ทดสอบการใช้งาน และปรับปรุงแก้ไขโปรแกรมที่พัฒนาแล้ว
- สรุปผลการทดสอบจากการใช้งานที่เกิดขึ้น
- จัดทำเอกสารประกอบโครงงาน

1.7 เนื้อหาของแต่ละบท

- บทที่ 2 หลักการและความรู้ที่เกี่ยวข้องกับการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช ประกอบด้วย ระบบตรวจจับการบุกรุก โปรแกรม Squid และการกำหนดกฎในไฟร์วอลล์
- บทที่ 3 การออกแบบระบบงานของการตรวจจับการโจมตีสำหรับเว็บแคชซึ่งจะประกอบไปด้วย สถานะต่างๆของระบบ, ผังการไหลของข้อมูล, รูปแบบของคอนฟิกไฟล์ และรูปแบบของ file ต่างๆ ที่เกี่ยวข้องกับโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช
- บทที่ 4 กล่าวถึงแสดงภาษา คอมไพเลอร์ และเครื่องมือต่างๆที่ใช้งานในการพัฒนาโปรแกรม
- บทที่ 5 เป็นการทดสอบการใช้โปรแกรมและผลที่ได้จากการทดสอบ รวมทั้งบทสรุปของการพัฒนาโปรแกรม และข้อเสนอแนะ

บทที่ 2
ระบบตรวจจับการบุกรุก เว็บพร็อกซี/แคชและไฟร์วอลล์

การสร้างโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชนั้นต้องเข้าใจหลักการและความรู้ที่เกี่ยวข้องเพื่อเป็นแนวทางในการออกแบบโปรแกรม หลักการและความรู้ที่นำมาใช้ในการออกแบบระบบประกอบด้วย ระบบตรวจจับการบุกรุก (Intrusion Detection System) ซึ่งจะกล่าวถึงความหมายและประเภทต่างๆของระบบตรวจจับการบุกรุกรวมทั้ง Framework ที่ใช้เป็นพื้นฐานสำคัญของงานวิจัยและสำหรับศึกษาการตรวจสอบการบุกรุก นอกจากระบบตรวจจับการบุกรุกซึ่งเป็นหลักการสำคัญในการพัฒนาระบบแล้วโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชยังเกี่ยวข้องกับโปรแกรม Web Proxy/Cache โดยเลือกใช้โปรแกรม Squid เป็นโปรแกรม Web Proxy/Cache เนื่องจาก Squid เป็นโปรแกรมที่มีประสิทธิภาพและได้รับความนิยมในการใช้งาน ดังนั้นจะกล่าวถึงรายละเอียดของโปรแกรม Web Proxy/Cache และโปรแกรม Squid เฉพาะส่วนที่เกี่ยวข้องกับการนำมาใช้ในการพัฒนาโปรแกรม นอกจากความรู้พื้นฐานสองส่วนดังกล่าวแล้วโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชยังทำงานประสานงานกับไฟร์วอลล์เพื่อส่งคำสั่งให้ปฏิเสธการโจมตีที่ตรวจพบ กล่าวคือเมื่อโปรแกรมตรวจพบการโจมตี โปรแกรมจะส่งคำสั่งปฏิเสธการร้องขอข้อมูลของ client ต้นเหตุไปยังไฟร์วอลล์เพื่อให้ไฟร์วอลล์ปฏิเสธการร้องขอที่มากเกินจนเป็นการโจมตี นอกจากนี้โปรแกรมจะตรวจสอบการโจมตีโดยการสอบถามไปยังไฟร์วอลล์จนกระทั่งไม่พบการโจมตีจึงส่งคำสั่งยกเลิกการปฏิเสธการร้องขอข้อมูลไปยังไฟร์วอลล์เพื่ออนุญาตให้ client ต้นเหตุสามารถร้องขอข้อมูลได้ตามปกติ ดังนั้นความรู้เกี่ยวกับกฎและการกำหนดกฎของไฟร์วอลล์จึงจำเป็นสำหรับการพัฒนาโปรแกรมนี้ด้วย หลักการและความรู้ทั้งสามหัวข้อมีรายละเอียดดังหัวข้อต่อไปนี้

2.1 ระบบตรวจจับการบุกรุก (Intrusion Detection System)

ระบบตรวจจับการบุกรุก คือ ซอฟต์แวร์หรือฮาร์ดแวร์ที่คอยเฝ้าดู (monitor) เหตุการณ์หรือสิ่งผิดปกติที่เกิดขึ้นในระบบเครือข่ายหรือบนเครื่องเป้าหมาย แล้วทำการวิเคราะห์เหตุการณ์ที่ทำให้เกิดปัญหาต่อความปลอดภัย พยายามตรวจจับและเตือนภัยเมื่อผู้บุกรุกพยายามที่จะเข้ามาในระบบหรือเครือข่าย รายละเอียดในส่วนต่างๆของระบบตรวจจับการบุกรุกจะอธิบายหัวข้อถัดไปดังนี้

2.1.1 ประเภทของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุกสามารถแบ่งได้หลายลักษณะขึ้นอยู่กับหลักเกณฑ์ที่ใช้ในการแบ่งจากการศึกษาพบว่าสามารถแบ่งโดยใช้เกณฑ์ทั้งหมด 5 เกณฑ์ดังต่อไปนี้

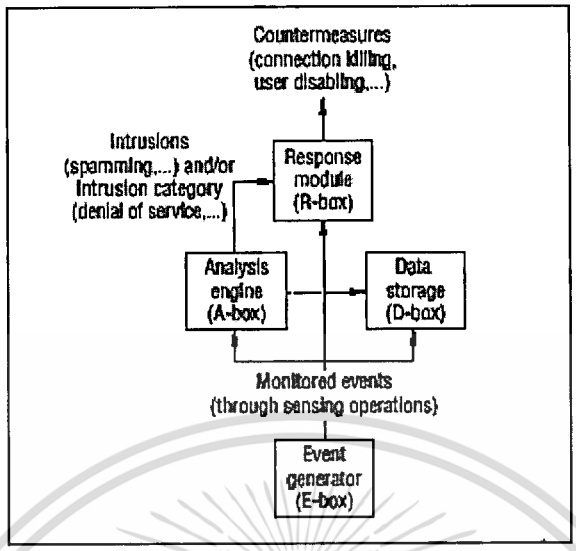
- แบ่งตามแหล่งกำเนิดข้อมูลสามารถแบ่งเป็น 4 ลักษณะ ได้แก่
  - พิจารณาข่าวสารจากเครือข่าย (Network-based) เป็นระบบตรวจจับการบุกรุกที่ทำงานในระดับเครือข่าย โดยเก็บรวบรวมและวิเคราะห์ข้อมูลจากเครือข่ายเพื่อหาผู้บุกรุก
  - พิจารณาข่าวสารจากเครื่องโฮสต์ (Host-based) เป็นการใช้ประโยชน์จากข่าวสารที่เก็บรวบรวมใน operating system audit trail และ system log file ของระบบคอมพิวเตอร์ เพื่อนำมาวิเคราะห์หาการบุกรุก
  - พิจารณาข่าวสารในระดับแอพพลิเคชัน (Application-based) จะเป็นกลุ่มย่อยของ Host-based เนื่องจากใช้แหล่งข้อมูลลักษณะเดียวกันคือจะใช้ log file แต่แตกต่างจาก log file ของ Host-based คือในระดับแอพพลิเคชันจะใช้เฉพาะ log file ของแอพลิเคชันเฉพาะตัวที่สนใจเท่านั้น เช่น log file ของโปรแกรม web server เป็นต้น
  - พิจารณาข่าวสารจากการสร้างข่าวสารเอง (Target-based) จะแตกต่างจากทั้ง 3 แหล่งข้อมูลข้างต้นเนื่องจากจะใช้ข่าวสารที่สร้างขึ้นเองโดยใช้การเข้ารหัสแบบ hash function เพื่อตรวจจับการเปลี่ยนแปลงต่อระบบของ object และจะเปรียบเทียบกับนโยบาย
- แบ่งตามการวิเคราะห์ข้อมูลเป็น 2 ลักษณะ ได้แก่
  - การตรวจจับแบบเทียบพฤติกรรมผู้ใช้กับรูปแบบการบุกรุก (misuse detection) เป็นเทคนิคที่ไม่ซับซ้อน สามารถตรวจจับโดยการเปรียบเทียบพฤติกรรมของผู้ใช้กับรูปแบบการโจมตีที่มี ดังนั้นจึงไม่มีปัญหาเกี่ยวกับการเตือนภัยผิดพลาดที่ส่งไปยังผู้ดูแลระบบ อีกทั้งยังรวดเร็วแต่ต้องมีการปรับปรุงข้อมูลการโจมตีให้ทันสมัยเสมอ
  - การตรวจจับแบบวิเคราะห์พฤติกรรมที่ผิดปกติ (anomaly detection) สามารถกำหนดพฤติกรรมที่ไม่ปกติบน host หรือบนเครือข่ายได้เอง โดยคาดหมายว่าพฤติกรรมที่ต้องการตรวจสอบทั้งหมดเป็นการบุกรุก โดยจะรวบรวมข้อมูลพฤติกรรมปกติของผู้ใช้ซึ่งอาจรวบรวมมาจากโฮสต์หรือเครือข่าย แล้วจึงใช้วิธีการวัดแบบต่างๆ เพื่อพิจารณาพฤติกรรมที่ต้องการตรวจสอบว่าเป็นการบุกรุกหรือไม่ การวัดมีด้วยกันวิธีหลายวิธี เช่น ใช้วิธีทางสถิติ (Statistical measures) วิธีวัดโดยการนับจำนวนครั้ง (Threshold detection) เช่นนับจำนวนการ login เข้าสู่ระบบ วิธีการใช้กฎ (Rule-based measures) เป็นต้น การตรวจจับการบุกรุกประเภทนี้ถึงแม้จะ

สามารถกำหนดพฤติกรรมที่ผิดปกติได้เองแต่ข้อเสียคือมีความผิดพลาดในวิเคราะห์และเตือนภัยต่อการบุกรุกสูง

- แบ่งตามเวลาที่ใช้ในการวิเคราะห์(Timing)[3] ซึ่งแบ่งได้ 2 ลักษณะคือ
  - แบบช่วงเวลา (batch) ข้อมูลถูกส่งเข้าสู่ตัววิเคราะห์อย่างไม่ต่อเนื่อง
  - แบบเรียลไทม์ (real-time) ข้อมูลถูกส่งเข้าสู่อย่างต่อเนื่อง
- แบ่งตามการตอบสนองต่อผู้บุกรุก (Response Operation) มี 2 ลักษณะคือ
  - แบบโต้ตอบ (Active Responses) เมื่อระบบตรวจจับการบุกรุกตรวจพบการบุกรุก ระบบจะกระทำการบางอย่างกับการบุกรุก เช่น การปิดบริการที่ถูกบุกรุก การปิดช่องโหว่ที่เกิดขึ้นเป็นต้น
  - แบบไม่โต้ตอบ (Passive Responses) คือเมื่อตรวจพบการบุกรุกจะทำเพียงส่งสัญญาณเตือนแจ้งไปยังผู้ดูแลระบบเท่านั้น
- แบ่งตามแผนการควบคุม (Control Strategy)
  - แบบรวมศูนย์ (Centralized) การทำงานของระบบตรวจจับการบุกรุกแบบรวมศูนย์ได้แก่ การเฝ้าดูการบุกรุก การตรวจจับ และ การรายงานจะขึ้นตรงต่อส่วนกลาง
  - แบบกระจายศูนย์ (Distributed) การทำงานในลักษณะนี้จะมีการเฝ้าดูการบุกรุกและตรวจจับซึ่งเป็นหน้าที่ของโฮสต์อื่นๆ ภายในเครือข่ายและจะส่งผลการตรวจจับไปให้ส่วนกลาง

2.1.2 เฟรมเวอร์คของระบบตรวจจับการบุกรุก

เนื่องจากระบบตรวจจับการบุกรุกได้รับความสนใจมากขึ้น ทำให้เกิดแนวคิดใหม่ๆ มากมายหลากหลาย ดังนั้นกลุ่มนักวิจัยจึงทำการกำหนดแบบจำลอง CIDF (Common Intrusion Detection Framework) ขึ้น โดยที่เฟรมเวอร์ค CIDF จะใช้เป็นพื้นฐานที่สำคัญของการวิจัยและการศึกษาการตรวจสอบการบุกรุก



ภาพที่ 2.1 Common Intrusion Detection Framework

ลักษณะของ CIDF แสดงดังภาพที่ 2.1 ซึ่งประกอบไปด้วย 4 ส่วนคือ

- Event generator (E-box) รับข่าวสารที่เกี่ยวข้องกับกระแสของเหตุการณ์(event stream) จากระบบเป้าหมาย
- Analysis engine (A-box) เป็นส่วนของการวิเคราะห์ข่าวสารที่ได้รับมาจาก E-box
- Data storage (D-box) เป็นส่วนบันทึกข่าวสารและสามารถดึงข่าวสารมาใช้ได้ในเวลาต่อมา
- Response module (R-box) เป็นส่วนที่ทำหน้าที่โต้ตอบต่อการบุกรุกหรือบล็อกผู้บุกรุก เพื่อป้องกันอันตรายที่จะเกิดขึ้น เช่น การยกเลิกการติดต่อกับผู้ใช้งานเมื่อตรวจพบความผิดปกติ การทำงานของ R-box อาจเกิดความผิดพลาดได้และส่งผลให้การเกิดโต้ตอบหรือการเตือนภัยผิดพลาด (false alarm) เนื่องจากผลของการทำงานที่ผิดพลาดของ A-box ในการวิเคราะห์การบุกรุก

2.2 เว็บพร็อกซี่/แคช (Web Proxy / Cache)

การใช้งานที่เพิ่มขึ้นของเครือข่ายอินเตอร์เน็ตเป็นเหตุให้ต้องหาวิธีการใช้ทรัพยากรทางด้านเครือข่ายให้เป็นไปอย่างมีประโยชน์สูงสุดจึงมีการนำ Web proxy /Cache Server มาช่วยแบ่งเบาภาระงานโดยการลดจำนวนการร้องขอ ข้อมูลภายในเครือข่ายได้ ดังนั้นในหัวข้อนี้จะเป็นการอธิบายหลักการทำงานของ Web Proxy/Cache Server และจะอธิบายถึงการทำงานร่วมกันระหว่าง Cache หลายๆตัว รวมทั้งโปรโตคอลที่ Web Proxy/Cache Server ที่สามารถรองรับได้ ส่วนสุดท้ายในหัวข้อ

Web Proxy / Cache จะกล่าวถึงโปรแกรม Squid ซึ่งเป็นโปรแกรมที่ใช้ในการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

2.2.1 หลักการทำงานของ Web Proxy/Cache Server

การทำงานของ Web Proxy Cache สามารถแบ่งได้เป็นสองส่วนดังต่อไปนี้

- ส่วนแรกเป็นส่วนของ proxy ทำหน้าที่เป็นตัวแทนของ client เช่น Browser, Ftp client หรือโปรโตคอลอื่นที่สนับสนุน เพื่อร้องขอออบเจ็กต์ต่างๆ เช่น ภาพ เอกสาร HTML เสียง ในรูปแบบของ URL
- ส่วนที่สองเป็นส่วนของ Cache ทำหน้าที่เก็บออบเจ็กต์ต่างๆที่ client เคยร้องขอ การเก็บออบเจ็กต์ดังกล่าวจะเก็บอย่างเป็นระเบียบเพื่อความสะดวกในการค้นหาเมื่อมีการร้องขอออบเจ็กต์ซ้ำซ้อน

Web Proxy Cache ทำงานในลักษณะ Client/Server คือ สามารถทำหน้าที่เป็นได้ทั้ง Client และ Server โดยจะทำหน้าที่เป็น Server เมื่อมี client ร้องขอข้อมูลผ่านโปรโตคอล HTTP, FTP หรือโปรโตคอลอื่นๆ ที่ Web Proxy Cache นั้นสนับสนุน ส่วนการทำงานของโปรแกรม Web Proxy Cache ที่ทำหน้าที่เป็น client นั้นจะหมายถึงโปรแกรม Web Proxy Cache ทำหน้าที่เป็นตัวแทนของ client ในการติดต่อร้องขอข้อมูลไปยัง Web Server จริงที่ให้บริการ (Original Server) และหลังจากได้รับออบเจ็กต์แล้ว Web Proxy Cache จะทำการสำเนาออบเจ็กต์เก็บไว้และเก็บค่าต่างๆ ลงใน log file เพื่อใช้ในการเปรียบเทียบการร้องขอออบเจ็กต์ในครั้งต่อไป กล่าวคือเมื่อ Web Proxy Cache ได้รับการร้องขอออบเจ็กต์จาก client จะตรวจสอบข้อมูลการร้องขอเพื่อพิจารณาการทำงานต่อไปโดยหากพบว่าออบเจ็กต์ที่ร้องขอมีอยู่ในแคชและเป็นข้อมูลที่ไม่ล้าสมัยก็จะนำออบเจ็กต์นั้นส่งกลับไปให้ client ซึ่งไม่จำเป็นต้องติดต่อไปยัง Web Server จริงที่ให้บริการ(Original Server) ดังนั้นจะเห็นได้ว่าโปรแกรม Web Proxy Cache มีประโยชน์เมื่อมีการร้องขอออบเจ็กต์ที่ซ้ำซ้อนกัน

2.2.2 โปรโตคอลที่ Web Proxy Cache รองรับ

โปรแกรม Web Proxy Cache สามารถรองรับการทำงานร่วมกับโปรโตคอลได้หลายตัว ทั้งนี้ขึ้นอยู่กับการสนับสนุนของแต่ละโปรแกรมที่จะรองรับการทำงานโปรโตคอลใดบ้าง โดยทั่วไปสามารถแบ่งลักษณะของโปรโตคอลได้เป็น 2 ลักษณะตามการใช้งานคือ

- โปรโตคอลที่ให้บริการผู้ใช้งานเช่น HTTP, FTP, Gopher, WAIS, SSL เป็นต้น

- โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่าง Cache เช่น HTTP, ICP, Cache Digest, SNMP, HTCP เป็นต้น

2.2.3 ลักษณะการทำงานร่วมกันของ Cache หลายตัว

ในปัจจุบัน Web Proxy Cache เป็นโปรแกรมที่นิยมใช้กันอย่างแพร่หลายในองค์กร ซึ่งในแต่ละองค์กรอาจมีการติดตั้ง Cache ไว้มากกว่าหนึ่งตัวโดยแต่ละตัวมีการทำงานร่วมกัน ดังนั้นจึงมีการจัดลักษณะการทำงานร่วมกันของ Cache ได้สองประเภทคือ

- Simple Web Cache เป็นการติดตั้ง Cache เป็นแบบลำดับชั้นในลักษณะ Simple tree คือ Cache ระดับล่างสามารถมี Cache ชั้นบนได้ไม่เกินหนึ่งตัว ลักษณะการทำงานของ Cache คือเมื่อ client ทำการร้องขอข้อมูลมายัง Cache ซึ่งเป็น Cache ระดับล่างหากตรวจสอบแล้วไม่พบออบเจ็กต์ตามที่ client ต้องการ Cache ระดับล่างนั้นจะร้องขอข้อมูลไปที่ Cache ระดับที่สูงกว่าเพื่อค้นหาข้อมูลตามที่ต้องการ หากไม่พบข้อมูลตามที่ต้องการจะทำการติดต่อไปยังเครื่องที่ให้บริการเว็บต้นทางเพื่อร้องขอข้อมูล เว็บแคชในลักษณะนี้มีข้อเสียคืออาจเกิดปัญหาความซ้ำซ้อนของข้อมูลในแต่ละระดับชั้น ดังนั้นจึงมีการปรับปรุงมาใช้เป็นแบบ Cooperate Web Cache
- Cooperate Web Cache เป็นการนำ Cache มาติดตั้งในลักษณะแบบกระจายโดยมีการแบ่งหน้าที่ความรับผิดชอบของ Cache เป็น Domain คือเมื่อมีการร้องขอข้อมูลจาก client ภายใน Domain ที่ Cache ตัวใดรับผิดชอบก็จะรับการร้องขอนั้นมาตรวจสอบหากพบออบเจ็กต์ตามที่ร้องขอมาและยังไม่หมดอายุก็สามารถจัดส่งให้ตามความต้องการ แต่หากไม่พบก็จะส่ง Message ไปถาม Cache ตัวอื่นๆว่ามีออบเจ็กต์ตามที่ต้องการหรือไม่ หากมี Cache ตอบกลับมาหลายตัวจะทำการเปรียบเทียบเวลาที่สั้นที่สุดที่สามารถรับออบเจ็กต์นั้น หากไม่มีใน Cache ตัวใดเลยก็จะส่งการร้องขอไปยัง Web Server จริงที่ให้บริการ(Original Server)

ทั้งสองลักษณะมีโปรโตคอลที่ทำการติดต่อสื่อสารระหว่าง Cache โดยทำการติดต่อแลกเปลี่ยนค่ากำหนดที่ใช้ในการทำงานภายในกลุ่มของ Proxy Web Server เช่น โปรโตคอล ICP เป็นโปรโตคอลที่ติดต่อสื่อสารระหว่าง Cache โดยใช้ ICP Message เพื่อแลกเปลี่ยนข้อมูลเกี่ยวกับ URLs ที่มีอยู่ใน Cacheใกล้เคียง

2.2.4 โปรแกรม Squid

โปรแกรม Web Proxy Cache ในปัจจุบันมีด้วยกันมากมาย แต่สำหรับการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชได้เลือกใช้ โปรแกรม Squid เนื่องจากมีประสิทธิภาพและเป็นฟรีโปรแกรมซึ่งสามารถหาใช้งานได้ง่าย

โปรแกรม Squid เป็น Internet proxy caching program ซึ่งเกิดจากพัฒนาของ Harvest project และได้รับการสนับสนุนจาก National Laboratory of Network Research (NLANR) โปรแกรม Squid สนับสนุนโปรโตคอลต่างๆเช่น FTP, gopher และ ข้อมูลแบบ HTTP และโปรโตคอลที่สามารถใช้ในการอ้างถึงแบบ URL เช่น SSL อีกทั้งยังสามารถทำงานแบบ cache hierarchies โดยใช้โปรโตคอล ICP, HTCP, CARP, Cache Digest ในการสื่อสารระหว่าง Cache ได้อีกด้วย

เนื่องมาจาก Squid พัฒนามาบน Digital Unix ที่ทำงานร่วมกับ GNU C compiler ดังนั้นจึงสามารถนำ Squid ไปติดตั้งบนระบบปฏิบัติการที่มีคอมไพเลอร์ของ GNU C ตัวอย่างของระบบปฏิบัติการที่โปรแกรม Squid สามารถติดตั้งใช้งานได้คือ Linux, FreeBSD, NetBSD, BSDI, OSF and Digital Unix, IRIX, SunOS/Solaris, NeXTStep, SCO Unix, AIX, HP-UX, OS/2 นอกจากนี้โปรแกรม Squid รุ่นใหม่ยังสามารถคอมไพล์ได้บน Window NT ด้วย GNU-Win32 package ได้อีกด้วย

การทำงานของโปรแกรม Squid นั้นเกิดจากการทำงานร่วมกันระหว่างส่วนประกอบหลายๆส่วนและจะมีการบันทึกการร้องขอข้อมูลของ client และผลการทำงานที่เกิดขึ้นไว้ใน log file ที่แตกต่างกันตามประเภทข้อมูล เช่น Cache.log เป็น log file ที่เก็บข้อมูลเกี่ยวกับความผิดพลาดที่เกิดขึ้นจากการทำงานของโปรแกรม Store.log บันทึกข้อมูลเกี่ยวกับการเก็บหรือนำข้อมูลออกจาก Cache ส่วน log file ชื่อ Access.log จะบันทึกข้อมูลเกี่ยวกับการร้องขอของ client รวมถึงการแลกเปลี่ยน ICP Message ที่เกิดจากการทำงานของโปรแกรม

การวิเคราะห์ log file ส่วนมากจะใช้ entry ของ access.log ในการวิเคราะห์ ซึ่งปัจจุบัน log file มีรูปแบบอยู่ 2 รูปแบบด้วยกันได้แก่ Common log file และ Native log file สำหรับการเลือกใช้งาน log file รูปแบบใดรูปแบบหนึ่งจะขึ้นอยู่กับการกำหนดค่า emulate httpd log โดยปกติแล้ว log file จะเป็นแบบ Native log file แต่หากมีการกำหนดตัวเลือกโดยถ้าหากเลือกใช้ emulate httpd log จะหมายถึงการเลือกใช้ log file ให้เป็นรูปแบบในลักษณะ Common log file รูปแบบของ log file ในแบบ Common log file และแบบ Native log file จะมีรายละเอียดดังนี้

- รูปแบบ Common log file จะเป็นรูปแบบที่ใช้มากใน HTTP Server ซึ่งจะประกอบด้วย 7 field ดังภาพที่ 2.2

Remote-host	rfc931	authuser	date	method URL"	status	bytes

ภาพที่ 2.2 แสดงรูปแบบของ common log file

ข้อมูลต่างๆของ common log file จะแตกต่างจากรูปแบบของ Native log file แต่จะไม่แสดงรายละเอียดเนื่องจากไม่ได้นำข้อมูลจาก common log file นี้มาใช้ในการวิเคราะห์ในการพัฒนาโปรแกรมเนื่องจากมีรูปแบบและรายละเอียดไม่ตรงตามที่ต้องการ

- รูปแบบ Native log file เป็นรูปแบบที่เหมาะจะนำมาใช้ในการวิเคราะห์เนื่องจากมีลักษณะของข้อมูลที่จัดเก็บมากกว่ารูปแบบของ common log file ลักษณะของ Native log file format แสดงดังภาพที่ 2.3 ดังนี้

time	duration	client address	result codes	bytes	request method	URL	rfc931	hierarchy code	type

ภาพที่ 2.3 แสดงรูปแบบ Native log file

โดยทั่วไป Native log file จะประกอบไปด้วย ข้อมูลอย่างน้อย 10 คอลัมน์แสดงดังภาพที่ 2.3 โดยแต่ละคอลัมน์จะมีรายละเอียดดังตารางที่ 2.1

ตารางที่ 2.1 ตารางแสดงข้อมูล Native log file และความหมาย

ชื่อ field	ความหมาย
time	ใช้บันทึกเวลา timestamp ของ Unix ที่เกิด transaction (มีหน่วยเป็นวินาที)
duration	เวลาที่ transaction ทำงานกับ cache มีหน่วยเป็นวินาที (แต่มีความละเอียดสูงถึงมิลลิวินาที)
client address	IP address ของclient ที่ร้องขอ object
result codes	จะประกอบไปด้วย 2 ส่วนและคั่นด้วยเครื่องหมาย "/" ซึ่งมีรูปแบบคือ Squid result code/status codes ซึ่งแต่ละส่วนจะมีรายละเอียดคือ 1. Squid result code จะบอกชนิดของ request เช่น TCP_MISS (หมายถึง ร้องขอ object แต่ไม่มีใน cache) เป็นต้น 2. Status codes จะเก็บ HTTP result code เช่น 404 จะหมายถึงหาไม่พบ เป็นต้น
Bytes	บอกถึงขนาดของข้อมูลทั้งหมดที่ส่งไปยัง client
request method	วิธีของ client ที่ใช้ในการร้องขอ object เช่น GET, POST, PUT , DELETE เป็นต้น
URL	บอกถึง URL ที่ client ร้องขอ
rfc931	บันทึก ident lookup สำหรับการร้องขอของclient ซึ่งจะมีการกำหนดค่าเริ่มต้นเป็น ident_lookup off หมายถึง จะไม่มีค่าในคอลัมน์นี้ซึ่งจะมีเครื่องหมาย "-" บันทึกอยู่แทน
hierarchy code	ประกอบไปด้วย 3 รายการดังนี้ 1. hierarchy tag อาจจะประกอบด้วยคำนำหน้าที่ว่า TIMEOUT_ ถ้าหากเกิดเวลาการรอการตอบกลับของทุกๆ ICP จากเพื่อนบ้านหมด 2. เป็นรหัสที่บอกถึงการร้องขอว่าถูกควบคุมการร้องขออย่างไร เช่น DIRECT เป็น object ที่รับมาโดยตรงจาก server 3. IP address หรือ hostname ของ server ที่มีการส่งการร้องขอไป
Type	บันทึกชนิดของ object ซึ่งจะเห็นใน header ของ HTTP reply ถ้าเป็นการแลกเปลี่ยน ICP จะไม่มีคอลัมน์นี้แต่จะใส่เป็นเครื่องหมาย "-" แทน

เพื่อทำให้เกิดความเข้าใจจึงแสดงตัวอย่างของ access.log เพียงสองแถวดังภาพที่ 2.4 โดยภาพที่ 2.4a แสดงตัวอย่างของ access.log จำนวน 1 แถวเป็นข้อมูลบอกรายละเอียด

การร้องขอออบเจ็กต์ประเภทรูปภาพ หมายเลข IP ADDRESS ที่ client ที่ร้องขอคือ 192.168.0.5 โดยร้องขอออบเจ็กต์ที่ไม่มีเก็บอยู่ใน cache ได้สำเร็จ และจำนวนการร้องขอ คือ 7257 byte ใช้วิธีการแบบ GET โดย client มีการร้องขอ URL http://www.cscoms.com/football/images/football_03.jpg และได้รับ object โดยตรงจาก web server ที่มี IP ADDRESS เป็น 202.183.255.25 เป็นต้น ส่วนตัวอย่างที่ 2 แสดงดังภาพที่ 2.4 b จะเป็นการร้องขอข้อมูลที่เกิดจากเครื่อง client ที่มีหนอนอินเตอร์เน็ตแบบ CodeRed แอบแฝงทำงานอยู่ ซึ่งจะทำให้เกิดการโจมตีแบบ DoS attack โดยเครื่อง client ดังกล่าวจะร้องขอออบเจ็กต์ด้วยรูปแบบนี้ซ้ำๆ กันมากเกินปกติเป็นสาเหตุให้ พื้นที่เก็บข้อมูลเต็มจนทำให้โปรแกรม Web Proxy Cache หยุดทำงานได้ในที่สุด

```
1022908004.002 1886 192.168.0.5 TCP_MISS/200 7257 GET
http://www.cscoms.com/football/images/football_03.jpg - DIRECT/202.183.255.25
image/jpeg
```

ภาพที่ 2.4a แสดงตัวอย่างของ access.log

```
1029085144.350 12 161.246.15.181 NONE/411 1620 GET
http://www.worm.com/default.ida? – NONE/- -
```

ภาพที่ 2.4b ตัวอย่างของ access.log ซึ่งถูก CodeRed โจมตี

ภาพที่ 2.4 แสดงตัวอย่างของ access.log

2.3 ไฟร์วอลล์ (Firewall)

ไฟร์วอลล์ (Firewall) มีหน้าที่กรองแพ็กเกต โดยสามารถกำหนดให้แพ็กเกตใดผ่านหรือไม่ผ่านเข้า-ออกจากเครือข่ายภายในได้ ดังนั้นการใช้ไฟร์วอลล์จะเป็นการสร้างความปลอดภัยให้กับเครือข่ายที่ต้องติดต่อไปยังเครือข่ายภายนอกได้

2.3.1 ประเภทของไฟร์วอลล์

ไฟร์วอลล์สามารถแบ่งออกเป็น 2 ประเภทคือ

- การกรองแพ็กเกต (Packet Filtering) เป็นการอนุญาตหรือปฏิเสธแพ็กเกตต่างๆที่จะเข้ามาในระบบหรือออกจากระบบ โดยผู้ดูแลระบบสามารถตั้งกฎต่างๆกำหนดชนิดของ แพ็กเกตเพื่ออนุญาตหรือปฏิเสธการเข้า-ออกของแพ็กเกตนั้นได้ การกรองแพ็กเกตอาจจะพบในเราท์เตอร์บริดจ์ หรือบนโฮสท์ ซึ่งอาจจะเรียกวิธีการนี้ว่า “Screening”
- พร็อกซีบริการ (Proxy Service) เป็นแอพพลิเคชันทำขึ้นมาเฉพาะหรือเป็นโปรแกรมเซิร์ฟเวอร์ (Server Program) ที่ทำงานบนโฮสท์ที่เป็นไฟร์วอลล์ (Firewall Host) ซึ่งตำแหน่งของพร็อกซีบริการ(Proxy Service) นั้นจะตั้งอยู่ระหว่างผู้ใช้ภายในกับอินเตอร์เน็ตภายนอกและจะทำหน้าที่ในการติดต่อขอใช้บริการต่างๆจากภายนอกแทนผู้ใช้บริการ ดังนั้นพร็อกซีบริการ (Proxy Service) จะทำหน้าที่ดูแลการติดต่อสื่อสารทั้งหมดระหว่างผู้ใช้งานจริงกับภายนอกเช่น การใช้งาน FTP หรือการใช้งาน Telnet เป็นต้น

2.3.2 ไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี

ระบบปฏิบัติการฟรีบีเอสดีเป็นระบบปฏิบัติการยูนิกส์ที่พัฒนาต่อมาจาก BSD ของ Berkeley ซึ่งระบบปฏิบัติการฟรีบีเอสดีได้รับการพัฒนาจากกลุ่มคนที่ทำงานวิจัยของคณะวิทยาศาสตร์คอมพิวเตอร์ของมหาวิทยาลัยเบอร์กเลย์ที่แคลิฟอเนีย(Univversity of California, Berkeley UCB)

ระบบปฏิบัติการฟรีบีเอสดีมีบริการของไฟร์วอลล์ซ่อนอยู่และถูกนำไปรวมอยู่ในเคอร์เนลทำให้สามารถดูแลและควบคุมพฤติกรรมโดยรวมและเปลี่ยนแปลงนโยบายทางด้านความปลอดภัยของการติดต่อสื่อสารที่ระดับไอพีได้ ซึ่งไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีที่กล่าวถึงนี้ คือ ไอพีไฟร์วอลล์ (IP Firewall) เป็น ไฟร์วอลล์ประเภทการกรองแพ็กเกต (Packet Fillering) การใช้งานจะต้องมีการคอมไพล์เคอร์เนล ซึ่งต้องทำการแก้ไขคอนฟิกส์ไฟล์ของเคอร์เนลโดยการเพิ่มตัวเลือก (Option) เพื่อให้สามารถใช้งานไอพีไฟร์วอลล์ (IP Firewall) ได้

2.3.3 ไอพีไฟร์วอลล์ (IP Firewall)

ไอพีไฟร์วอลล์ (IP Firewall) เป็นไฟร์วอลล์ (Firewall) ที่ควบคุมการเข้าถึงในระดับไอพี (IP Layer) และระดับทรานสปอร์ต (Transport Layer) ในทีซีพี/ไอพี ซึ่งสามารถกำหนดให้มีการ

อนุญาต(Allow) หรือการปฏิเสธ (Deny) แพ็กเกตได้ มีการกำหนดแอดเดรสปลายทางและต้นทาง และสามารถกำหนดลักษณะอื่นๆได้ดังจะกล่าวถึงต่อไป

2.3.3.1 การกำหนดกฎของไอพีไฟร์วอลล์ (IP Firewall)

จากที่ได้กล่าวมาแล้วว่าไอพีไฟร์วอลล์ (IP Firewall) เป็นการทำงานแบบการกรองแพ็กเกตซึ่งการกรองแพ็กเกตจะมีการตรวจสอบแพ็กเกตที่เข้ามากับรายการของกฎซึ่งรายการของกฎนั้นจะเรียงลำดับจากลำดับที่ 1 ถึง 65534 ซึ่งลำดับนี้จะเป็นลำดับการตรวจสอบแพ็กเกตที่เข้ามาด้วย แต่สำหรับกฎลำดับที่ 65535 จะเป็นกฎ Default rule ไม่สามารถแก้ไขเปลี่ยนแปลงได้โดยที่กฎลำดับที่ 65535 นี้มักเป็นการกำหนดให้ปฏิเสธแพ็กเกตที่เข้ามาทั้งหมด ส่วนลำดับอื่นๆสามารถกำหนดเป็นกฎได้ กฎทั้งหมดจะมีความสัมพันธ์กับตัวนับ (Counter) มี 2 ตัวด้วยกันคือ ตัวนับแพ็กเกต และ ตัวนับไบต์ โดยที่ตัวนับเหล่านี้จะถูกเปลี่ยนแปลงเมื่อมีแพ็กเกตที่เข้ามาสามารถจับคู่กับกฎ การตรวจสอบแพ็กเกตจะทำโดยการเทียบระหว่างแพ็กเกตกับกฎจนกระทั่งเข้าคู่กัน การกำหนดกฎต่างๆของไอพีไฟร์วอลล์ (IP Firewall) มีพื้นฐานดังต่อไปนี้

- Addition เป็นการเพิ่มกฎ
- Delete เป็นการลบกฎที่มีอยู่ในรายการกฎออก
- Flush เป็นการลบกฎทั้งหมดออก
- Show/List แสดงรายละเอียดทั้งหมดของกฎ
- Zero/Resetlog เป็นการปรับค่าตัวนับต่างๆ

รูปแบบทั่วไปของการใช้คำสั่งของไอพีไฟร์วอลล์(ipfw) แสดงดังภาพที่ 2.5

```
ipfw [prob match_probability] action [log[logamount number] proto from ser to dst [interface_spec][options]
```

ภาพที่ 2.5 แสดงรูปแบบทั่วไปของคำสั่ง ipfw

ภาพที่ 2.5 เป็นภาพแสดงรูปแบบทั่วไปของการกำหนดกฎในไอพีไฟร์วอลล์ โดยจะแสดงเป็นตัวแปรต่างๆ ที่มีความหมายแตกต่างกัน ความหมายของตัวแปรแต่ละตัวแสดงในตารางที่ 2.2

ตารางที่ 2.2 แสดงความหมายของตัวแปรแต่ละตัวในรูปแบบทั่วไปของการใช้คำสั่ง ipfw

ตัวแปร	ความหมาย
prob	ความน่าจะเป็นที่จะสุ่มแพ็กเกตที่เข้าคู่กับกฎเพื่อให้เป็นไปตาม action ที่กำหนด
action	กิจกรรมที่กำหนดให้กระทำกับแพ็กเกตนั้นๆเช่น อนุญาต หรือ ปฏิเสธ
log	ให้แสดงค่าที่อยู่ใน log ที่กำหนด(ปกติทางหน้าจอ)
proto	โปรโตคอลที่ต้องการจะทำการกรอง
ser	ไอพีแอดเดรสต้นทาง
dst	ไอพีแอดเดรสปลายทาง
interface_spec	เป็นการระบุช่องทางและทิศทางการเข้ามาหรือออกไปของแพ็กเกต
options	ตัวเลือกที่สามารถเพิ่มได้

จากรูปแบบทั่วไปของการใช้คำสั่ง ipfw ที่แสดงความหมายไปแล้วนั้นสามารถแสดงรายละเอียดของการกำหนดกฎต่างๆในไอพีไฟร์วอลล์ซึ่งมีรูปแบบคำสั่งที่แตกต่างกันดังนี้

- รูปแบบคำสั่งของ Addition/Deletion แสดงดังภาพที่ 2.6 และความหมายของตัวแปรในตารางที่2.3

```
ipfw [-N] command [index] action [log] protocol address [options]
```

ภาพที่ 2.6 แสดงรูปแบบคำสั่งของการ Addition/Deletion

ภาพที่ 2.6 แสดงรูปแบบคำสั่งการ เพิ่มหรือ ลบกฎในไอพีไฟร์วอลล์โดยแสดงความหมายของตัวแปรต่างๆในตารางที่ 2.3

ตารางที่ 2.3 แสดงตัวแปรและความหมายของคำสั่ง Addition/Deletion

ตัวแปร	ตัวเลือก	ความหมาย
[-N]	-	ใช้แยกส่วนของแอดเดรสและชื่อบริการต่างๆใน Output
command	add	ใช้ในการเพิ่มกฎ
	delete	ใช้ลบกฎ
[index]	-	เป็นการระบุตำแหน่งของกฎ
action	allow	เป็นการอนุญาตให้แพ็กเกตผ่านได้
	deny	เป็นการปฏิเสธไม่ให้แพ็กเกตผ่าน
	reject	เป็นการลบแพ็กเกตที่เข้าคู่กับกฎและจะส่งข้อความไปที่โฮสต์ไอซีเอ็มพีที่ไม่สามารถขยายต่อไปได้(Unreachable)
	unreach code	เป็นการลบแพ็กเกตที่เข้าคู่กับกฎและจะส่งข้อความไปที่โฮสต์ไอซีเอ็มพีที่ไม่สามารถขยายต่อไปได้ด้วยโค้ดซึ่งมีค่าตั้งแต่ 0 – 255
	reset	ใช้กับแพ็กเกตที่เป็นทีซีพีเท่านั้น โดยจะลบแพ็กเกตที่เข้าคู่กับกฎและจะส่งข้อความที่เป็นทีซีพีรีเซต(TCP reset Message)
	count	เป็นการนับตัวนับแพ็กเกต
	divert port	ทำการ Divert แพ็กเกตที่เข้าคู่กับกฎไปที่ยุ่งกับพอร์ตที่ระบุ
	tee port	คัดลอกแพ็กเกตที่เข้าคู่กับกฎไปยังพอร์ตที่ระบุ
[log]	-	ทำให้เกิดการเข้าคู่กฎเพื่อไปแสดงผลที่ System console ถ้าเคอร์เนลได้รับการคอมไพล์ด้วย options IPFIREWALL_VERBOSE
protocol	all	เป็นการกำหนดให้จับคู่กับทุกไอพีของแพ็กเกต
	icmp	เป็นการจับคู่กับแพ็กเกตที่มีโปรโตคอลเป็น ไอซีเอ็มพี
	tcp	เป็นการจับคู่กับแพ็กเกตที่มีโปรโตคอลเป็น ทีซีพี
	udp	เป็นการจับคู่กับแพ็กเกตที่มีโปรโตคอลเป็น ยูดีพี
address	-	มีการระบุแอดเดรสได้ดังภาพที่ 2.7 และความหมายในตารางที่ 2.4

ตารางที่ 2.3 แสดงตัวแปรและความหมายของคำสั่ง Addition/Deletion (ต่อ)

ตัวแปร	ตัวเลือก	ความหมาย
option	frag	จับคู่ถ้าแพ็กเกตไม่ได้อยู่ใน fragment แรกของดาต้าแกรม
	in	จับคู่ถ้าแพ็กเกตนั้นเป็นแพ็กเกตที่เข้ามา
	out	จะจับคู่ถ้าแพ็กเกตนั้นเป็นแพ็กเกตที่ออกไป
	ipoption spec	จะจับคู่ถ้าส่วนหัวของไอพี มี comma separated list ของตัวเลือกที่ถูกระบุใน spec เช่น ssr(strict source route)
	established	จะจับคู่ถ้าแพ็กเกตนั้นเป็นส่วนของการจัดสร้างการเชื่อมต่อของทีซีพี
	setup	จะจับคู่ถ้าแพ็กเกตนั้นเป็นแพ็กเกตนั้นเป็นแพ็กเกตที่พยายามจะจัด สร้างการเชื่อมต่อของทีซีพี
	tcpflags flags	จะจับคู่ถ้าส่วนหัวของทีซีพีมี comma separate list ของ flags เช่น fin syn rst psh ack และ urg
	icmptypes tytps	จะจับคู่ถ้าชนิดของไอซีเอ็มพีปรากฏอยู่ในรายการของ type

จากรูปแบบคำสั่งของการ Addition/Deletion และความหมายของตัวแปรและตัวเลือกต่างๆที่แสดงดังตารางที่ 2.3 นั้นสามารถระบุ address ได้หลายลักษณะดังแสดงความหมายเพิ่มเติมในรูปแบบการระบุแอดเดรสในภาพที่ 2.7 และตารางที่ 2.4 แสดงความหมายและตัวเลือกของการระบุแอดเดรส

```
from address/mark[port] to address/mark [port] [via interface]
```

ภาพที่ 2.7 แสดงรูปแบบการระบุแอดเดรส

ตารางที่ 2.4 แสดงความหมายและตัวเลือกของการระบุแอดเดรส

ตัวแปร	ตัวเลือก	ความหมาย
address/mark	Address	ไอพีแอดเดรส
	address/mark-bits	ไอพีแอดเดรสที่มีสับเน็ตมาร์ก(Sub-netmark)แบบย่อ
	address/mark-pattern	ไอพีแอดเดรสที่มีสับเน็ตมาร์ก(Sub-netmark)แบบย่อ
[port]	port[,port[,port[…]]]	เป็นการระบุถึงพอร์ตเพียงพอร์ตเดียวหรือระบุเป็นรายการของพอร์ต
	port[,port[,port[…]]]	port-port เป็นการระบุช่วงของพอร์ต
'via'	-	เป็นสิ่งที่เลือกได้และอาจจะมีการระบุไอพีแอดเดรสหรือชื่อโดเมนของอินเตอร์เฟสของไปพีอินเตอร์เฟสภายในหรือชื่ออินเตอร์เฟส เช่น ed0 เพื่อจับคู่กับแพ็กเกตที่ผ่านเข้ามาในอินเตอร์เฟสนี้

- รูปแบบคำสั่งของ Listing แสดงดังในภาพที่ 2.8

```
Ipfw [-a][-t][-N] list
```

ภาพที่ 2.8 แสดงรูปแบบคำสั่งของการ listing

จากภาพที่ 2.8 เป็นการแสดงรูปแบบการใช้คำสั่งเพื่อให้แสดงรายการและรายละเอียดของกฎต่างๆที่ได้ระบุไว้แล้ว โดยจะมีพารามิเตอร์ทั้งหมด 3 ตัวซึ่งแต่ละตัวมีความหมายแสดงในตารางที่ 2.5

ตารางที่ 2.5 แสดงความหมายของพารามิเตอร์คำสั่งการ Listing

พารามิเตอร์	ความหมาย
-a	ขณะที่ทำการแสดงรายการของกฎจะแสดงค่าของตัวนับ
-t	แสดงการจับคู่กฎกับ packetในครั้งสุดท้ายในแต่ละกฎ
-N	การพยายามแยกส่วนของแอดเดรสและชื่อบริการต่างๆ

- รูปแบบคำสั่งของ Flushing ดังแสดงในภาพที่ 2.9

ipfw flush

ภาพที่ 2.9 แสดงรูปแบบคำสั่งของการ Flushing

สิ่งที่ต้องระวังเมื่อมีการใช้คำสั่งนี้คือ ค่าเริ่มต้นของนโยบายจะเป็นการปฏิเสธการบริการทุกอย่างในเครือข่ายจนกว่าจะมีการเพิ่มกฎเข้าไป

- รูปแบบคำสั่งของ Clearing ดังแสดงในภาพที่ 2.10

Ipfw zero[index]

ภาพที่ 2.10 แสดงรูปแบบคำสั่งของการ Clearing

เมื่อการใช้คำสั่งนี้ไม่มีส่วนของ index จะทำให้ตัวนับแพ็กเกตทั้งหมดถูกลบค่าออกไปด้วยแต่ถ้ามีการระบุค่าใน index การลบค่าตัวนับแพ็กเกตจะกระทำเฉพาะกฎนั้นๆ

2.3.3.2 ตัวอย่างกฎของไอพีไฟร์วอลล์ (IP Firewall)

จากรูปแบบทั่วไปของคำสั่งต่างๆที่ได้กล่าวมาแล้วต่อไปนี้จะแสดงตัวอย่างการใช้คำสั่งของไอพีไฟร์วอลล์ในฟรีบีเอสดี

- การปฏิเสธแพ็กเกตทั้งหมดจากโฮสต์ที่ชื่อว่า evil.crackers.org เพื่อเทลเน็ตมาที่พอร์ตของโฮสต์ nice.people.org

  #ipfw add deny tcp form evil.crackers.org to nice.people.org 23

- การปฏิเสธและการบล็อกทุกแพ็กเกตของทีซีพีจากเครือข่าย cracker.org (Class C) เพื่อที่จะเข้ามายังเครื่อง nice.people.org ในทุกพอร์ต

  # ipfw add denymlog tcp form evil.crackers.org/24 to nice.people.org

- การดูเรคคอร์ตของรายการของกฎ

#ipfw -a list หรือ

#ipfw –al

- การดูกฎที่เข้าคู่กับแพ็กเกตได้ในครั้งสุดท้าย

ipfw –at l

สรุป

เว็บพร็อกซี/แคชมีหลากหลาย เช่นโปรแกรม Squid ทำให้องค์กรใช้ทรัพยากรทางด้านเครือข่ายเป็นไปอย่างมีประโยชน์สูงสุด ซึ่งจะช่วยแบ่งเบาภาระงานโดยการลดจำนวนการร้องขอข้อมูลภายในเครือข่ายได้ ดังนั้นการนำเอาโปรแกรมเว็บ พร็อกซี/แคชมาใช้ในองค์กรมีประโยชน์มาก ดังนั้นการรักษาความปลอดภัยให้กับโปรแกรมเว็บ พร็อกซี/แคชก็เป็นสิ่งจำเป็นเช่นกัน

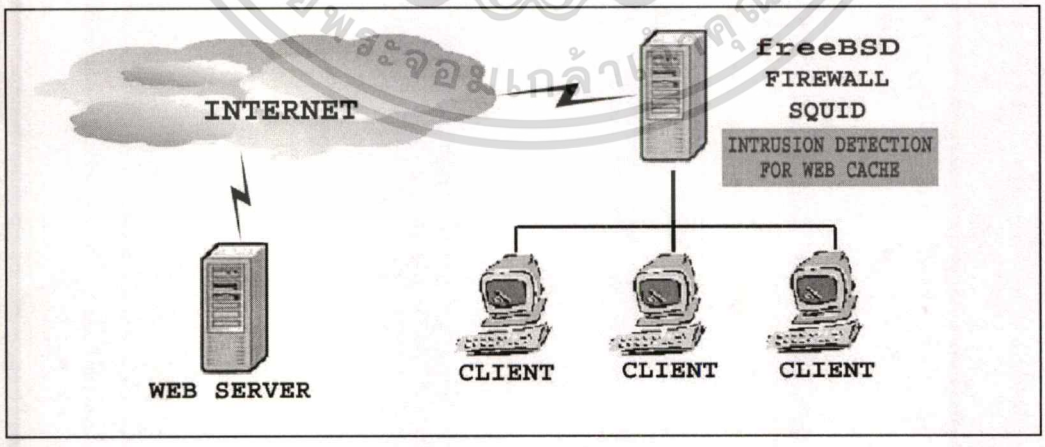
การทำงานของระบบตรวจจับการบุกรุก และไฟร์วอลล์ จะเป็นการรักษาความปลอดภัยให้กับระบบซึ่งจะมีหน้าที่และการทำงานที่แตกต่างกันโดยระบบตรวจจับการบุกรุก จะใช้สำหรับการ monitor เหตุการณ์ต่างๆที่เกิดขึ้นภายในระบบแต่ไฟร์วอลล์สามารถจะอนุญาตหรือไม่อนุญาตให้มีการติดต่อสื่อสารกันได้ การใช้งานระบบรักษาความปลอดภัยเพียงอย่างหนึ่งอย่างใดเพื่อให้ระบบมีความปลอดภัยเป็นไปได้แต่อาจจะไม่ดีที่สุด ดังนั้นหากต้องการให้ระบบมีความปลอดภัยให้มากขึ้นความใช้วิธีการรักษาความปลอดภัยหลากหลายหน้าที่

บทที่ 3
การออกแบบโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

เนื้อหาในบทที่ 2 ได้อธิบายถึงหลักการและความรู้ที่จำเป็นสำหรับการนำไปใช้ในการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช ส่วนในบทที่ 3 นี้จะกล่าวถึงการออกแบบโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชโดยจะอธิบายถึงภาพรวมของระบบ และสถานะของการร้องขอข้อมูลที่เกิดขึ้นภายในระบบ เพื่อนำมาวิเคราะห์และออกแบบการทำงานของโปรแกรม หลังจากทราบสถานะของการร้องขอแล้วจะอธิบายถึงรายละเอียดในส่วนของไฟล์ทั้งหมดที่ออกแบบเพื่อใช้ในการทำงานของโปรแกรม จากนั้นจะอธิบายรายละเอียดของโปรแกรม การไหลของข้อมูลภายในระบบ และสุดท้ายจะเป็น Flow Chart ของโปรแกรมซึ่งเนื้อหาต่างๆจะกล่าวเรียงตามลำดับเป็นหัวข้อดังหัวข้อต่างๆ ต่อไปนี้

3.1 ภาพรวมการทำงานของโปรแกรม

เพื่อความเข้าใจการทำงานของโปรแกรมจะแสดงภาพรวมของระบบซึ่งทำให้เห็นตำแหน่งของการติดตั้งโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชและสภาพแวดล้อมการทำงานของโปรแกรมดังแสดงในภาพที่ 3.1

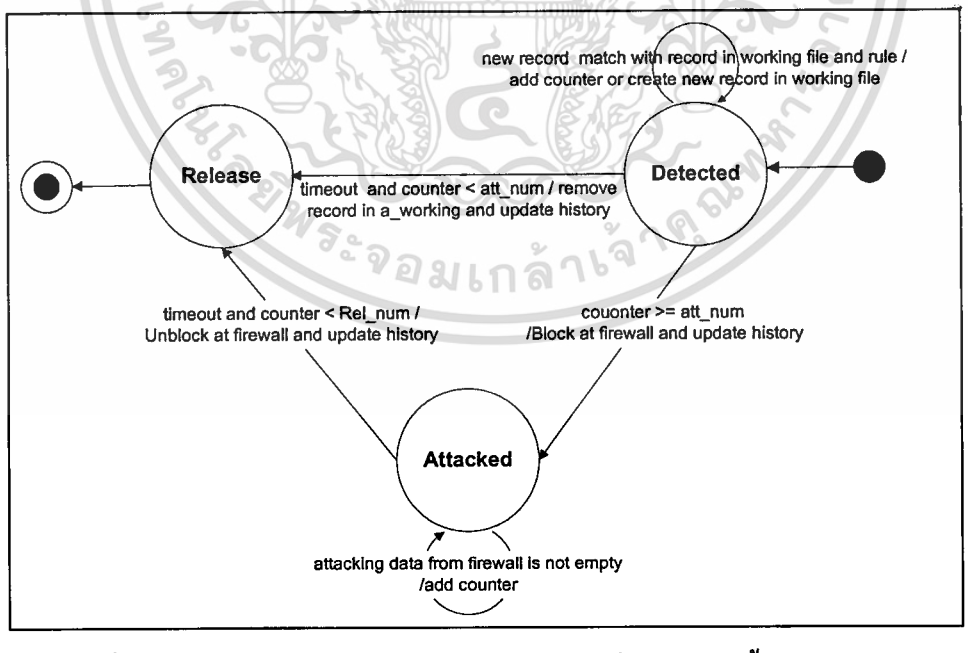


ภาพที่ 3.1 แสดงตำแหน่งที่ตั้งของโปรแกรม

โปรแกรมตรวจจับการโจมตีทำงานบนเครื่องคอมพิวเตอร์ที่มีการติดตั้งระบบปฏิบัติการฟรีบีเอสดี(FreeBSD) และมีโปรแกรมที่เกี่ยวข้อง 2 โปรแกรมด้วยกันคือ โปรแกรม Squid และไอพีไฟร์วอลล์ (IP Firewall) ของระบบปฏิบัติการฟรีบีเอสดี(FreeBSD) เมื่อมีการร้องขอข้อมูลออบเจ็กต์จาก client โปรแกรม Squid จะบันทึกการร้องขอออบเจ็กต์ต่างๆเก็บไว้ซึ่งทำให้โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชสามารถดึงข้อมูลในส่วนการร้องขอนี้มาวิเคราะห์เพื่อหาความผิดปกติและหากพบการร้องขอที่มากเกินปกติจากเครื่อง client ตัวใดตัวหนึ่งจะส่งคำสั่งตอบสนองไปยังไฟร์วอลล์ให้ปฏิเสธการร้องขอนั้น หลังจากนั้นโปรแกรมจะตรวจสอบการโจมตีเพื่อยกเลิกการปฏิเสธการร้องขอโดยโปรแกรมจะพิจารณาการโจมตีที่ไฟร์วอลล์หากไม่พบการโจมตีจะส่งคำสั่งไปให้ไฟร์วอลล์ยกเลิกการปฏิเสธการร้องขอเพื่อให้เครื่อง client ร้องขอข้อมูลได้ตามปกติ ซึ่งรายละเอียดของการทำงานจะกล่าวถึงในส่วนต่อไป

3.2 สถานะของการตรวจจับการถูกโจมตี

ก่อนออกแบบโปรแกรมควรวิเคราะห์สถานะของการตรวจจับการถูกโจมตีที่เกิดขึ้นภายในระบบเนื่องจากจะทำให้ทราบถึงสถานะต่างๆที่เกิดขึ้นในระบบ สถานะทั้งหมดที่เป็นไปได้มี 3 สถานะซึ่งจะแสดงดังภาพที่ 3.2



ภาพที่ 3.2 แสดงสถานะการตรวจจับการถูกโจมตีที่เป็นไปได้ทั้งหมดในระบบ

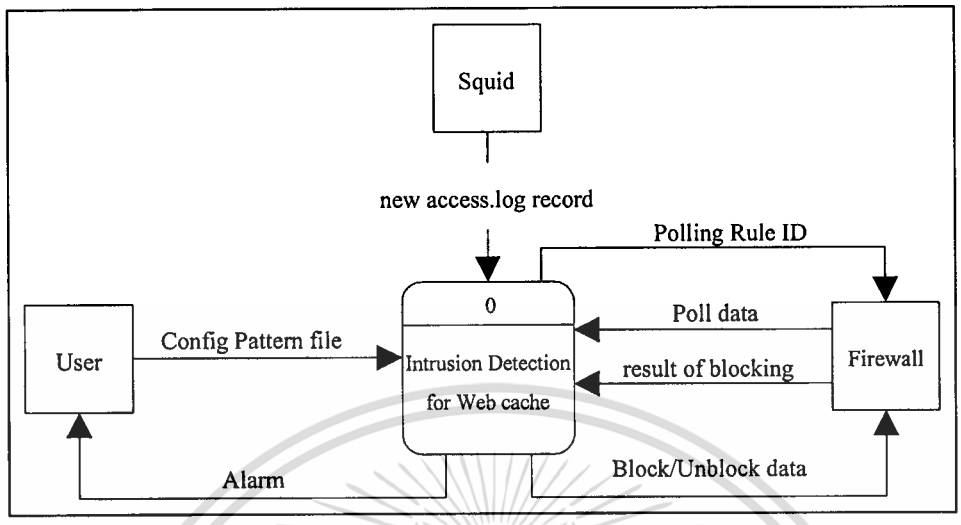
จากภาพสามารถอธิบายการเปลี่ยนสถานะของการร้องขอออบเจ็กต์ได้ดังนี้

1) สถานะ Detected เมื่อมีการร้องขอออบเจ็กต์เกิดขึ้นจาก Client หนึ่งๆจะถูกเปลี่ยนสถานะเป็นสถานะ Detected เมื่อตรวจสอบพบว่าเข้าคู่กับรูปแบบที่กำหนดไว้
2) สถานะ Attacked การร้องขอออบเจ็กต์จะเปลี่ยนมาเป็นสถานะ Attacked ได้จะมีเงื่อนไขดังนี้
   - การร้องขอออบเจ็กต์จะต้องมีสถานะเป็น Detected มาก่อนเท่านั้น
   - เมื่อมีจำนวนของการร้องขอออบเจ็กต์ซ้ำๆกันมากกว่าหรือเท่ากับจำนวนที่กำหนดไว้
3) สถานะ Release การร้องขอออบเจ็กต์จะเปลี่ยนเป็นสถานะ Release ได้จะต้องสอดคล้องกันกรณีใดกรณีหนึ่งในสองกรณีต่อไปนี้
   - การร้องขอออบเจ็กต์ที่มีสถานะเป็น Detected และตรวจสอบพบจำนวนการร้องขอออบเจ็กต์น้อยกว่าที่กำหนด โดยระยะเวลาในการร้องขอออบเจ็กต์มากกว่าเวลาที่กำหนดจะถือว่าการร้องขอออบเจ็กต์นั้นเปลี่ยนสถานะจาก Detected เป็นสถานะ Release
   - การร้องขอออบเจ็กต์มีสถานะเป็น Attacked และพบว่าเมื่อใดที่เวลาในการตรวจสอบการร้องขอออบเจ็กต์มากกว่าหรือเท่ากับเวลาที่ได้กำหนดแต่ มีจำนวนของการร้องขอออบเจ็กต์ซ้ำๆกันนั้นน้อยกว่าจำนวนที่ได้กำหนด หากมีเงื่อนไขครบตามที่ได้กล่าวมาการร้องขอนั้นจะเปลี่ยนสถานะจาก Attacked มาเป็น Release

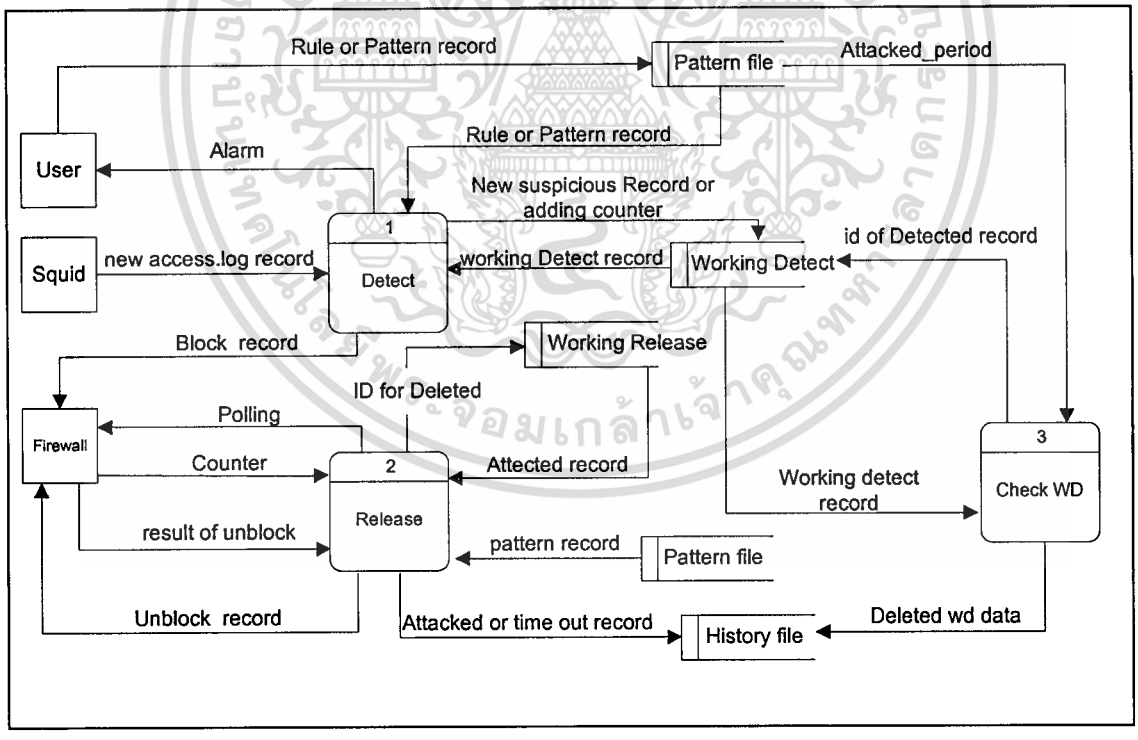
เมื่อทราบสถานะการร้องขอออบเจ็กต์ของระบบแล้วสามารถวิเคราะห์และออกแบบโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชซึ่งจะอธิบายในหัวข้อถัดไป

3.3 ผังการไหลของข้อมูลของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

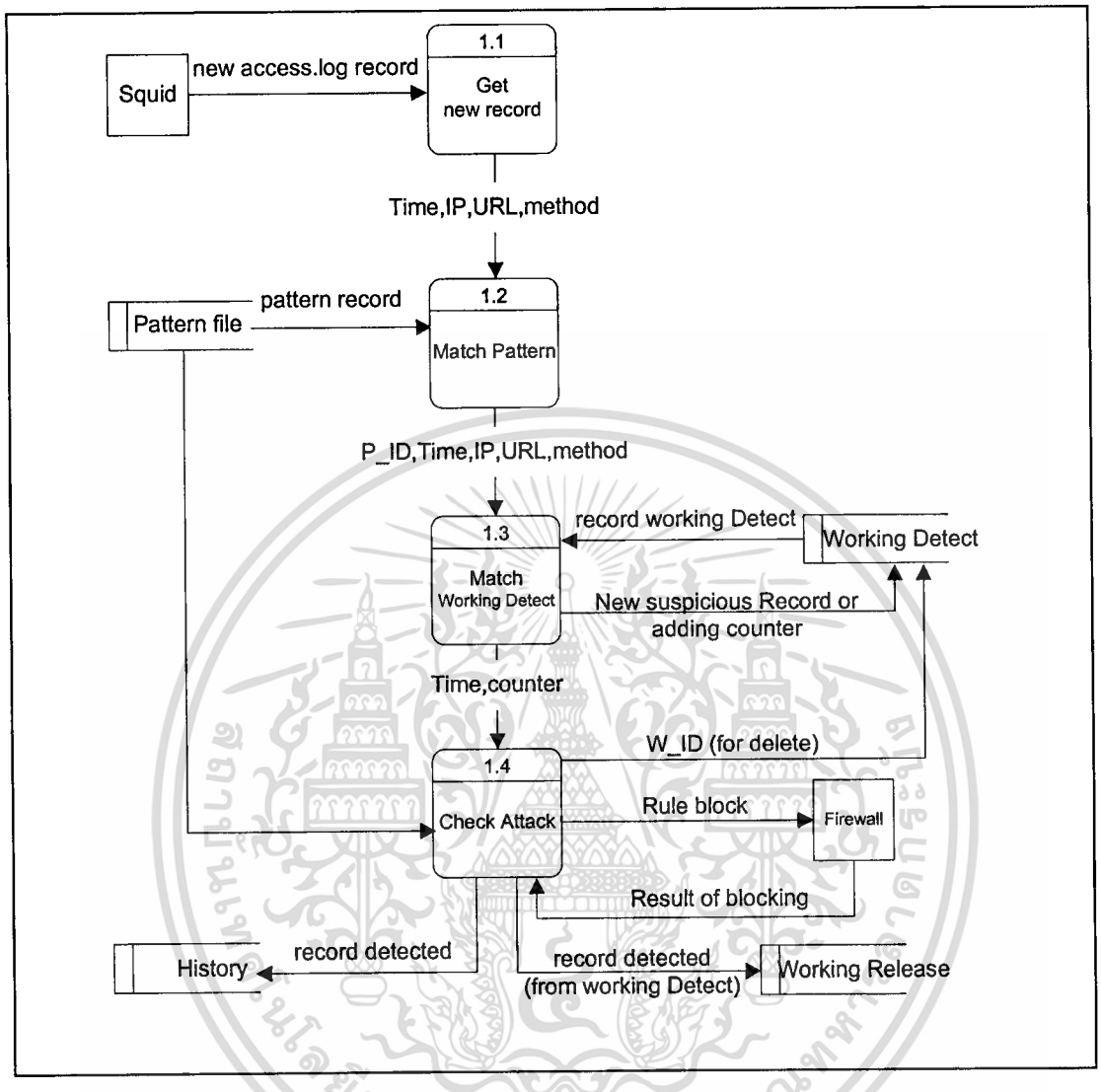
จากภาพรวมของการทำงานและสถานะของการตรวจจับการถูกโจมตีข้างต้นสามารถเขียนเป็นแผนภาพแบบ Context Diagram ซึ่งแสดงภาพรวมของระบบและการติดต่อกับสิ่งแวดล้อมภายนอกอันได้แก่ Squid, ไฟร์วอลล์ และผู้ใช้งานดังภาพที่ 3.3 และเพื่อให้เห็นการไหลของข้อมูลให้ละเอียดมากขึ้นจึงแสดงเป็นกระบวนการย่อยซึ่งเป็นแผนภาพการไหลของข้อมูลระดับที่ 0 ดังภาพที่ 3.4 ซึ่งจะเห็นได้ว่ามีหน้าที่การทำงานแบ่งย่อยได้เป็น 3 หน้าที่ได้แก่ Detected, Release และการตรวจสอบ time out ในworking detect ในชื่อ Check WD ในส่วนของ ภาพที่ 3.5 ภาพที่ 3.6 และภาพที่ 3.7 จะแสดงผังการไหลข้อมูลของระบบตรวจจับการโจมตีในระดับที่ 1 กระบวนการที่ 1 กระบวนการที่ 2 และกระบวนการที่ 3 ตามลำดับดังภาพต่อไปนี้



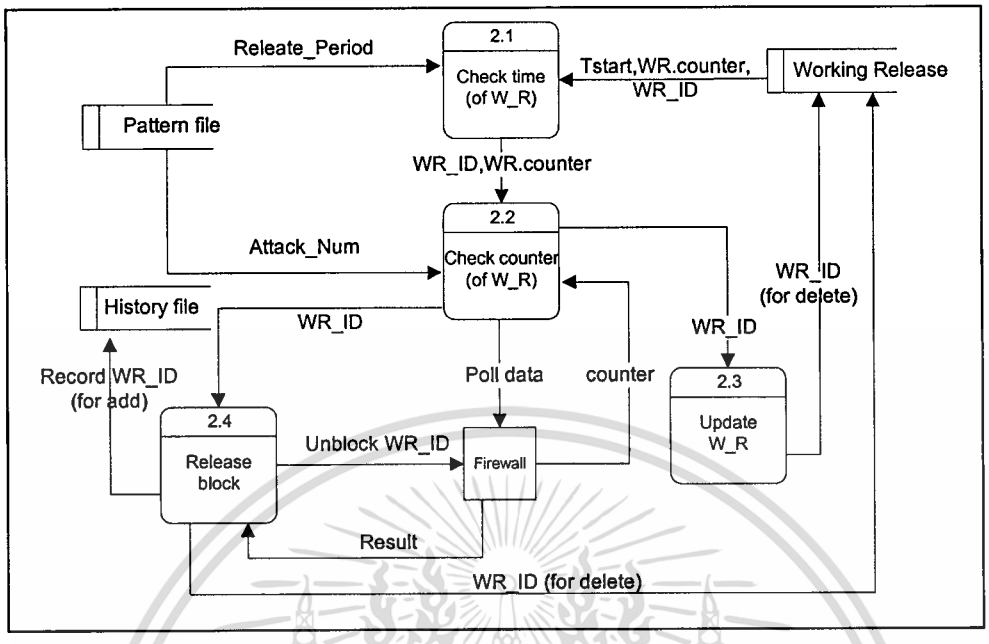
ภาพที่ 3.3 แสดง Context Diagram ของระบบตรวจจับการโจมตีสำหรับเว็บแคช



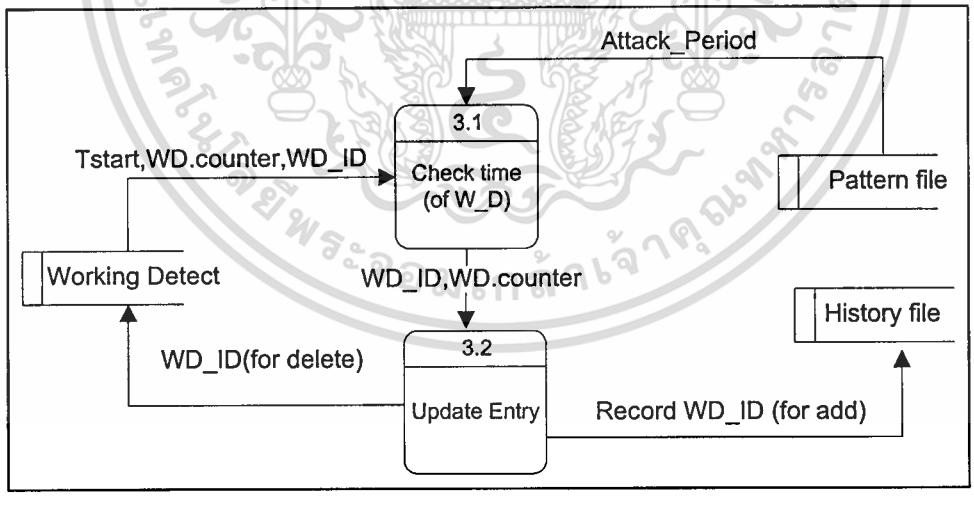
ภาพที่ 3.4 แสดงผังการไหลของข้อมูลของระบบตรวจจับการโจมตีในระดับที่ 0



ภาพที่ 3.5 แสดงผังการไหลข้อมูลของระบบตรวจจับการโจมตีในระดับที่ 1 กระบวนการที่ 1



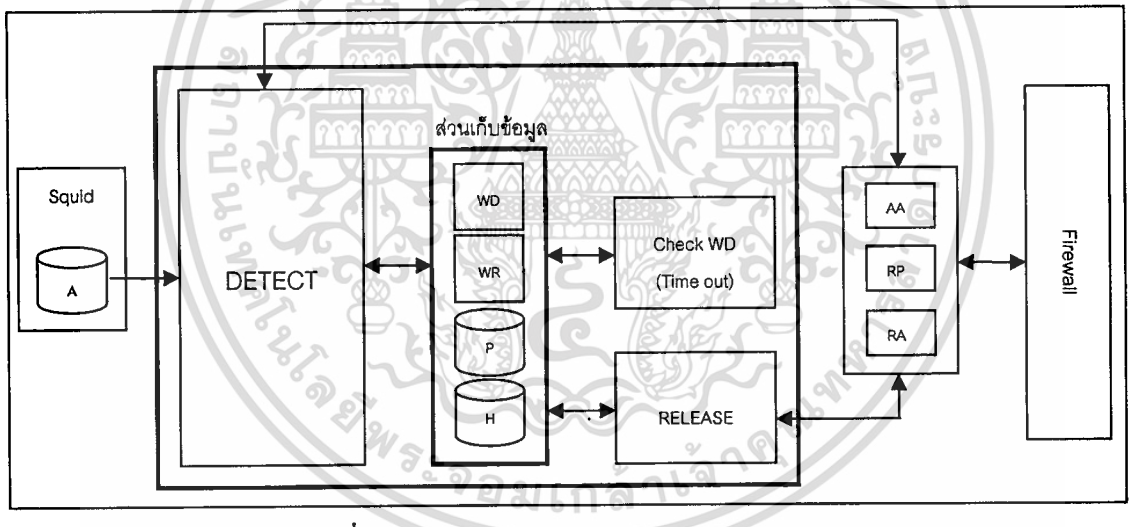
ภาพที่ 3.6 แสดงผังการไหลของข้อมูลของระบบตรวจจับการโจมตีในระดับที่ 1 กระบวนการที่ 2



ภาพที่ 3.7 แสดงผังการไหลของข้อมูลของระบบตรวจจับการโจมตีในระดับที่ 1 กระบวนการที่ 3

3.4 การทำงานของโปรแกรม

ภาพที่ 3.8 เป็นภาพการทำงานทั้งหมดของโปรแกรมที่ประสานงานร่วมกันระหว่างโปรแกรม Squid และ โปรแกรม IP Firewall โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชที่ได้ออกแบบมานั้น ประกอบไปด้วยโปรแกรมย่อยสามส่วนคือ ส่วนแรกเป็นการเฝ้าตรวจจับการโจมตี (Detect) ส่วนที่สองคือการตรวจสอบการปลดบล็อก(Release) และส่วนสุดท้ายคือการตรวจสอบ time out ของ Working Detect(WD) นอกจากนี้ยังมีส่วนที่สำคัญอีกส่วนหนึ่งคือไฟล์ข้อมูล 4 ไฟล์ได้แก่ Pattern file (ในภาพจะใช้สัญลักษณ์เป็นP), Working_Detected(ในภาพจะใช้สัญลักษณ์เป็นWD), Working_Released (ในภาพจะใช้สัญลักษณ์เป็นWR) และ History file(ในภาพจะใช้สัญลักษณ์เป็นH) และส่วนสำคัญอีกส่วนหนึ่งคือส่วนของโปรแกรมภายนอกซึ่งใช้ในการติดต่อกับไฟร์วอลล์ในโปรแกรมที่ได้พัฒนาขึ้นจะเรียกใช้โปรแกรมภายนอก 3 โปรแกรมได้แก่ Attack Action(AA), Release Poll(RP) และ Release Action(RA) ซึ่งจะกล่าวถึงรายละเอียดต่อไป



ภาพที่ 3.8 แสดงภาพรวมการทำงานของโปรแกรม

3.4.1 รูปแบบข้อมูลที่ใช้ในการทำงานของโปรแกรม

ก่อนที่จะกล่าวถึงรายละเอียดการทำงานของโปรแกรมจะกล่าวถึงรูปแบบข้อมูลทั้ง 4 ไฟล์ที่เกี่ยวข้องกับการทำงาน ดังต่อไปนี้

1) Pattern File มีรูปแบบซึ่งแสดงดังภาพที่ 3.9

P_ID	IP address/mask	URL	Method	Attack_Num	Attack_Period
Attack_Action	Release_Poll	Release_Num	Release_Period	Release_Action	

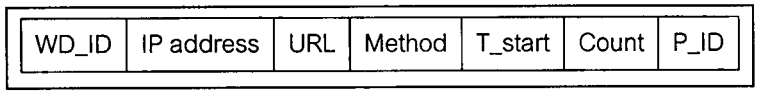
ภาพที่ 3.9 แสดงรูปแบบของ Pattern file

ใน Pattern file จะเป็นไฟล์สำหรับให้ผู้ใช้งานกำหนดรูปแบบการโจมตีซึ่งสามารถระบุได้ตามคอลัมน์ต่างๆมีทั้งหมด 11 คอลัมน์โดยแต่ละคอลัมน์จะมีความหมายแสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงความหมายและประเภทข้อมูลในแต่ละคอลัมน์ของ Pattern file

ชื่อคอลัมน์	ประเภทข้อมูล	ความหมาย
P_ID	unsigned int	หมายเลขลำดับของการกำหนดรูปแบบโดยเริ่มต้นตั้งแต่หมายเลข 1 เป็นต้นไป
IP address/ Mask	char 32	หมายเลข IP address และ subnet mask ของ Client ที่ต้องการตรวจสอบการโจมตี โดยระบุเป็นรูปแบบดังตัวอย่างต่อไปนี้ 161.246.49.147/255.255.255.192
URL	char 255	URL ที่เป็นรูปแบบของการโจมตีที่ต้องการตรวจสอบ เช่น http://www.worm.com/default.ida? ระบุได้ไม่เกิน 255 ตัวอักษร
Method	char 10	วิธีการร้องขอออบเจ็กต์ เช่น GET, POST, HEAD, PUT, DELETE, TRACE, OPTIONS, CONNECT เป็นต้น
Attack_Num	unsigned int	กำหนดจำนวนการร้องขอออบเจ็กต์ที่ซ้ำๆกันเพื่อใช้ตรวจสอบการโจมตีมีหน่วยเป็นจำนวนครั้งของการร้องขอออบเจ็กต์
Attack_Period	double	กำหนดช่วงเวลาเพื่อตรวจสอบการโจมตีโดยมีหน่วยเป็นวินาที
Attack_Action	char 50	กำหนด path และชื่อไฟล์ที่ใช้กำหนดคำสั่งที่ต้องการให้กระทำเมื่อมีการโจมตีเกิดขึ้นเช่น /usr/local/IdsWC/block เป็นต้น
Release_Poll	char 50	กำหนด path และชื่อไฟล์ที่ใช้กำหนดคำสั่งที่ต้องการให้ทำเพื่อต้องการสอบถามจำนวนนับการโจมตีเช่น /usr/local/IdsWC/poll เป็นต้น
Release_Num	unsigned int	จำนวนการร้องขอออบเจ็กต์ซึ่งตรวจพบที่ไฟร์วอลใช้เพื่อตรวจสอบการหยุดโจมตี
Release_Period	double	เป็นช่วงเวลาที่ใช้ตรวจสอบการหยุดโจมตี
Release_Action	char 50	กำหนด path และชื่อไฟล์ที่ใช้บันทึกรูปกำหนดให้กระทำเมื่อต้องยกเลิกปฏิเสธการโจมตีเช่น /usr/local/IdsWC/release เป็นต้น

2) Working Detected มีรูปแบบแสดงดังภาพที่ 3.10



ภาพที่ 3.10 แสดงรูปแบบของ Working Detected และ Working Released

เป็นไฟล์ข้อมูลที่ใช้ทำงานในส่วนของการตรวจจับการโจมตี ซึ่งเก็บข้อมูลการร้องขอออบเจ็กต์ที่มีโอกาสถูกโจมตีประกอบด้วย 7 คอลัมน์ โดยในแต่ละคอลัมน์จะมีความหมายแตกต่างกันดังตารางที่ 3.2

ตารางที่ 3.2 แสดงความหมายและประเภทข้อมูลในแต่ละคอลัมน์ของWorking Detected

ชื่อคอลัมน์	ประเภทข้อมูล	ความหมาย
WD_ID	unsigned int	เก็บลำดับการบันทึกการร้องขอออบเจ็กต์ที่ตรวจพบว่าเข้าคู่กับรูปแบบของ pattern file
IP address	char 16	เก็บค่า IP address ของการร้องขอออบเจ็กต์ซึ่งได้รับมาจาก access.log
URL	char 256	เก็บค่า URL ของการร้องขอออบเจ็กต์ซึ่งได้รับมาจาก access.log
Method	char 10	เก็บค่า method ของ การร้องขอออบเจ็กต์ซึ่งได้รับมาจาก access.log
T_start	double	เก็บค่าของเวลาครั้งแรกที่มีการร้องขอออบเจ็กต์ซึ่งมีรูปแบบตรงตาม pattern file
Count	unsigned int	เป็นจำนวนนับของการร้องขอออบเจ็กต์โดยจะเพิ่มขึ้นครั้งละ 1 เมื่อพบการร้องขอที่ซ้ำซ้อนกัน
P_ID	unsigned int	เป็นหมายเลขของรูปแบบใน Pattern file ที่เข้าคู่กับการร้องขอออบเจ็กต์นี้

3) Working Released มีรูปแบบแสดงดังภาพที่ 3.10

Working Released มีรูปแบบเหมือนกับ Working Detected เนื่องจากมีการใช้งานเพื่อเก็บข้อมูลที่มีรูปแบบลักษณะเดียวกัน แต่จุดประสงค์ในการใช้งานแตกต่างกันคือ Working Released เก็บข้อมูลการร้องขอที่ตรวจพบว่าเป็นการโจมตีและรอการตรวจสอบเพื่อยกเลิกการปฏิเสธการร้องขอออบเจ็กต์ โดยตารางที่ 3.3 แสดงความหมายของแต่ละคอลัมน์ที่แตกต่างจาก Working Detected ดังนี้

ตารางที่ 3.3 แสดงความหมายและประเภทข้อมูลในแต่ละคอลัมน์ของ Working Released

ชื่อคอลัมน์	ประเภทข้อมูล	ความหมาย
WD_ID	unsigned int	เก็บลำดับการบันทึกการร้องขอออบเจ็กต์ที่เป็นการโจมตี
IP address	char 16	เก็บค่า IP address ของการร้องขอออบเจ็กต์ซึ่งได้รับมาจาก working detect
URL	char 256	เก็บค่า URL ของการร้องขอออบเจ็กต์ซึ่งได้รับมาจาก working detect
Method	char 10	เก็บค่า method ของ การร้องขอออบเจ็กต์ซึ่งได้รับมาจาก working detect
T_start	double	เก็บค่าของเวลาปัจจุบันซึ่งจะเปลี่ยนแปลงทุกครั้งเมื่อมีการตรวจสอบเพื่อยกเลิกการโจมตี
Count	unsigned int	เก็บค่าจำนวนนับของการโจมตี
P_ID	unsigned int	เป็นหมายเลขรูปแบบใน Pattern file ที่เข้าคู่กับการร้องขอออบเจ็กต์นี้

4) History File มีรูปแบบซึ่งแสดงดังภาพที่ 3.11

T_record	Table	Reason	ID	IP address	URL	Method	T_start	Count	P_ID

ภาพที่ 3.11 แสดงรูปแบบของ History file

ไฟล์นี้ทำหน้าที่เก็บการร้องขอออบเจ็กต์ต่างๆที่เคยเกิดขึ้นจากการทำงานของโปรแกรม ซึ่ง History file สามารถนำมาตรวจสอบการทำงานของโปรแกรมได้ History file ประกอบไปด้วยคอลัมน์ทั้งหมด 11 คอลัมน์โดยในคอลัมน์ส่วนที่แรเงาดังภาพที่3.11 เป็นข้อมูลที่มาจาก Working Detected หรือ Working Released ดังนั้นรายละเอียดของคอลัมน์ที่เพิ่มเติมมี 3 คอลัมน์แสดงดังในตารางที่ 3.4

ตารางที่ 3.4 แสดงความหมายของคอลัมน์และตัวเลือกใน history file

ชื่อคอลัมน์	ประเภทข้อมูล	ความหมาย
T_record	double	เก็บค่าของเวลาบันทึกข้อมูลการร้องขอลงใน History file
Table	char	เป็นชื่อแหล่งข้อมูลได้แก่ Working Detected หรือ Working Released ซึ่งจะบ่งบอกถึงที่มาของ record ที่ได้บันทึกลงไปใน History file
Reason	char	เก็บเหตุผลของการบันทึกซึ่งมีความหมายแสดงดังตารางที่3.5

เหตุผลของการบันทึกข้อมูลลงใน history file จะแสดงรายละเอียดของความหมายในแต่ละเหตุผลที่แตกต่างกันในตารางที่ 3.5 ดังต่อไปนี้

ตารางที่ 3.5 แสดงความหมายของเหตุผลต่างๆใน history file

หมายเลขเหตุผล	ความหมาย
1	■ Release (time out เปลี่ยนสถานะจาก detected เป็น release) จากการร้องขอออบเจ็กต์ในตาราง Working Detected ■ ตรวจสอบจากโปรแกรม detect เมื่อได้รับการร้องขอในรูปแบบที่มีการบันทึกใน Working Detected แล้ว(ขึ้นอยู่กับการร้องขอที่รับเข้ามา)
2	■ attacked (เปลี่ยนสถานะจาก detected เป็น attacked) ■ ตรวจพบการโจมตีของการร้องขอออบเจ็กต์ในตาราง Working Detected จึงบันทึกการร้องขอนี้ลง working release และ history file
3	■ Release (time out เปลี่ยนสถานะจาก detected เป็น release)จากการร้องขอออบเจ็กต์ในตาราง Working Detected ■ ตรวจสอบจากโปรแกรม check working time out ซึ่งจะมีการตรวจสอบ time out โดยไม่ขึ้นกับการร้องขอที่รับเข้ามา
4	■ error (เปลี่ยนสถานะจาก detected เป็น release) ■ เป็นความผิดพลาดที่เกิดจากการตรวจสอบไม่พบหมายเลขของ pattern file ที่ตรงกับการร้องขอออบเจ็กต์ใน working detected
5	■ error (เปลี่ยนสถานะจาก attacked เป็น release) ■ เป็นความผิดพลาดที่เกิดจากการตรวจสอบไม่พบหมายเลขของ pattern file ตรงกับการร้องขอออบเจ็กต์ใน working released
6	■ Release (เปลี่ยนสถานะจาก attacked หรือ release) ■ ตรวจสอบไม่พบการโจมตีใน working release

3.4.2 ส่วนโปรแกรมภายนอกที่เกี่ยวข้อง

การทำงานของโปรแกรมจำเป็นต้องติดต่อกับโปรแกรมภายนอกในส่วนการจำกัดการโจมตี ดังภาพที่ 3.8 จะเห็นว่าโปรแกรมทำงานติดต่อกับโปรแกรมภายนอกเพื่อปฏิเสธการร้องขอหรือยกเลิกปฏิเสธดังนั้นต้องทำความเข้าใจในส่วนของโปรแกรมภายนอกมี 3 ส่วนดังนี้

3.4.2.1 Attack Action

เป็นโปรแกรมภายนอกที่ถูกโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชเรียกใช้เมื่อตรวจพบการโจมตี การเรียกใช้โปรแกรมภายนอกนี้จะได้รับการกำหนดตำแหน่งและชื่อของโปรแกรมใน Pattern File จากผู้ใช้งานตามรูปแบบที่ได้กล่าวมาแล้วในหัวข้อที่ 3.4.1 การทำงานของโปรแกรมย่อย Attack Action นี้จะมีรูปแบบการเรียกใช้แสดงดังภาพที่ 3.12 ดังนี้

ภาพที่ 3.12 แสดงรูปแบบการเรียกใช้งานโปรแกรมภายนอก Attack Action

ส่วนตารางที่ 3.6 จะแสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Attack Action ดังต่อไปนี้

ตารางที่ 3.6 แสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Attack Action

ชื่อคอลัมน์	ความหมาย
Att_action name name	ที่อยู่และชื่อของโปรแกรมภายนอก Attack Action ที่เรียกใช้เมื่อตรวจพบการโจมตี
Client IP	เป็นพารามิเตอร์ซึ่งหมายถึงหมายเลขของ IP address ของเครื่อง Client เป้าหมายที่ตรวจพบว่าโจมตีเครื่อง Server
Server IP	เป็นพารามิเตอร์ซึ่งหมายถึงหมายเลขของ IP address ของเครื่อง Server ที่ถูกเครื่อง Client เป้าหมายโจมตี

ผลของการเรียกใช้โปรแกรมภายนอก Attack Action จะได้ค่าหมายเลขอ้างอิงของไฟร์วอลล์ที่ตรงกับการปฏิเสธการร้องขอข้อมูล โดยโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชจะใช้เทคนิคการบันทึกและอ่านค่าหมายเลขอ้างอิงดังกล่าวจากไฟล์ข้อมูลซึ่งมีรูปแบบของการบันทึกลงในไฟล์แสดงดังภาพที่ 3.13 และแสดงความหมายของแต่ละคอลัมน์ดังตารางที่ 3.7

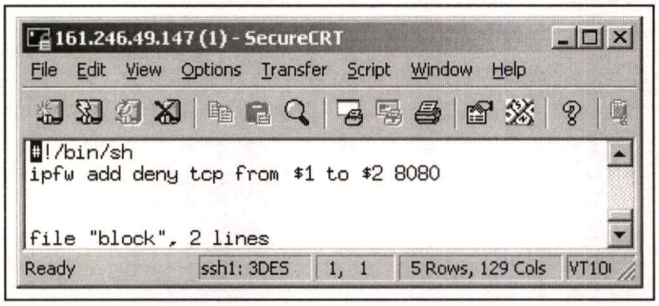
Ref_ID	action	Protocol	form	Client IP	to	Server IP	Port

ภาพที่ 3.13 แสดงรูปแบบไฟล์ที่เก็บผลการทำงานของโปรแกรมภายนอก Attack Action

ตารางที่ 3.7 แสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Attack Action

ชื่อคอลัมน์	ความหมาย
Ref_ID	หมายเลขอ้างอิงที่ตรงกับการปฏิเสธการโจมตี
action	เป็น action ของคำสั่ง IPFW ซึ่งได้อธิบายไว้ในบทที่ 2
Protocol	เป็น Protocol ที่ได้กำหนดให้มีการปฏิเสธการร้องขอซึ่งได้อธิบายไว้ในบทที่ 2
from	ดูคำอธิบายในบทที่ 2
Client IP	หมายเลขของ IP address ของเครื่อง Client เป้าหมายที่ตรวจพบว่าโจมตีเครื่อง Server
to	ดูคำอธิบายในบทที่ 2
Server IP	หมายเลขของ IP address ของเครื่อง Server ที่ถูกเครื่อง Client เป้าหมายโจมตี
Port	หมายเลข Port

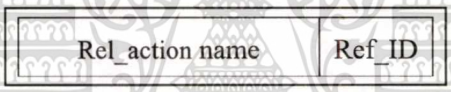
จากภาพที่ 3.13 และตารางที่ 3.7 จะเป็นลักษณะเฉพาะของการทำงานโปรแกรมโปรแกรมภายนอกที่ได้กำหนดไว้ หลังจากได้บันทึกลงไฟล์ข้อมูลแล้วโปรแกรมจะอ่านค่าของ ID ของกฎของ IPFW เก็บไว้เป็นหมายเลขอ้างอิงเพื่อเรียกใช้งานต่อไป ตัวอย่างของโปรแกรมภายนอก Attack Action แสดงดังภาพที่ 3.14 ซึ่งเป็นสคริปต์ที่เขียนด้วยภาษาเชลล์สคริปต์



ภาพที่ 3.14 แสดงตัวอย่างโปรแกรมภายนอก Attack Action

3.4.2.2 Release Action

เป็นโปรแกรมภายนอกที่ถูกโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชเรียกใช้เช่นกันกับ Attack Action แต่จะเรียกใช้เมื่อตรวจสอบไม่พบการโจมตีจึงต้องการยกเลิกการป้องกันการโจมตีซึ่งโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชมีรูปแบบการแบบการเรียกใช้แสดงดังภาพที่ 3.15 ดังนี้



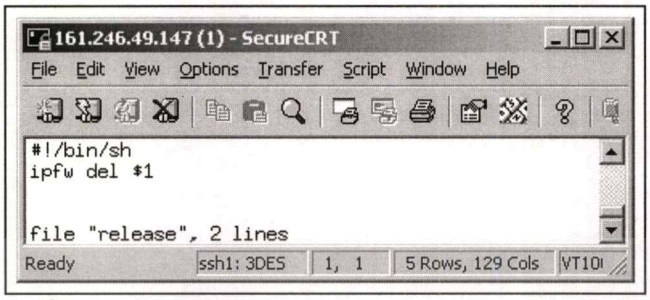
ภาพที่ 3.15 แสดงรูปแบบการเรียกใช้งานโปรแกรมภายนอก Release Action

จากภาพที่ 3.15 สามารถอธิบายความหมายของแต่ละคอลัมน์ได้ดังตารางที่ 3.8

ตารางที่ 3.8 แสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Release Action

ชื่อคอลัมน์	ความหมาย
Rel_action name	ที่อยู่และชื่อของโปรแกรมภายนอก Release Action ที่เรียกใช้เมื่อตรวจไม่พบการโจมตี
Ref_ID	เป็นพารามิเตอร์ซึ่งหมายถึงหมายเลขของ Reference ของไฟร์วอลล์ที่เกี่ยวข้องกับการปฏิเสธการโจมตีที่ได้จากในหัวข้อที่แล้วและบันทึกไว้

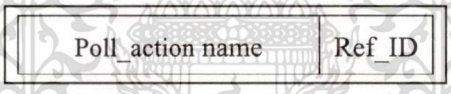
ตัวอย่างของโปรแกรมภายนอก Release Action ซึ่งเขียนด้วยภาษาเชลล์สคริปต์อย่างง่ายดังภาพที่ 3.16



ภาพที่ 3.16 แสดงตัวอย่างโปรแกรมภายนอก Release Action

3.4.2.3 Release Poll

เป็นโปรแกรมภายนอกที่ถูกโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชเรียกใช้เช่นกันแต่จะเรียกใช้เมื่อต้องการตรวจสอบจำนวน packet ที่เครื่องโจมตีส่งมายังเครื่อง Server ซึ่งโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชมีรูปแบบการเรียกใช้งานดังภาพที่ 3.17 และความหมายของแต่ละคอลัมน์แสดงตารางที่ 3.9



ภาพที่ 3.17 แสดงรูปแบบการเรียกใช้งานโปรแกรมภายนอก Release Poll

ตารางที่ 3.9 แสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Release Poll

ชื่อคอลัมน์	ความหมาย
Rel_action name	ที่อยู่และชื่อของโปรแกรมภายนอก Release Poll ที่เรียกใช้เมื่อต้องการสอบถามจำนวนของการโจมตี
Ref_ID	พารามิเตอร์ซึ่งหมายถึงหมายเลขของ Reference ของไฟร์วอลล์ที่เกี่ยวข้องกับการปฏิเสธการโจมตี

ผลจากการทำงานของโปรแกรมภายนอก Release Poll ได้จำนวนการร้องขอข้อมูลเป็น Packet จากเครื่องเป้าหมายมายังเครื่อง Server โดยโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชจะใช้เทคนิคการอ่านและเขียนไฟล์เพื่อเก็บผลจากการทำงานดังกล่าวเช่นเดียวกันกับการเก็บผลของAttack Action ซึ่งรูปแบบของไฟล์ข้อมูลดังกล่าวจะเก็บข้อมูลดังภาพที่ 3.18

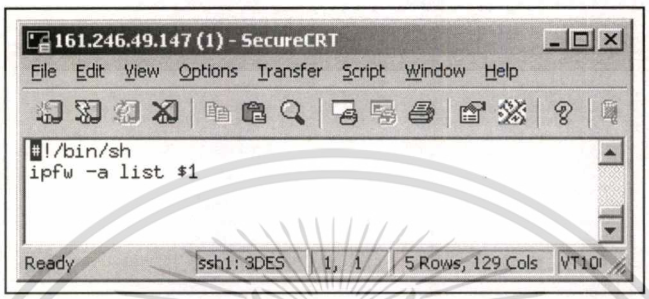
Ref_ID	Packet_num	Byte_num	action	protocal	from	Client IP	to	Server IP	port

ภาพที่ 3.18 แสดงรูปแบบไฟล์ที่เก็บผลการทำงานของโปรแกรมภายนอก Release Poll

ตารางที่ 3.10 แสดงความหมายของรูปแบบการเรียกใช้งานโปรแกรมภายนอก Attack Action

ชื่อคอลัมน์	ความหมาย
Ref_ID	หมายเลขอ้างอิงที่ตรงกับการปฏิเสธการโจมตี
Packet_num	จำนวนการร้องขอคิดเป็น Packet
Byte_num	จำนวนการร้องขอคิดเป็น Byte
action	เป็น action ของคำสั่ง IPFW ซึ่งได้อธิบายไว้ในบทที่ 2
Protocol	เป็น Protocol ที่ได้กำหนดให้มีการปฏิเสธการร้องขอซึ่งได้อธิบายไว้ในบทที่ 2
from	ดูคำอธิบายในบทที่ 2
Client IP	หมายเลขของ IP address ของเครื่อง Client เป้าหมายที่ตรวจพบว่าโจมตีเครื่อง Server
to	ดูคำอธิบายในบทที่ 2
Server IP	หมายเลขของ IP address ของเครื่อง Server ที่ถูกเครื่อง Client เป้าหมายโจมตี
port	หมายเลข Port เช่น 8080

เมื่อมีการบันทึกข้อมูลนี้ลงไฟล์แล้วโปรแกรมจะอ่านค่า Packet_num เพื่อนำมาตรวจสอบว่ายังคงมีการโจมตีเกิดขึ้นหรือไม่และจะใช้ค่านี้บันทึกลงในข้อมูลการทำงานของโปรแกรมเมื่อการร้องขอนี้ยังคงเป็นการโจมตีอยู่ ตัวอย่างของโปรแกรมภายนอก Release Poll ที่เขียนด้วยภาษาเชลล์สคริปต์แสดงดังภาพที่ 3.19



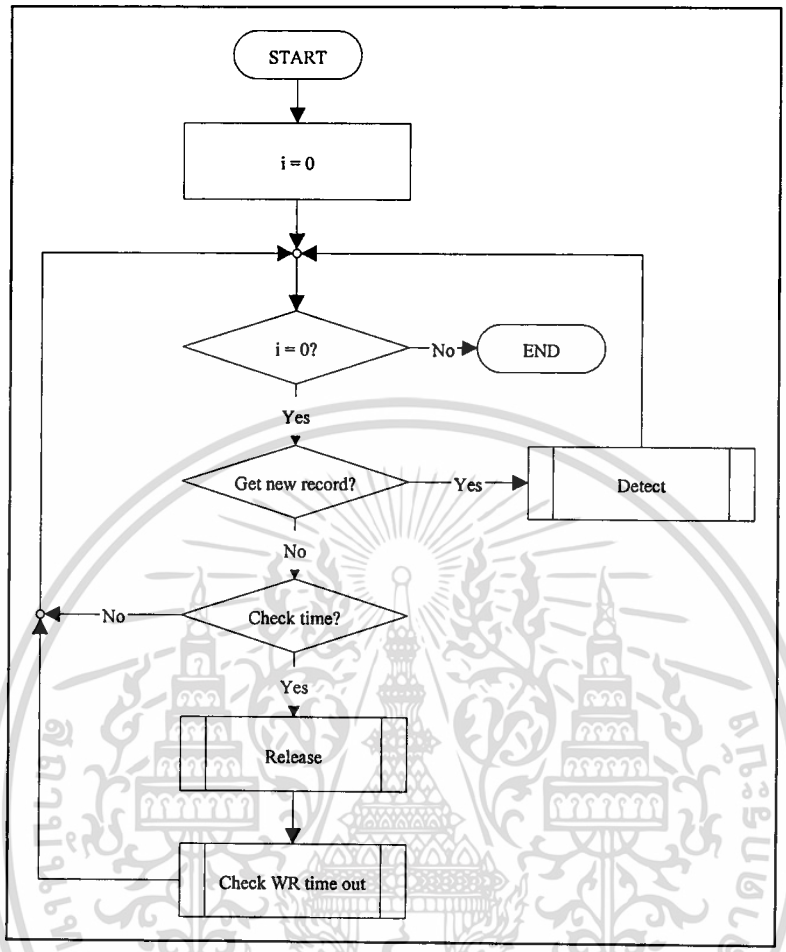
ภาพที่ 3.19 แสดงตัวอย่างโปรแกรมภายนอก Release Poll

3.4.3 หน้าที่และการทำงานของส่วนการทำงานย่อยของโปรแกรม

จากภาพที่ 3.8 แสดงภาพรวมของโปรแกรมซึ่งประกอบด้วยส่วนการทำงานย่อยๆ ทั้งหมด 3 ส่วน ในหัวข้อนี้จะอธิบายหน้าที่และลำดับการทำงานของส่วนย่อยของโปรแกรมทั้ง 3 ส่วนโดยส่วนการทำงานย่อยของโปรแกรมทั้ง 3 จะถูกควบคุมจากการทำงานของส่วนโปรแกรมหลักซึ่งสามารถแสดงได้ดังผังการทำงานดังภาพที่ 3.20 ส่วนลำดับของการทำงานในส่วนอื่นๆจะแสดงดังภาพที่ 3.21-3.23 ตามลำดับ ดังต่อไปนี้

3.4.3.1 ลำดับการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

จากภาพลำดับการทำงานของโปรแกรมหลักจะเห็นว่ามีการวนซ้ำตลอดเวลาเพื่อตรวจสอบข้อมูลเข้าซึ่งรับมาจาก access.log หากพบข้อมูลเข้าจะส่งการทำงานให้กับโปรแกรมย่อย Detect ดังภาพที่ 3.16 เพื่อตรวจจับการโจมตี เมื่อดำเนินการเสร็จในแต่ละรอบจะวนทำงานเช่นนี้อีกซ้ำ แต่หากไม่พบข้อมูลเข้าจาก access.log ตรวจสอบเวลาหากถึงกำหนดเวลาจะส่งการทำงานให้กับโปรแกรมย่อยอีก 2 ส่วน ได้แก่ Release ซึ่งทำหน้าที่ตรวจสอบการโจมตีและส่งคำสั่งยกเลิกการปฏิเสธการร้องขอให้แก่ไฟร์วอลล์แสดงต่อไปในภาพที่ 3.17 หลังจากทำงานในส่วน Release แล้วจะทำงานในส่วนของโปรแกรมย่อยอีกส่วนหนึ่งคือตรวจสอบ time out ของ working Detect ซึ่งจะตรวจสอบการหมดเวลาของข้อมูลใน Working Detected ลำดับการทำงานแสดงในภาพที่ 3.18



ภาพที่ 3.20 แสดงลำดับการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

3.4.3.2 หน้าที่และลำดับการทำงานของส่วนการตรวจจับการโจมตี (Detected)

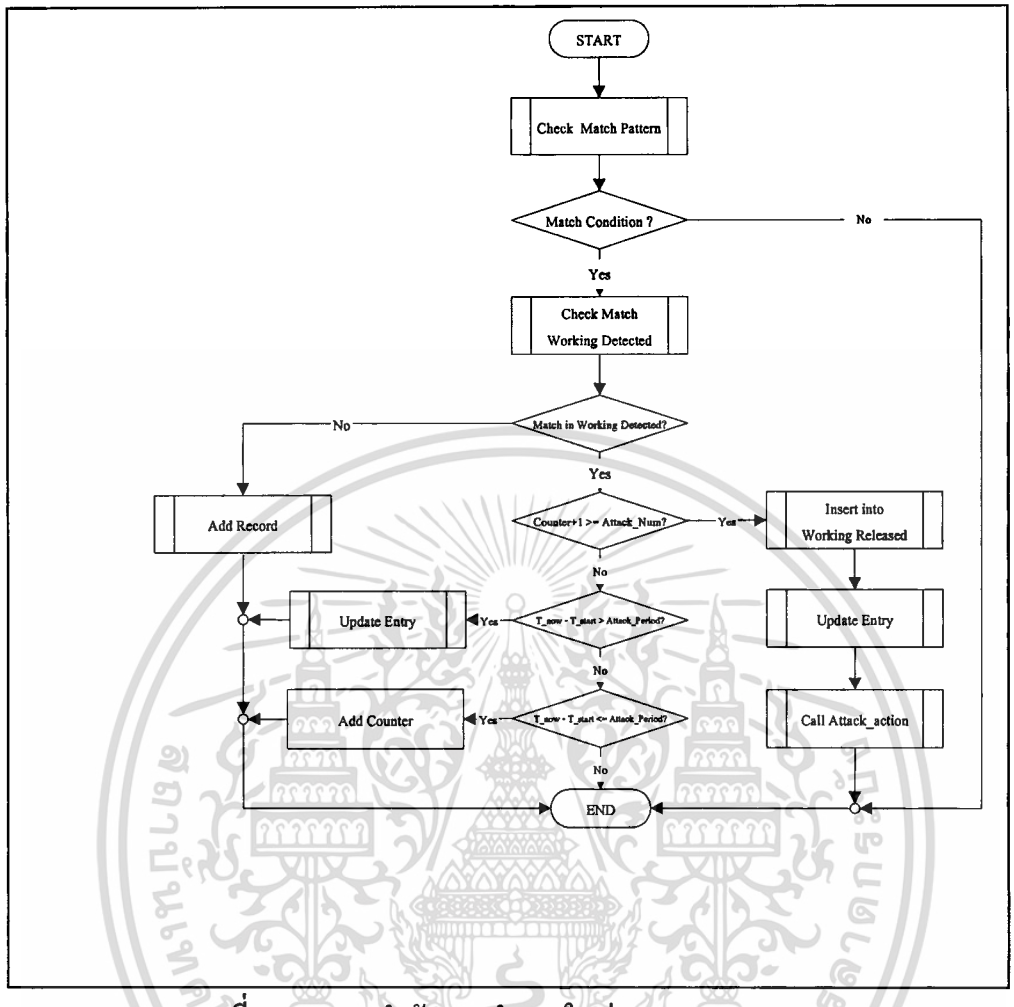
โปรแกรมย่อยส่วน Detected จะทำงานเกี่ยวข้องกับส่วนเก็บข้อมูลและเกี่ยวข้องโดยอ้อมกับข้อมูลใน access.log เนื่องจากการทำงานของโปรแกรมหลักจะทำงานเกี่ยวข้องโดยตรงคือรับข้อมูลแล้วจะส่งข้อมูลให้กับส่วนการทำงานของ Detect นอกจากนั้นส่วนของ Detect จะทำงานเกี่ยวข้องกับส่วนติดต่อกับไฟร์วอลล์อีกด้วย การทำงานของส่วน Detect ทำหน้าที่หลักดังต่อไปนี้

- ตรวจสอบการร้องขอและพิจารณาการโจมตีจากจำนวนและเวลาของการร้องขอข้อมูลดังกล่าว
- ส่งคำสั่งเพื่อปฏิเสธการร้องขอไปยังไฟร์วอลล์
- มีส่วนเกี่ยวข้องกันการตรวจสอบ time out

- บันทึกข้อมูลที่เกิดการเปลี่ยนสถานะต่างๆ ลงใน history file

ลำดับการทำงานในส่วนของโปรแกรมย่อย Detected สามารถอธิบายการทำงานได้ดังต่อไปนี้

1) ค้นหา record ใน Pattern file ที่เข้าคู่ได้กับ new record หากไม่พบทำในข้อ 2) หากพบทำในข้อ 3)
2) สิ้นสุดการทำงานในส่วนนี้
3) ค้นหา record ใน Working Detected ที่ตรงกับ new record หากไม่พบทำให้ข้อ 4) พบทำในข้อ 5)
4) เพิ่ม node ของ new record นี้ลงใน Working Detected เพื่อใช้ในการตรวจสอบหาการโจมตีต่อไป แล้วทำข้อ 2) ต่อไป
5) ตรวจสอบจำนวนนับถ้าหากพบจำนวนที่มากกว่าที่กำหนดไว้ไม่ว่าเวลาจะน้อยกว่า, เท่ากับ หรือมากกว่าก็ตามจะถือว่าเป็นการโจมตี (หากเวลามากกว่าจะมีส่วนของการตรวจสอบ time out ทำหน้าที่ตรวจสอบเพื่อลบข้อมูลที่ไม่จำเป็นนี้ออกโดยอัตโนมัติ) การตรวจสอบจำนวนนับจะหมายถึง Count+1 >= Attack_Num หากค่าเป็นจริงจะทำต่อในข้อ 6) แต่หาก Count+1 < Attack_Num ทำในข้อ 7)ต่อไป
6) เมื่อมีจำนวนนับมากกว่าจำนวนที่กำหนดไว้หมายความว่า record ดังกล่าวตรงตามรูปแบบการโจมตีที่ได้กำหนดไว้ทุกประการดังนั้นโปรแกรมจะทำดังนี้
   - เพิ่ม node ใหม่ของ new record ลงใน Working Release
   - Update entry หมายถึงการบันทึกข้อมูล record นี้ซึ่งอยู่ใน Working Detected ลงใน history file ลบ record นี้ซึ่งอยู่ใน Working Detected ออกไปเนื่องจากไม่มีประโยชน์ต่อโปรแกรมนี้อีกแล้ว
   - เรียกโปรแกรมภายนอกคือ Attack_Action ซึ่งเป็นการส่งคำสั่งไปให้ไฟร์วอลล์ปฏิเสธการร้องขอข้อมูล หลังจากทำทั้ง 3 ขั้นตอนเสร็จเรียบร้อยแล้วจะทำต่อข้อ 2)
7) ตรวจสอบผลต่างของเวลาดังเงื่อนไขต่อไปนี้ T_now – T_start > Attack_Period หากตรงตามเงื่อนไขนี้มีหมายความว่าช่วงเวลาที่ต้องการตรวจสอบล่วงเลยไปแล้วแต่จำนวนที่ตรวจพบยังคงน้อยกว่าจำนวนที่กำหนดไว้ดังนั้น node นี้ใน Working Detected ถือเป็น time out ทำข้อ 8) แต่หากไม่ตรงจะทำข้อ 9)
8) จากข้อ 7) time out node ดังกล่าวใน Working Detect นี้ไม่มีประโยชน์ต้อง Update Entry หลังจากนั้นจะทำข้อ 2)



ภาพที่ 3.21 แสดงลำดับการทำงานในส่วนของการDetected

9) จากข้อ 7)ต้องตรวจสอบต่อไปคือใช้เงื่อนไข T_now – T_start <= Attack_Period หากตรงตามเงื่อนไขนี้หมายความว่าช่วงเวลาที่ต้องการตรวจสอบยังคงอยู่ในช่วงของเวลาที่ได้กำหนดไว้ แต่ยังมีค่าจำนวนนับไม่ครบตามกำหนดดังนั้นจะทำข้อ 10) และหากไม่ตรงตามเงื่อนไขจะทำข้อ 2)

10) จากข้อ 9) ต้องเพิ่มจำนวนนับครั้งละ 1 ให้กับ counter ใน Working Detect เมื่อเสร็จสิ้นการทำงานจะทำข้อ 2) ต่อไป

หมายเหตุ T_now จะหมายถึง เวลา ณ ปัจจุบันที่ใช้ในการตรวจสอบ new record ที่ได้รับมาจากโปรแกรมหลัก

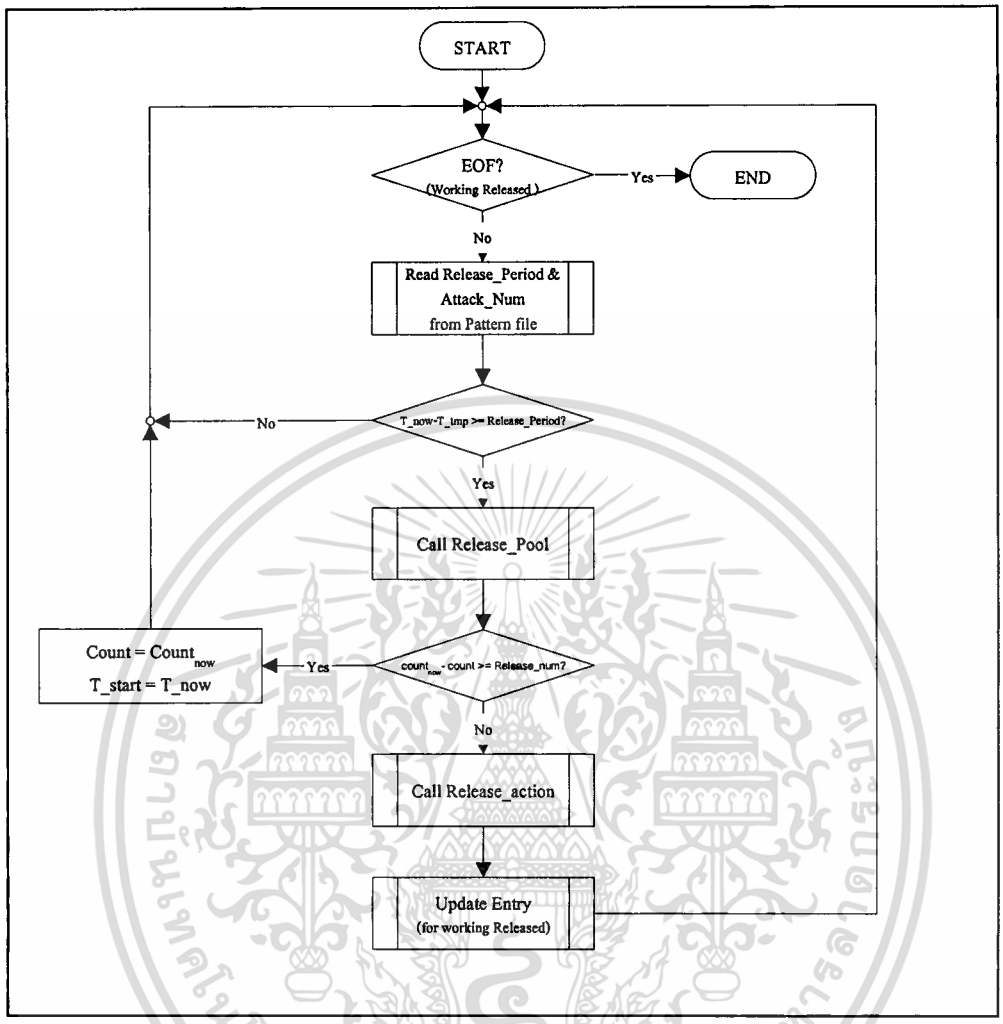
3.4.3.3 หน้าที่การทำงานของการตรวจสอบการปลดบล็อก(Release)

โปรแกรมย่อยส่วนของการตรวจสอบการปลดบล็อก(Release) การทำงานส่วนนี้เกี่ยวข้องกับส่วนเก็บข้อมูลได้แก่ Working Released(WR), Pattern file(P) และ History File(P) และที่สำคัญคือเกี่ยวข้องกับส่วนติดต่อกับไฟร์วอลล์หน้าที่การทำงานของส่วนนี้มีดังต่อไปนี้

- เรียกใช้ RP หมายถึง Release Poll เพื่อหาจำนวนนับที่ไฟล์วอลล์
- ตรวจสอบจำนวนนับซึ่งเป็นการโจมตีของ node ใน Working Released ตามเวลาที่กำหนดหากจำนวนน้อยกว่าที่กำหนดจะส่งคำสั่งให้ไฟร์วอลล์ยกเลิกปฏิเสธการร้องขอนั่นคือการเรียกใช้งาน RA หรือ Release action นั่นเอง
- ตรวจสอบ time out ของข้อมูลใน Working Released
- บันทึกข้อมูลที่เกิดการเปลี่ยนสถานะต่างๆ ลงใน history file

ลำดับการทำงานในส่วน Released สามารถอธิบายได้ดังต่อไปนี้

1) ตรวจสอบจนกระทั่งข้อมูลใน Working Release หมด หากยังไม่หมดทำข้อ 3) หากตรวจสอบทุกข้อมูลจนหมดแล้วทำข้อ 2)
2) สิ้นสุดการทำงานในส่วนของโปรแกรมย่อย Released
3) อ่านค่า ช่วงเวลาและจำนวนการ Release จาก Pattern file แล้วทำข้อ 4) ต่อไป
4) ตรวจสอบเวลาดังนี้
   - ถ้า T_now – T_start >= Attack_period หมายถึงครบกำหนดตามช่วงเวลาที่ต้องการตรวจสอบแล้วซึ่งระบุใน pattern file ดังนั้นทำต่อข้อ 5)
   - T_now – T_start < Attack_period หมายถึง ยังไม่ถึงช่วงเวลาที่ต้องการให้ตรวจสอบดังนั้นจะทำข้อ 1)
5) เรียกโปรแกรมที่ทำหน้าที่ส่งคำสั่งสอบถามไปยังไฟร์วอลล์เพื่อถามจำนวน Counter แล้ว ทำข้อ6) ต่อไป



ภาพที่ 3.22 แสดงลำดับการทำงานในส่วนของการ Released

6) ตรวจสอบจำนวนนับดังต่อไปนี้

- $count_{now}$ – count >= Release_num หมายถึงยังคงมีการโจมตีอยู่เนื่องจากจำนวนของการร้องขอออบเจ็กต์ที่ไฟร์วอลล์ยังมีมากเกินจำนวนที่กำหนดทำข้อ 7) ต่อไป
- $count_{now}$ – count < Release_num หมายถึงสิ้นสุดการโจมตีเนื่องจากจำนวนการร้องขอออบเจ็กต์ที่ไฟร์วอลล์มีน้อยกว่าจำนวนที่กำหนด ดังนั้นถือว่าการร้องขอข้อมูลเปลี่ยนสถานะจาก Attack เป็น Release ทำขั้นตอนที่ 8) ต่อไป

7) ปรับปรุงค่า $count_{now}$ ที่ได้รับมา ลงใน Working Release แทน count เดิม และปรับปรุงค่าเวลาโดยบันทึกเวลาปัจจุบันแทนเวลาเดิมแล้วทำการวนซ้ำโดยการทำข้อ 1)
8) เรียกโปรแกรมเพื่อให้ส่งคำสั่งไปยังไฟร์วอลล์ทำการยกเลิกการปฏิเสธการร้องขอแล้วทำข้อ 9)
9) Update Entry แล้วทำการวนซ้ำโดยการทำข้อ 1)

หมายเหตุ $count_{now}$ จะหมายถึง จำนวนนับที่ได้จากการ Poll ถามที่ไฟร์วอลล์ส่วน count เป็นจำนวนนับที่บันทึกใน Working Release

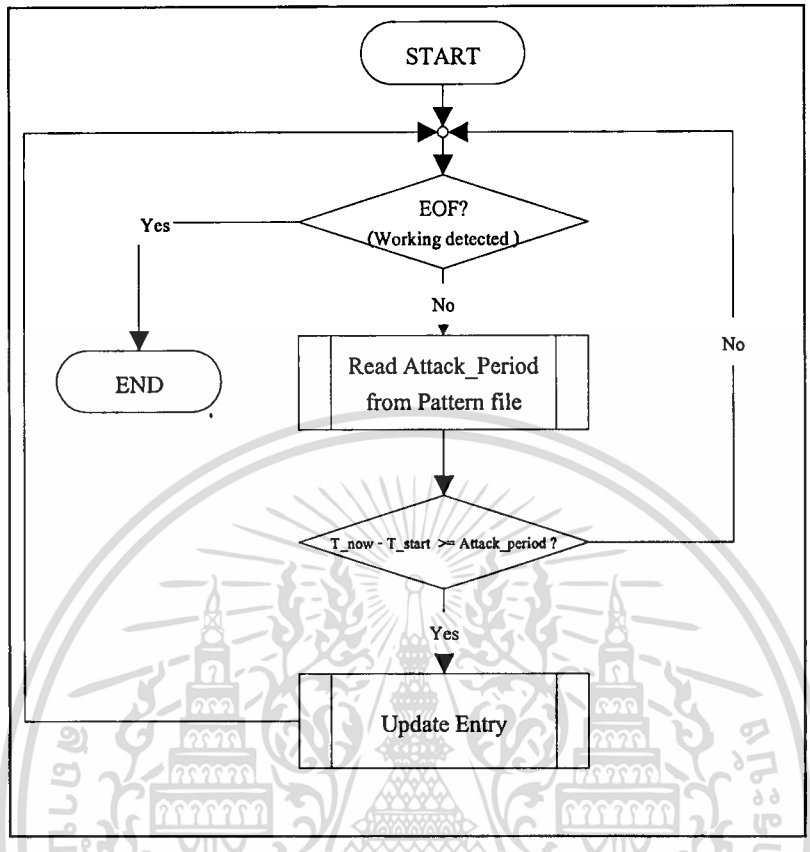
3.4.2.4 หน้าที่การทำงานของการตรวจสอบ time out (Check WD)

โปรแกรมย่อยส่วนสุดท้ายจะเป็นการปรับปรุงข้อมูลใน Working Detect ที่ไม่มีประโยชน์ในการทำงานของโปรแกรมแล้วจึงต้องมีการลบออกจาก Working Detect เกี่ยวข้องกับส่วนเก็บข้อมูลเพียงอย่างเดียวอันได้แก่ Working Detected(WD), Pattern file(P) และ History File(P) มีหน้าที่หลักๆ ดังต่อไปนี้

- ตรวจสอบข้อมูล time out ของข้อมูลทุกๆ nodeใน Working Detected(WD)ตามเวลาที่กำหนดและลบข้อมูลที่ time out แล้วทิ้ง
- บันทึกข้อมูลที่เกิดการเปลี่ยนสถานะต่างๆ ลงใน history file

ลำดับการทำงานของโปรแกรมย่อยนี้แสดงดังภาพที่ 3.18 และการทำงานดังนี้

1) ตรวจสอบจนกระทั่งข้อมูลใน Working Detect หมด หากยังไม่หมดทำข้อ 3) หากตรวจสอบทุกข้อมูลแล้วทำข้อ 2)
2) สิ้นสุดการทำงานของโปรแกรมย่อยในส่วนนี้
3) อ่านช่วงเวลาจาก Pattern file เพื่อทำการตรวจสอบต่อไปในขั้นตอนที่ 4)
4) ตรวจสอบเวลาดังต่อไปนี้
   - T_now – T_start >= Attack_period ทำข้อ 5)
   - T_now – T_start < Attack_period ทำการวนซ้ำโดยย้อนกลับไปทำข้อ 1)
5) Update Entry แล้วทำการวนซ้ำต่อไปโดยการย้อนกลับไปทำข้อ 1)



ภาพที่ 3.23 แสดงลำดับการทำงานการตรวจสอบ time out ของ Working Detect

บทที่ 4
การพัฒนาโปรแกรม ทดสอบ สรุปผล และข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการพัฒนาโปรแกรมซึ่งจะใช้ระบบปฏิบัติการ ภาษาและคอมไพเลอร์ และเครื่องมือต่างๆ ที่ใช้ในการพัฒนาโปรแกรมและกล่าวถึงการทดสอบการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชที่ได้ออกแบบและพัฒนาโปรแกรมไปแล้ว โดยจะมีการสร้างสถานการณ์จำลองส่งการร้องขอออบเจ็กต์ที่ทำให้เกิดการโจมตีขึ้นในลักษณะต่างๆเพื่อให้ครอบคลุมการทำงานของโปรแกรม และแสดงผลลัพธ์ที่ได้จากการทดสอบดังต่อไปนี้

4.1 การพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

การพัฒนาโปรแกรมจำเป็นต้องใช้เครื่องมือและภาษาในการพัฒนาซึ่งจะกล่าวถึงดังต่อไปนี้

4.1.1 ระบบปฏิบัติการที่ใช้ในการพัฒนาระบบ

ระบบปฏิบัติการที่ใช้ในการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชคือฟรีบีเอสดี สาเหตุที่เลือกใช้ระบบปฏิบัติการฟรีบีเอสดี เนื่องจากระบบปฏิบัติการฟรีบีเอสดีเป็นระบบปฏิบัติการยูนิกซ์ฟรีที่น่าสนใจรวมทั้งยังมีความสามารถสูงที่จะนำมาทำเป็น Server ทำให้มีความต้องการศึกษาการใช้งานบนระบบฟรีบีเอสดี

4.1.2 ภาษาและคอมไพเลอร์ที่ใช้ในการพัฒนาระบบ

ภาษาที่ใช้ในการพัฒนาโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชคือภาษาซีและใช้ GNU C compiler บนระบบปฏิบัติการฟรีบีเอสดี เนื่องจากภาษาซีเป็นภาษาระดับสูงซึ่งไม่ขึ้นกับฮาร์ดแวร์ แต่สามารถทำงานได้รวดเร็วเหมือนภาษาระดับต่ำ ดังนั้นจึงเลือกใช้ภาษาซีในการพัฒนาโปรแกรมนี้

4.1.3 เครื่องมือที่ใช้ในการพัฒนาระบบ

1) Editplus และ ee editor ใช้ในการเขียนโปรแกรม
2) Microsoft Word ใช้ในการจัดทำเอกสารประกอบโครงงาน
3) Visio และ Photoshop ใช้ในการวาดภาพประกอบ เช่นโฟลว์ชาร์ท (Flow Chart) และรูปภาพ

4.2 การทดสอบโปรแกรม

ในส่วนต่อไปจะกล่าวถึงทดสอบการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชที่ได้ออกแบบและพัฒนาโปรแกรมไปแล้ว โดยจะมีการสร้างสถานการณ์จำลองส่งการร้องขอออบเจ็กต์ที่ทำให้เกิดการโจมตีขึ้นในลักษณะต่างๆเพื่อให้ครอบคลุมการทำงานของโปรแกรมซึ่งก่อนจะทำการทดสอบต้องมีการเตรียมระบบเพื่อทดสอบโดยต้องติดตั้งโปรแกรมที่เกี่ยวข้องกับโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชบนเครื่อง server ดังภาคผนวก ก ซึ่งจะบอกถึงการติดตั้งและการใช้งาน จากนั้นจึงจะสมารถทำการทำสอบแต่ก่อนจะไปถงการทดสอบจะอธิบายรูปของผลการทำงานดังนี้

4.2.1 รูปแบบผลการทำงานของโปรแกรม

รูปแบบผลการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชแสดงดังภาพที่ 4.1 และจะแสดงความหมายของแต่ละคอลัมน์ในตารางที่ 4.1

Time working	status	Node ID	counter	IP Address	URL	method	time

ภาพที่ 4.1 แสดงผลการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

ตารางที่ 4.1 แสดงความหมายของผลการทำงานในแต่ละคอลัมน์

ชื่อคอลัมน์	ความหมาย
Time working	เวลาขณะที่โปรแกรมทำงาน
Status	สถานะของการตรวจจับการถูกโจมตีได้แก่ Detected, Attacked, Release รวมทั้งเหตุการณ์ที่เกิดขึ้นเช่น add now node , time out
Node ID	หมายเลขของ node
Counter	จำนวนนับของแต่ละ node
IP Address	หมายเลข IP ของแต่ละ node
URL	URL ของแต่ละ node
Method	method ของแต่ละ node
Time	เวลาที่เกิดการร้องขอออบเจ็กต์

4.2.2 การทดสอบโปรแกรม

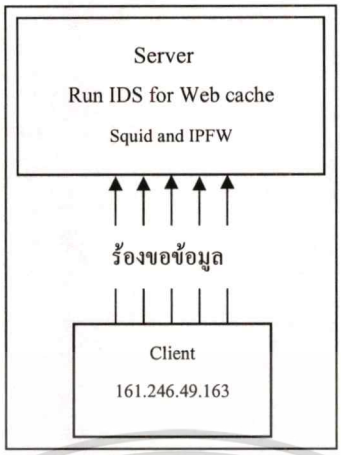
ในหัวข้อนี้เป็นการทดสอบโปรแกรมและแสดงผลการทดสอบซึ่งได้ทดสอบไว้ 2 กรณี ดังต่อไปนี้

4.2.2.1 การทดสอบที่ 1

เป็นการทดสอบเพื่อตรวจสอบการโจมตีจากเครื่อง client เพียงเครื่องเดียว

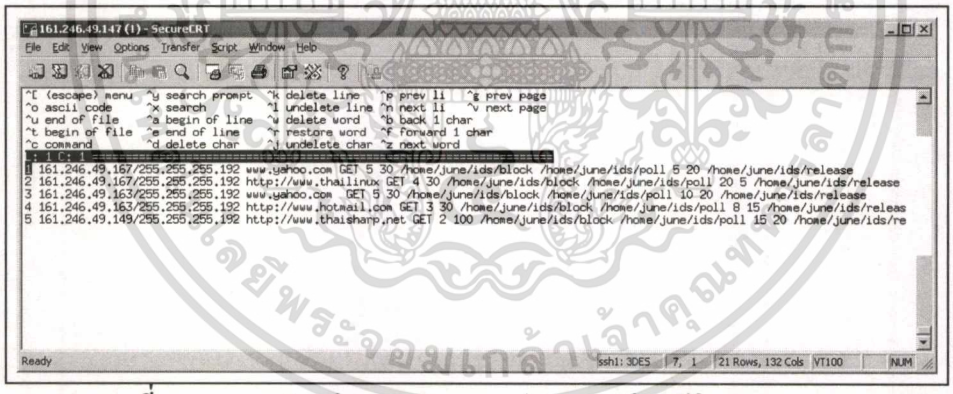
- ลักษณะการทดสอบที่ 1

การทดสอบที่ 1 แสดงการจำลองการโจมตีที่เกิดจากเครื่องเป้าหมายเพียงเครื่องเดียวซึ่งมีหมายเลข IP Address คือ 161.246.49.167 ทำการโจมตีโดยส่งการร้องขอไปยังเครื่อง server หมายเลข IP Address เป็น 161.246.49.147 ซึ่งมีการติดตั้งโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช และโปรแกรมอื่นๆที่เกี่ยวข้องไว้แล้ว ลักษณะการโจมตีแสดงได้ดังภาพที่ 4.2



ภาพที่ 4.2 จะแสดงลักษณะการทดสอบที่ 1

เมื่อสั่งให้โปรแกรมตรวจจับการโจมตีทำงานและเกิดการร้องขอข้อมูลจากเครื่อง Client ขึ้น โปรแกรมจะตรวจสอบการร้องขอดังกล่าวโดยพิจารณาระหว่างการร้องขอข้อมูลจาก access.log ของโปรแกรม Squid กับรูปแบบจากการกำหนดของผู้ใช้ใน Pattern file ซึ่งแสดงไว้ในภาพที่ 4.3 ดังนี้



ภาพที่ 4.3 แสดงตัวอย่างการกำหนดรูปแบบการโจมตีใน Pattern.conf

- ผลการทดสอบที่ 1

เมื่อทำการทดสอบโดยส่งการร้องขอข้อมูลจากเครื่อง Client ไปยังเครื่อง Server ภาพที่ 4.4 แสดงข้อมูลการร้องขอที่เกิดขึ้นใน access.log ส่วนผลการทำงานของโปรแกรมแสดงในภาพที่ 4.5 ซึ่งการอธิบายการทำงานของโปรแกรมต้องพิจารณาข้อมูลจากภาพทั้งสองประกอบกันเพื่อให้เห็นการทำงานอย่างชัดเจน

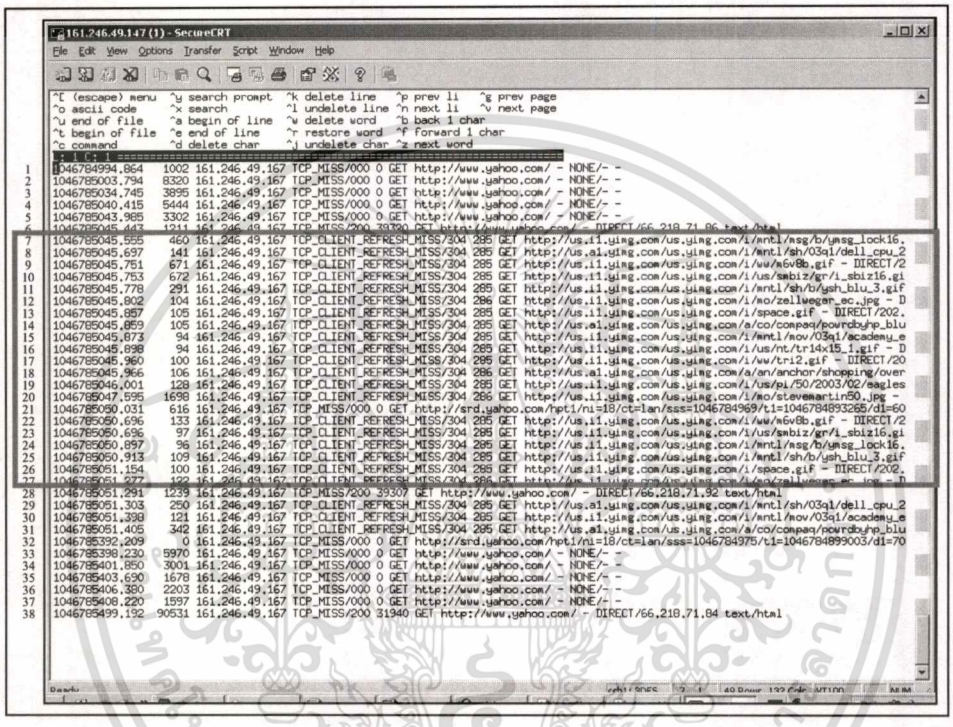
การทำงานของโปรแกรมสำหรับการทดสอบที่ 1 สามารถอธิบายเป็นกรณีย่อยๆได้ดังนี้

1) การร้องขอที่ไม่เป็นการโจมตีแต่มีรูปแบบตรงกับรูปแบบที่กำหนด

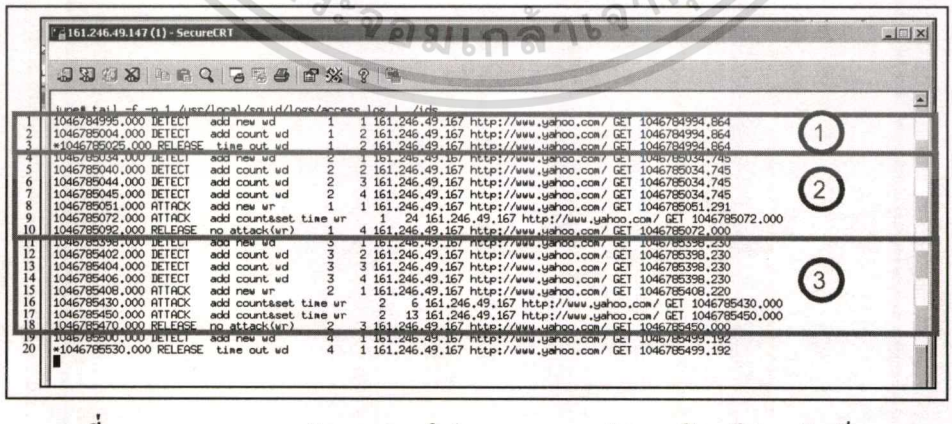
2) การร้องขอที่ตรวจพบว่าเป็นการโจมตี

3) การร้องขอที่ตรวจพบว่าเป็นการโจมตีอย่างต่อเนื่อง

4) การร้องขอที่ไม่เป็นการโจมตีมีรูปแบบไม่ตรงตามรูปแบบที่กำหนด



ภาพที่ 4.4 แสดงข้อมูลการร้องขอออบเจ็กต์ใน access.log



ภาพที่ 4.5 แสดงผลการทำงานของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช

การทำงานในแต่ละกรณีอธิบายได้ดังต่อไปนี้

<u>กรณีที่ 1</u> การร้องขอที่ไม่เป็นการโจมตีแต่มีรูปแบบตรงกับรูปแบบที่กำหนด

กรณีที่ 1 เกิดการร้องขอข้อมูลขึ้นจาก Client แต่การร้องขอดังกล่าวมีข้อมูลไม่ตรงตามข้อกำหนดการโจมตี การทำงานแสดงในภาพที่ 4.5 กรอบหมายเลข 1

- การทำงานในแถวที่ 1 สามารถอธิบายได้ดังนี้
  - เริ่มต้นเมื่อมีข้อมูลการร้องขอที่ได้รับมาจาก access.log เข้ามายังโปรแกรม สังเกตได้จากภาพที่ 4.4 ในแถวที่ 1 พบว่าเป็นการร้องขอข้อมูลที่เกิดขึ้นใน access.log ณ เวลา 1046784994.864
  - โปรแกรมจะตรวจสอบข้อมูลที่รับเข้ามากับรูปแบบที่ได้กำหนดไว้ใน Pattern file ซึ่งพบว่าสามารถจับคู่ได้กับรูปแบบที่ได้กำหนดไว้สังเกตได้จากภาพที่ 4.3 แถวที่ 1 โดยตารางที่ 4.2 แสดงการเปรียบเทียบรูปแบบการร้องขอที่เข้าคู่กันกับรูปแบบใน Pattern file

ตารางที่ 4.2 แสดงการเปรียบเทียบรูปแบบที่เข้าคู่กันของการทดสอบที่ 1

ข้อมูล	รูปแบบใน Pattern file	รูปแบบการร้องขอออบเจ็กต์
IP Address	161.246.49.167/255.255.255.192	161.246.49.167
URL	www.yahoo.com	http://www.yahoo.com
Method	GET	GET

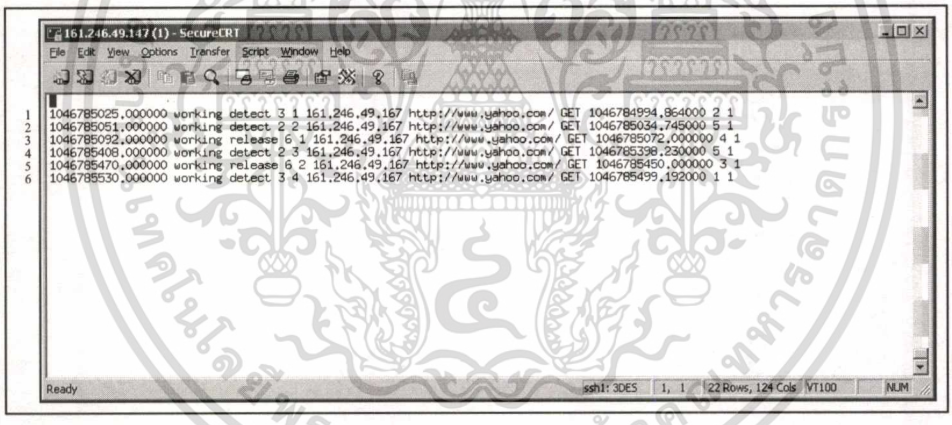
  - เมื่อพบว่าเข้าคู่กันจะเพิ่มโหนดการร้องขอนี้ลงใน working detect เพื่อใช้พิจารณาการร้องขอที่จะเข้ามาต่อไป โดยจากภาพที่ 4.5 เป็นภาพการทำงานของโปรแกรมแสดงการเพิ่มโหนดใหม่ที่เกิดขึ้นจากการเข้าคู่กันของการร้องขอสังเกตได้จากแถวที่ 1 กำหนดค่าหมายเลขโหนดของ working detect เป็น 1 และจำนวนนับเป็น 1 แสดงสถานะเป็น DETECT พร้อมทั้งแสดงหมายเลข IP Address, URL, Method และเวลาของการร้องขอ
- การทำงานในแถวที่ 2 สามารถอธิบายได้ดังนี้
  - เมื่อมีการร้องขอข้อมูลเข้ามาอีกครั้ง ดังภาพที่ 4.4 แถวที่ 2 ซึ่งเป็นข้อมูลการร้องขอออบเจ็กต์ที่เกิดขึ้น Squid ณ เวลา 1046785003.794
  - โปรแกรมทำงานตรวจสอบแล้วพบว่ามีรูปแบบที่เข้าคู่กันกับรูปแบบใน Pattern file และตรงตามโหนดใน working detect ที่เคยพบแล้วจึงเพิ่มจำนวนนับแล้วพิจารณาการโจมตีคือ พิจารณาจำนวนนับซึ่งพบว่าน้อยกว่าที่ได้กำหนดไว้ใน Pattern file และ พิจารณาช่วงเวลาโดยพิจารณาจาก เวลา

ปัจจุบัน(1046785004.000) – เวลาที่ทำการร้องขอที่ตรวจพบเป็นครั้งแรก (1046784994.864) ได้ผลเป็น 9.136 น้อยกว่าเวลาที่กำหนดคือ 30 ดังนั้นจึงเพิ่มจำนวนนับให้อีก 1 แสดงดังภาพที่ 4.5 สังเกตจะพบว่ามีการเพิ่มจำนวนนับในโหนดที่ 1

- การทำงานในแถวที่ 3 สามารถอธิบายได้ดังนี้

- เมื่อโปรแกรมไม่ได้รับข้อมูลการร้องขอจาก access.log จะทำการตรวจสอบข้อมูลใน working detect เพื่อจัดการลบข้อมูลที่ไม่ต้องการใช้แล้วออกไป ซึ่งในที่นี้โปรแกรมตรวจพบโหนดที่1 มีระยะเวลาเกินเวลาที่กำหนด คือ เวลาปัจจุบัน(1046785025.000) – เวลาเริ่มต้นการตรวจจับแต่ล่ะโหนด(1046784994.864) มีค่าเป็น 30.136 ซึ่งมากว่า 30 ที่กำหนดไว้ โปรแกรมจึงลบโหนดนี้ออกจากการทำงานของโปรแกรมเพื่อไม่ให้เปลืองทรัพยากรของระบบ

เมื่อโปรแกรมทำการเปลี่ยนแปลงเช่นการลบโหนดดังกล่าวจะบันทึกการเปลี่ยนแปลงลงใน history file ดังภาพที่ ดังภาพที่ 4.6 แถวที่ 1

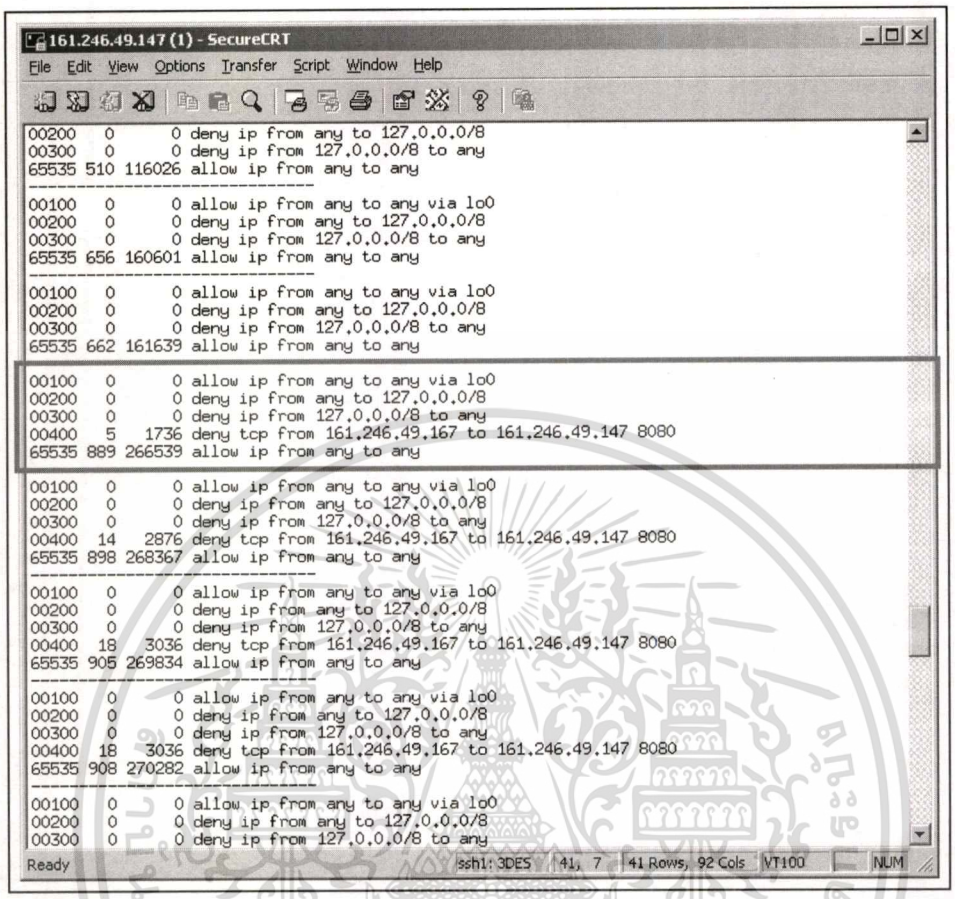


ภาพที่ 4.6 แสดงข้อมูลที่บันทึกลงใน history.log

<u>กรณีที่ 2</u> การร้องขอที่ตรวจพบว่าเป็นการโจมตี

เป็นกรณีซึ่งโปรแกรมตรวจพบการโจมตีและส่งคำสั่งไปยังไฟร์วอลล์เพื่อปฏิเสธการร้องขอข้อมูล หลังจากนั้นโปรแกรมจะตรวจสอบการโจมตีจนกระทั่งไม่พบการโจมตีแล้วจึงส่งคำสั่งไปยังไฟร์วอลล์เพื่อยกเลิกปฏิเสธการร้องขอ ผลการทำงานของโปรแกรมแสดงให้เห็นในภาพที่ 4.5 การอบที่ 2 และสามารถอธิบายได้ดังต่อไปนี้

- การทำงานในแถวที่ 4 สามารถอธิบายได้ดังนี้
  - เมื่อได้รับข้อมูลการร้องขอที่ได้รับมาจาก access.log เข้ามายังโปรแกรม สังเกตได้จากภาพที่ 4.4 ในแถวที่ 3 ซึ่งเป็นการร้องขอข้อมูลที่เกิดขึ้นใน access.log ณ เวลา 1046785034.745
  - เช่นเดียวกันกับกรณีที่ 1 โปรแกรมจะตรวจสอบข้อมูลที่รับเข้ามากับรูปแบบที่ได้กำหนดไว้ใน Pattern file ซึ่งพบว่าสามารถจับคู่ได้กับรูปแบบที่ได้กำหนด
  - เมื่อพบว่าเข้าคู่กันจะเพิ่มโหนดการร้องขอนี้ลงใน working detect เพื่อใช้พิจารณาการร้องขอที่จะเข้ามาต่อไป โดยจากภาพที่ 4.5 แถวที่ 4 เพิ่มโหนดใหม่เป็นโหนดที่ 2 ของ working detect และมีจำนวนนับเป็น 1 แสดงสถานะเป็น DETECT พร้อมทั้งแสดงหมายเลข IP Address, URL, Method และเวลาของการร้องขอ
- การทำงานในแถวที่ 5-7 สามารถอธิบายได้ดังนี้
  - ข้อมูลการร้องขอที่ได้รับมาจาก access.log เข้ามายังโปรแกรม สามารถสังเกตได้จากภาพที่ 4.4 ในแถวที่ 4-6 เมื่อผ่านการทำงานของโปรแกรมแล้วพบว่าจำนวนนับยังน้อยกว่าที่กำหนด(ซึ่งจำนวนที่กำหนดไว้คือ 5) และพิจารณาแล้วพบว่าช่วงเวลายังไม่เกินเวลาที่กำหนดไว้เช่น การทำงานในแถวที่ 7 พิจารณาเวลาปัจจุบัน(1046785045.000)กับเวลาเริ่มต้นการพิจารณา(1046785034.745) มีผลต่างเป็น 10.255 ซึ่งน้อยกว่า 30 ที่กำหนดไว้ จึงมีการเพิ่มจำนวนนับจาก เดิม 3 เป็น 4 เป็นต้น แสดงผลการทำงานในภาพที่ 4.5 แถวที่ 5-7
- การทำงานในแถวที่ 8 สามารถอธิบายได้ดังนี้
  - ต่อมาเมื่อได้รับข้อมูลในภาพที่ 4.4 แถวที่ 28
  - โปรแกรมจะตรวจสอบพบว่ามีจำนวนการร้องขอรวมทั้งการร้องขอที่รับเข้ามาใหม่เป็นจำนวนทั้งหมด 5 ซึ่งตรงตามรูปแบบที่กำหนดไว้ว่าเป็นการโจมตี
  - เมื่อโปรแกรมพบการโจมตีจะส่งคำสั่งไปยังไฟร์วอลล์เพื่อให้ปฏิเสธการร้องขอข้อมูลจากเครื่องเป้าหมายทำให้สามารถตรวจสอบการทำงานของไฟร์วอลล์ได้ดังภาพที่ 4.7 ซึ่งในกรอบเป็นการปฏิเสธการรับข้อมูลจากเครื่องหมายเลข IP Address 161.246.49.167 ที่จะส่งไปยังเครื่อง Server หมายเลข IP Address 161.246.49.147 หมายเลข port 8080



ภาพที่ 4.7 แสดงกฎของ ipfw ที่เพิ่มขึ้นขณะโปรแกรมทำงาน

- เมื่อมีการเปลี่ยนแปลงจะมีการบันทึกการเปลี่ยนแปลงของโหนดที่ 2 ของ working detect ลงใน history file ดังภาพที่ 4.6 แถวที่ 2
- นอกจากนี้จะมีการสร้างโหนดใหม่ซึ่งใช้สำหรับพิจารณาการหยุดโจมตีของเครื่องเป้าหมายโดยเพิ่มโหนดใหม่ใน working release เป็นโหนดที่ 2 ดังภาพที่ 4.5 แถวที่ 8

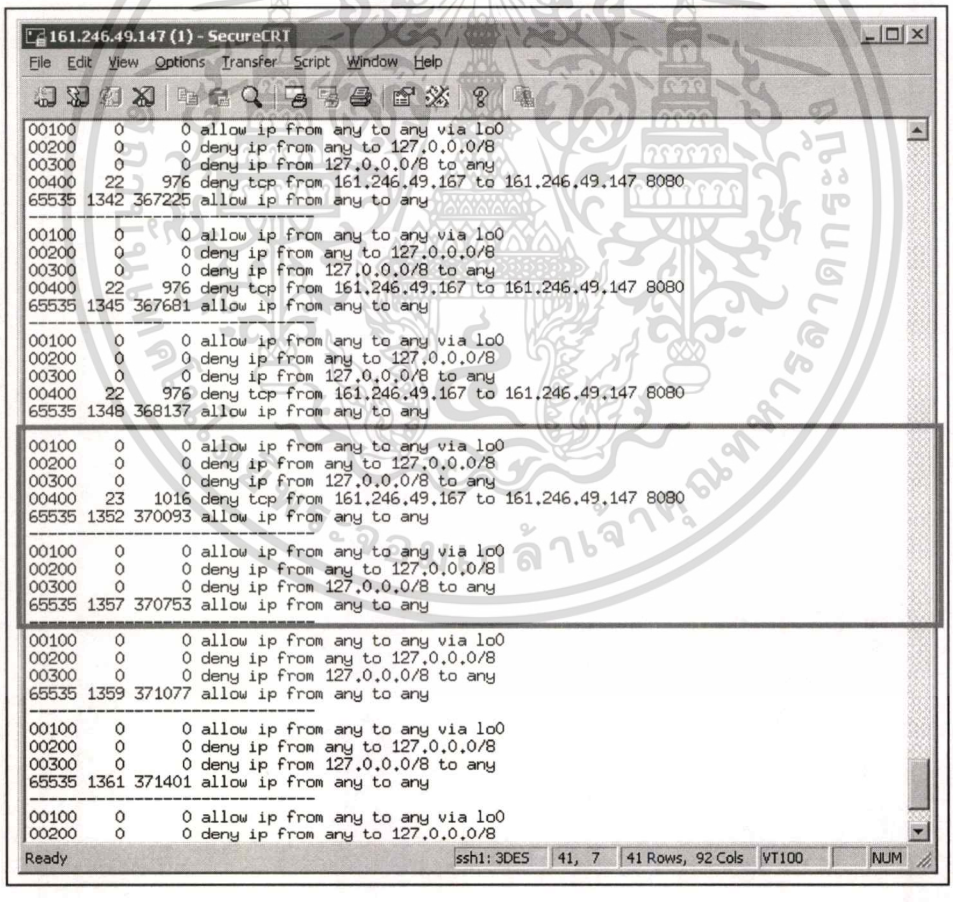
❑ การทำงานในแถวที่ 9 สามารถอธิบายได้ดังนี้

- โปรแกรมจะตรวจสอบการโจมตีเมื่อไม่ได้รับข้อมูลการร้องขอจาก access.log โดยพิจารณาการสิ้นสุดการโจมตีของเครื่อง Client เป้าหมาย ใช้เงื่อนไขของเวลาและจำนวนครั้งคือ พิจารณาตรวจสอบการสิ้นสุดการโจมตีตามที่ได้มีการกำหนดไว้ใน Pattern file หากยังไม่ครบกำหนดจะไม่ทำการส่งคำสั่งไปถามที่ไฟร์วอลล์ ในที่นี้จะพบว่าเวลาปัจจุบัน(1046785072.000)มีผลต่างจากเวลาเริ่มต้น (1046785051.291) ในการพิจารณาเป็น 20.709 มากกว่า 20 ซึ่งเกินช่วงเวลาที่ได้กำหนด

แล้วจะส่งคำสั่งถามจำนวนครั้งที่ Client ของการร้องขอข้อมูลเป็น packet มายัง Server แล้วนำมาพิจารณาพบว่ามีจำนวนครั้งการร้องขอ 24 ซึ่งมากกว่า 5 จึงไม่ถือว่าเป็นการสิ้นสุดการโจมตี จึงเพิ่มจำนวนนับดังกล่าว พร้อมทั้งปรับปรุงเวลาของการพิจารณา ดังภาพที่ 4.5 แถวที่ 9

- การทำงานในแถวที่ 10 สามารถอธิบายได้ดังนี้

- มีการตรวจสอบการสิ้นสุดการโจมตีอีกครั้งและพบว่าจำนวนของ packet ที่ตรวจพบ น้อยกว่าที่กำหนดคือ 4 น้อยกว่า 5 โปรแกรมจึงส่งคำสั่งเพื่อยกเลิกการปฏิเสธการโจมตีไปยังไฟร์วอลล์ ผลการทำงานของโปรแกรมเป็นดังภาพที่ 4.5 แถวที่ 10 ส่วนผลการทำงานของไฟร์วอลล์เป็นดังภาพที่ 4.8 ซึ่งจากภาพที่ 4.8 จะเห็นว่าในกรอบสี่เหลี่ยมจะแสดงกฎที่ลดลงจากเดิมเมื่อโปแกรมตรวจสอบไม่พบการโจมตีจึงส่งคำสั่งไปยังไฟร์วอลล์เพื่อยกเลิกการปฏิเสธการร้องขอข้อมูล
- เมื่อเกิดการเปลี่ยนแปลงสถานะจะมีการบันทึกข้อมูลลงใน history file แสดงดังภาพที่ 4.6 แถวที่ 3



ภาพที่ 4.8 แสดงกฎของ ipfw ที่ลดลงขณะโปรแกรมทำงาน

<u>กรณีที่ 3</u> การร้องขอที่ตรวจพบว่าเป็นการโจมตีอย่างต่อเนื่อง

ลักษณะการโจมตีเป็นลักษณะเดียวกับกรณีที่ 3 แต่จะเน้นให้เป็นว่าหากยังพบการโจมตีอยู่ โปรแกรมจะยังเพิ่มจำนวนนับจนกว่าจะไม่พบการโจมตีแล้วจึงจะส่งคำสั่งเพื่อยกเลิกการปฏิเสธการโจมตีดังภาพที่ 4.5 ในกรอบที่ 3

- การทำงานในแถวที่ 11-14 สามารถอธิบายได้ดังนี้
  - เป็นการตรวจพบการเข้าคู่ของรูปแบบจึงเพิ่มโหนดใหม่ลงใน working detect เป็นโหนดที่ 3
  - ต่อมาในแถวที่ 12-14 ตรวจพบการร้องขอที่ซ้ำกับรูปแบบที่เคยตรวจพบซึ่งตรงกับโหนดที่ 3 และยังไม่ครบตามเงื่อนไขของการโจมตีจึงเพิ่มจำนวนนับขึ้นครั้งละ 1
- การทำงานในแถวที่ 15 สามารถอธิบายได้ดังนี้
  - ตรวจพบการร้องขอที่ตรงตามรูปแบบในโหนดที่ 3 และพบว่ามีจำนวนเท่ากับจำนวนที่กำหนดให้เป็นการโจมตีดังนั้นจึงเพิ่มโหนดใน working release และปรับปรุงข้อมูลโดยการลบและคัดลองข้อมูลที่เกิดการเปลี่ยนแปลงนี้ลงใน history file ดังภาพที่ 4.6 แถวที่ 4
- การทำงานในแถวที่ 16, 17 สามารถอธิบายได้ดังนี้
  - โปรแกรมทำการตรวจสอบจำนวนการร้องขอข้อมูลที่เกิดขึ้นที่ไฟร์วอลล์แต่ยังคงมีจำนวนมากกว่าที่กำหนดไว้จึงทำการปรับปรุงจำนวนและเวลา
- การทำงานในแถวที่ 18 สามารถอธิบายได้ดังนี้
  - โปรแกรมตรวจไม่พบการโจมตีจึงส่งคำสั่งไปยังไฟร์วอลล์เพื่อยกเลิกการปฏิเสธการร้องขอและปรับปรุงข้อมูลพร้อมทั้งคัดลอกข้อมูลลงใน history file ดังภาพที่ 4.6 แถวที่ 5

<u>กรณีที่ 4</u> การร้องขอที่ไม่เป็นการโจมตีมีรูปแบบไม่ตรงตามรูปแบบที่กำหนด

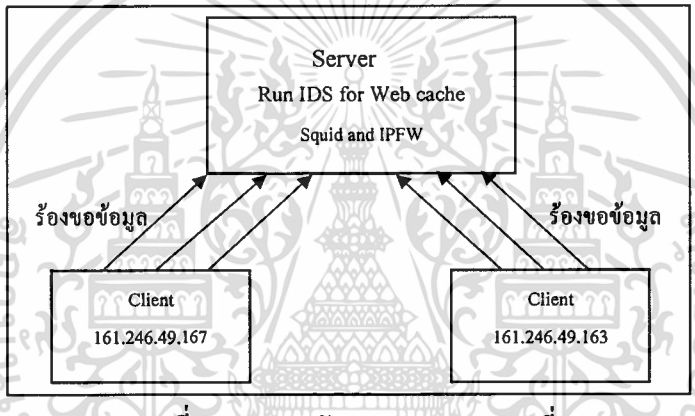
กรณีนี้จะแสดงให้เห็นการทำงานของโปรแกรมในการรับข้อมูลการร้องขอซึ่งไม่ตรงกับรูปแบบที่ได้กำหนดไว้ใน Pattern file โปรแกรมจะไม่สนใจข้อมูลดังกล่าว ตัวอย่างข้อมูลที่ไม่ตรงตามรูปแบบที่กำหนดแสดงในกรอบสี่เหลี่ยมดังภาพที่ 4.4 เมื่อเปรียบเทียบกับ Pattern file ในภาพที่ 4.3 จะพบว่าการร้องขอข้อมูลในกรอบสี่เหลี่ยมในภาพที่ 4.4 จะไม่ตรงกับแถวใดใน Pattern file เลย จากผลการทำงานของโปรแกรมจะไม่ปรากฏข้อมูลที่ไม่ตรงตามรูปแบบที่ได้กำหนดไว้

4.2.2.2 การทดสอบที่ 2

เป็นการทดสอบเพื่อตรวจสอบการโจมตีจากเครื่อง Client 2 เครื่องโดยทั้งสองเครื่องจะส่งการร้องร้องขอไปที่ server

- ลักษณะการทดสอบที่ 2

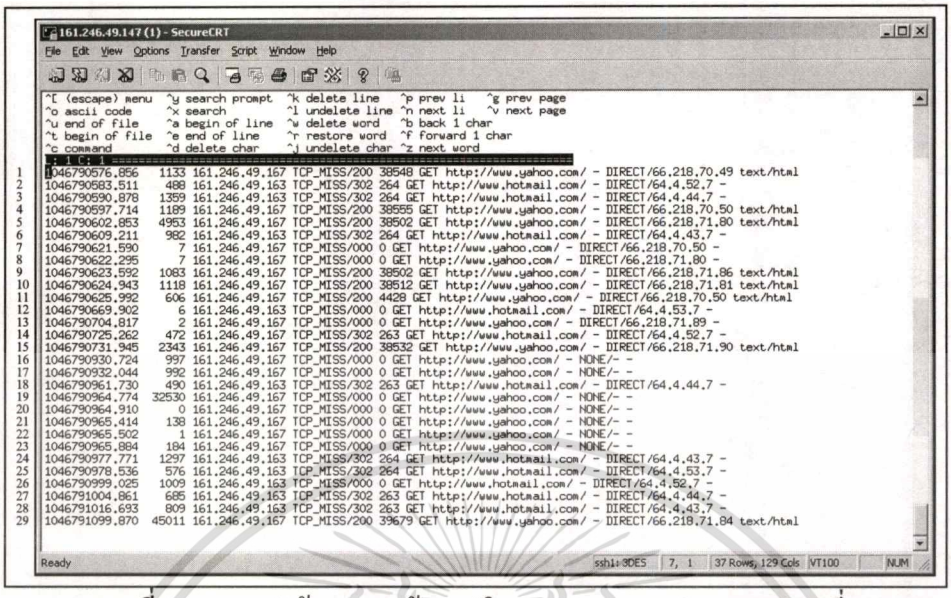
การทดสอบเพื่อตรวจสอบการโจมตีจากเครื่อง Client หลายเครื่อง ในการทดสอบนี้จะใช้จำนวนเครื่อง client 2 เครื่องคือเครื่องที่มีหมายเลข IP 161.246.49.163 และ 161.246.49.167 เพื่อทดสอบตามรูปแบบของ Pattern file ภาพที่ 4.9 แสดงลักษณะการทดสอบที่ 2 ดังนี้



ภาพที่ 4.9 แสดงลักษณะการทดสอบที่ 2

- ผลการทดสอบที่ 2

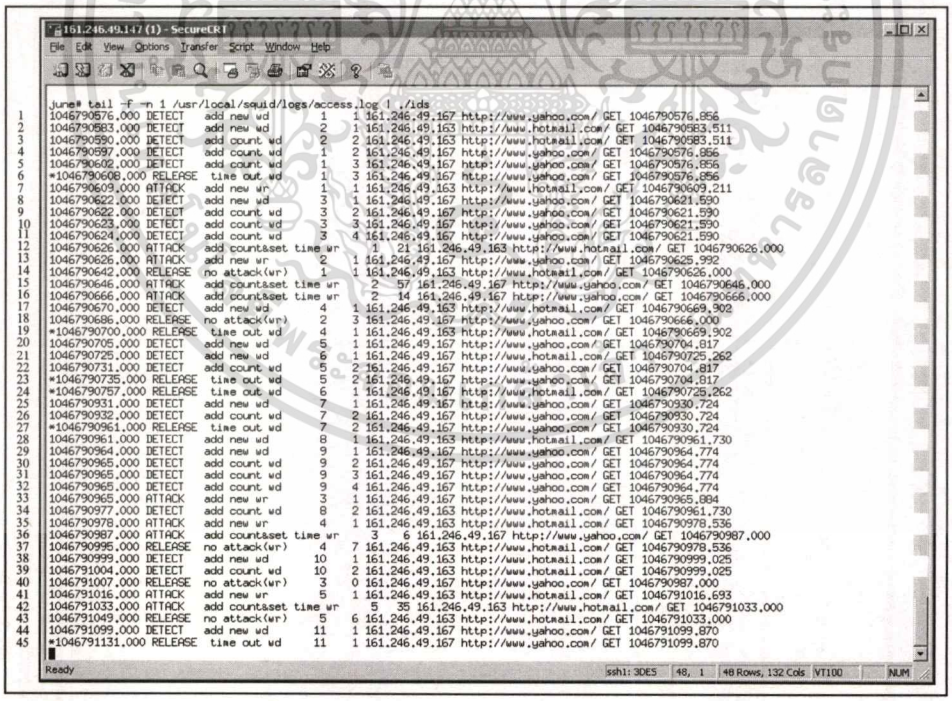
จากการทดสอบที่ 1 ที่ได้อธิบายไปแล้วนั้นเป็นการอธิบายโดยละเอียดเพื่อความเข้าใจการทำงานของโปรแกรมดังนี้ในการทดสอบที่ 2 นี้จะอธิบายถึงการทำงานเพื่อให้เห็นถึงการทำงานของโปรแกรมในลักษณะกระชับขึ้น โดยจะแสดงภาพการร้องขอข้อมูลซึ่งได้ตัดตอนเอาเฉพาะส่วนที่สำคัญ เนื่องจากข้อมูลการร้องขอมีจำนวนมากซึ่งมีทั้งส่วนที่ตรงตามรูปแบบที่ต้องการตรวจสอบและไม่ตรงตามรูปแบบดังนั้นเพื่อให้แสดงการร้องขอที่เกี่ยวข้องได้ครบถ้วนและกระชับจึงตัดตอนได้ดังภาพที่ 4.10 และภาพที่ 4.11 จะเป็น ภาพแสดงผลการทำงานของโปรแกรมหลังจากได้ทดสอบโปรแกรมไปแล้ว



```
161.246.49.147 (1) - SecureCRT
File  Edit  View  Options  Transfer  Script  Window  Help

^[ (escape) menu    ^y search prompt  ^k delete line   ^p prev li   ^g prev page
^o ascii code       ^x search         ^l undelete line ^n next li   ^v next page
^u end of file      ^a begin of line  ^w delete word   ^b back 1 char
^t begin of file    ^e end of line    ^r restore word  ^f forward 1 char
^c command          ^d delete char    ^j undelete char ^z next word
[root@]#
1  1046790576.856    1133 161.246.49.167 TCP_MISS/200 38548 GET http://www.yahoo.com/ - DIRECT/66.218.70.49 text/html
2  1046790583.511     488 161.246.49.163 TCP_MISS/302 264 GET http://www.hotmail.com/ - DIRECT/64.4.52.7 -
3  1046790590.878    1359 161.246.49.163 TCP_MISS/302 264 GET http://www.hotmail.com/ - DIRECT/64.4.44.7 -
4  1046790597.714    1189 161.246.49.167 TCP_MISS/200 38555 GET http://www.yahoo.com/ - DIRECT/66.218.70.50 text/html
5  1046790602.853    4953 161.246.49.167 TCP_MISS/200 38502 GET http://www.yahoo.com/ - DIRECT/66.218.71.80 text/html
6  1046790609.211     982 161.246.49.163 TCP_MISS/302 264 GET http://www.hotmail.com/ - DIRECT/64.4.43.7 -
7  1046790621.590       7 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - DIRECT/66.218.70.50 -
8  1046790622.295       7 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - DIRECT/66.218.71.80 -
9  1046790623.592    1083 161.246.49.167 TCP_MISS/200 38502 GET http://www.yahoo.com/ - DIRECT/66.218.71.86 text/html
10 1046790624.943    1118 161.246.49.167 TCP_MISS/200 38512 GET http://www.yahoo.com/ - DIRECT/66.218.71.81 text/html
11 1046790625.992     606 161.246.49.167 TCP_MISS/200 4429 GET http://www.yahoo.com/ - DIRECT/66.218.70.50 text/html
12 1046790669.902       6 161.246.49.163 TCP_MISS/000 0 GET http://www.hotmail.com/ - DIRECT/64.4.53.7 -
13 1046790704.817       2 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - DIRECT/66.218.71.89 -
14 1046790725.262     472 161.246.49.167 TCP_MISS/302 263 GET http://www.hotmail.com/ - DIRECT/64.4.52.7 -
15 1046790731.945    2343 161.246.49.167 TCP_MISS/200 38532 GET http://www.yahoo.com/ - DIRECT/66.218.71.90 text/html
16 1046790930.724     997 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
17 1046790932.044     992 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
18 1046790961.730     490 161.246.49.163 TCP_MISS/302 263 GET http://www.hotmail.com/ - DIRECT/64.4.44.7 -
19 1046790964.774   32530 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
20 1046790964.910       0 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
21 1046790965.414     138 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
22 1046790965.502       1 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
23 1046790965.884     184 161.246.49.167 TCP_MISS/000 0 GET http://www.yahoo.com/ - NONE/- -
24 1046790977.771    1297 161.246.49.163 TCP_MISS/302 264 GET http://www.hotmail.com/ - DIRECT/64.4.43.7 -
25 1046790978.536     576 161.246.49.163 TCP_MISS/302 264 GET http://www.hotmail.com/ - DIRECT/64.4.53.7 -
26 1046790999.025    1009 161.246.49.163 TCP_MISS/000 0 GET http://www.hotmail.com/ - DIRECT/64.4.52.7 -
27 1046791004.861     685 161.246.49.163 TCP_MISS/302 263 GET http://www.hotmail.com/ - DIRECT/64.4.44.7 -
28 1046791016.693     809 161.246.49.163 TCP_MISS/302 263 GET http://www.hotmail.com/ - DIRECT/64.4.43.7 -
29 1046791099.870   45011 161.246.49.167 TCP_MISS/200 39679 GET http://www.yahoo.com/ - DIRECT/66.218.71.84 text/html
Ready                                                    ssh1: 3DES  7, 1  37 Rows, 129 Cols  VT100  NUM
```

ภาพที่ 4.10 แสดงข้อมูลการร้องขอใน access.log ของการทดสอบที่ 2



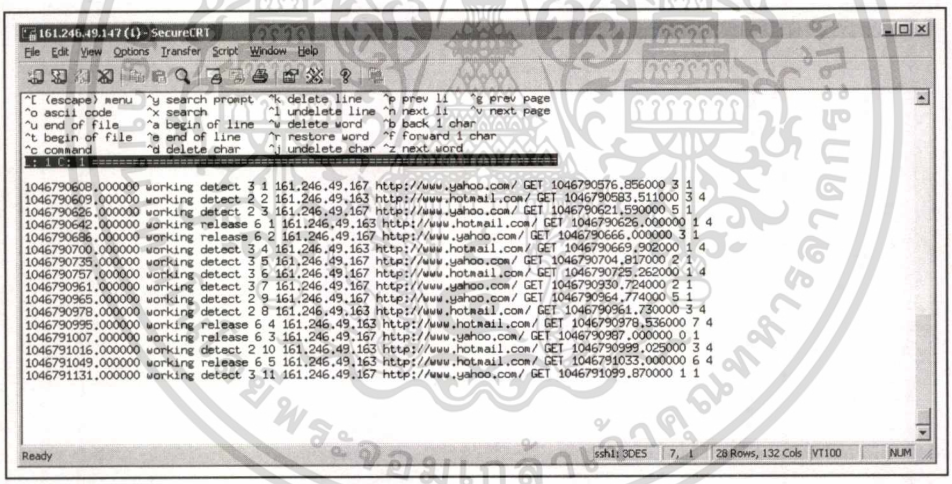
```
161.246.49.147 (1) - SecureCRT
File  Edit  View  Options  Transfer  Script  Window  Help

june# tail -f -n 1 /usr/local/squid/logs/access.log | ./ids
1  1046790576.000 DETECT   add new wd          1    1 161.246.49.167 http://www.yahoo.com/ GET 1046790576.856
2  1046790583.000 DETECT   add new wd          2    1 161.246.49.163 http://www.hotmail.com/ GET 1046790583.511
3  1046790590.000 DETECT   add count wd        2    2 161.246.49.163 http://www.hotmail.com/ GET 1046790583.511
4  1046790597.000 DETECT   add count wd        1    2 161.246.49.167 http://www.yahoo.com/ GET 1046790576.856
5  1046790602.000 DETECT   add count wd        1    3 161.246.49.167 http://www.yahoo.com/ GET 1046790576.856
6  *1046790608.000 RELEASE  time out wd        1    3 161.246.49.167 http://www.yahoo.com/ GET 1046790576.856
7  1046790609.000 ATTACK   add new wr          1    1 161.246.49.163 http://www.hotmail.com/ GET 1046790609.211
8  1046790622.000 DETECT   add new wd          3    1 161.246.49.167 http://www.yahoo.com/ GET 1046790621.590
9  1046790622.000 DETECT   add count wd        3    2 161.246.49.167 http://www.yahoo.com/ GET 1046790621.590
10 1046790623.000 DETECT   add count wd        3    3 161.246.49.167 http://www.yahoo.com/ GET 1046790621.590
11 1046790624.000 DETECT   add count wd        3    4 161.246.49.167 http://www.yahoo.com/ GET 1046790621.590
12 1046790626.000 ATTACK   add count&set time wr    1   21 161.246.49.163 http://www.hotmail.com/ GET 1046790626.000
13 1046790626.000 ATTACK   add new wr          2    1 161.246.49.167 http://www.yahoo.com/ GET 1046790625.992
14 1046790642.000 RELEASE  no attack(wr)       1    1 161.246.49.163 http://www.hotmail.com/ GET 1046790626.000
15 1046790646.000 ATTACK   add count&set time wr    2   57 161.246.49.167 http://www.yahoo.com/ GET 1046790646.000
16 1046790666.000 ATTACK   add count&set time wr    2   14 161.246.49.167 http://www.yahoo.com/ GET 1046790666.000
17 1046790670.000 DETECT   add new wd          4    1 161.246.49.163 http://www.hotmail.com/ GET 1046790669.902
18 1046790686.000 RELEASE  no attack(wr)       2    3 161.246.49.167 http://www.yahoo.com/ GET 1046790666.000
19 *1046790700.000 RELEASE  time out wd        4    1 161.246.49.163 http://www.hotmail.com/ GET 1046790669.902
20 1046790705.000 DETECT   add new wd          5    1 161.246.49.167 http://www.yahoo.com/ GET 1046790704.817
21 1046790725.000 DETECT   add new wd          6    1 161.246.49.167 http://www.hotmail.com/ GET 1046790725.262
22 1046790731.000 DETECT   add count wd        5    2 161.246.49.167 http://www.yahoo.com/ GET 1046790704.817
23 *1046790735.000 RELEASE  time out wd        5    2 161.246.49.167 http://www.yahoo.com/ GET 1046790704.817
24 *1046790757.000 RELEASE  time out wd        6    1 161.246.49.167 http://www.hotmail.com/ GET 1046790725.262
25 1046790931.000 DETECT   add new wd          7    1 161.246.49.167 http://www.yahoo.com/ GET 1046790930.724
26 1046790932.000 DETECT   add count wd        7    2 161.246.49.167 http://www.yahoo.com/ GET 1046790930.724
27 *1046790961.000 RELEASE  time out wd        7    2 161.246.49.167 http://www.yahoo.com/ GET 1046790930.724
28 1046790961.000 DETECT   add new wd          8    1 161.246.49.163 http://www.hotmail.com/ GET 1046790961.730
29 1046790964.000 DETECT   add new wd          9    1 161.246.49.167 http://www.yahoo.com/ GET 1046790964.774
30 1046790965.000 DETECT   add count wd        9    2 161.246.49.167 http://www.yahoo.com/ GET 1046790964.774
31 1046790965.000 DETECT   add count wd        9    3 161.246.49.167 http://www.yahoo.com/ GET 1046790964.774
32 1046790965.000 DETECT   add count wd        9    4 161.246.49.167 http://www.yahoo.com/ GET 1046790964.774
33 1046790965.000 ATTACK   add new wr          3    1 161.246.49.167 http://www.yahoo.com/ GET 1046790965.884
34 1046790977.000 DETECT   add count wd        8    2 161.246.49.163 http://www.hotmail.com/ GET 1046790961.730
35 1046790978.000 ATTACK   add new wr          4    1 161.246.49.163 http://www.hotmail.com/ GET 1046790978.536
36 1046790987.000 ATTACK   add count&set time wr    3    6 161.246.49.167 http://www.yahoo.com/ GET 1046790987.000
37 1046790995.000 RELEASE  no attack(wr)       4    7 161.246.49.163 http://www.hotmail.com/ GET 1046790978.536
38 1046790999.000 DETECT   add new wd          10   1 161.246.49.163 http://www.hotmail.com/ GET 1046790999.025
39 1046791004.000 DETECT   add count wd        10   2 161.246.49.163 http://www.hotmail.com/ GET 1046790999.025
40 1046791007.000 RELEASE  no attack(wr)       3    0 161.246.49.167 http://www.yahoo.com/ GET 1046790987.000
41 1046791016.000 ATTACK   add new wr          5    1 161.246.49.163 http://www.hotmail.com/ GET 1046791016.693
42 1046791033.000 ATTACK   add count&set time wr    5   35 161.246.49.163 http://www.hotmail.com/ GET 1046791033.000
43 1046791049.000 RELEASE  no attack(wr)       5    6 161.246.49.163 http://www.hotmail.com/ GET 1046791033.000
44 1046791099.000 DETECT   add new wd          11   1 161.246.49.167 http://www.yahoo.com/ GET 1046791099.870
45 *1046791131.000 RELEASE  time out wd        11   1 161.246.49.167 http://www.yahoo.com/ GET 1046791099.870
Ready                                                    ssh1: 3DES  48, 1  48 Rows, 132 Cols  VT100  NUM
```

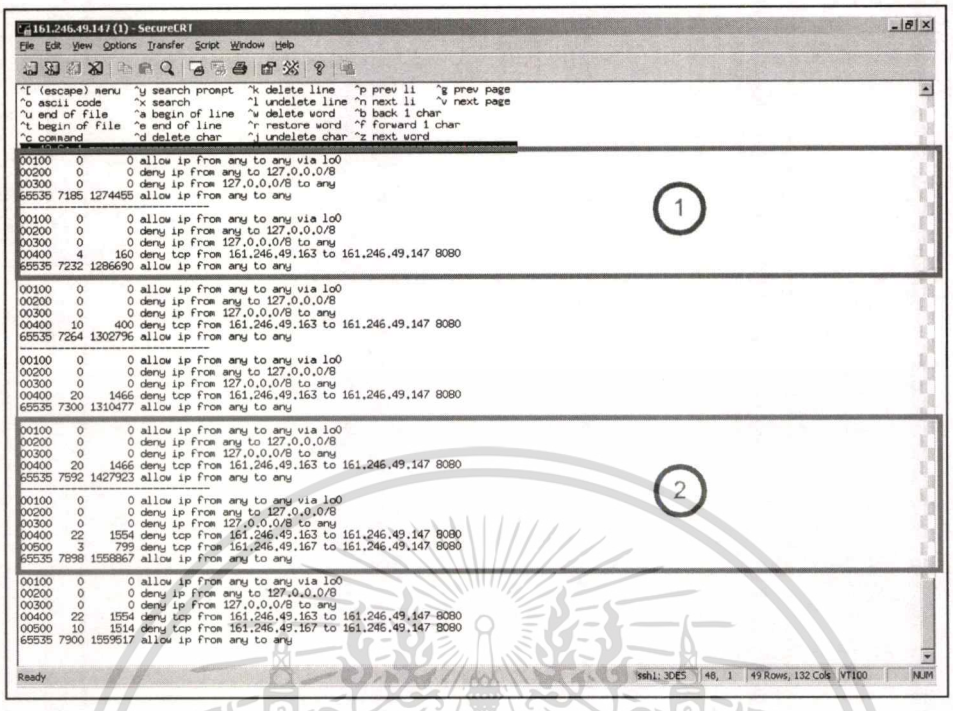
ภาพที่ 4.11 แสดงผลการทำงานการทดสอบที่ 2

จากภาพทั้งภาพที่ 4.13 และภาพที่ 4.14 สามารถสรุปการทำงานได้ดังต่อไปนี้

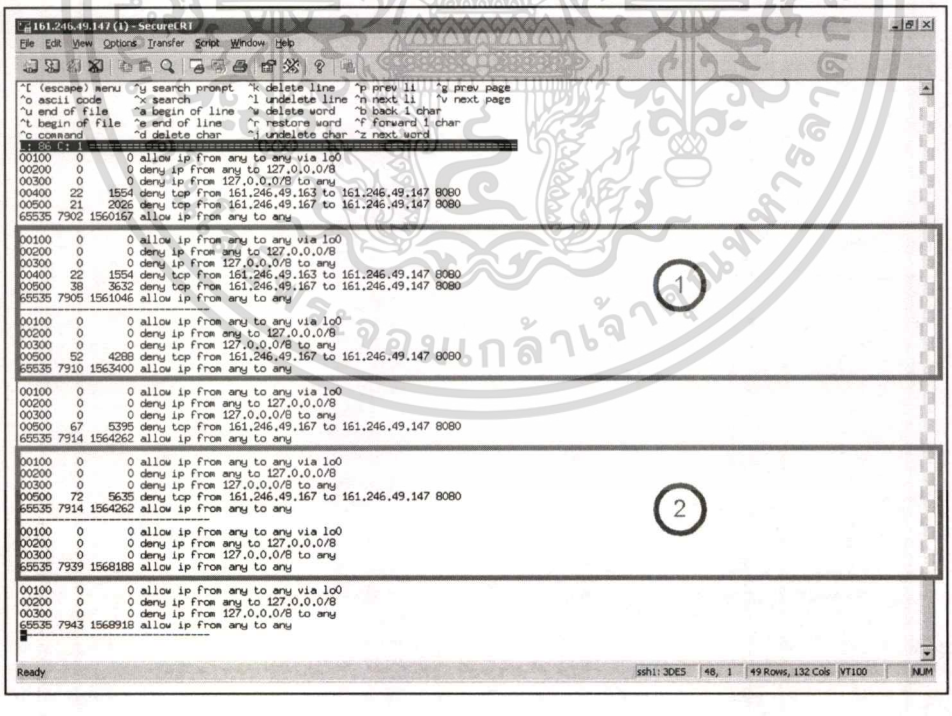
- ทั้งสองเครื่องสมารถเรียกใช้งานเครื่อง Server ได้โดยหากมีการตรวจพบรูปแบบการร้องขอที่เข้าคู่กับรูปแบบใน Pattern file จะเพิ่มโหนดใหม่ใน working detect
- เมื่อเครื่องหนึ่งเครื่องใดถูกตรวจพบว่ามีการร้องขอมากเกินไปจะถือว่าเป็นการโจมตีตามการกำหนดของรูปแบบการโจมตีจะถูกไฟร์วอลล์จำกัดการร้องขอข้อมูลซึ่งจะไม่กระทบถึงเครื่องอีกเครื่องหนึ่งจากภาพที่ 4.14 แถวที่ 7 เครื่อง IP 161.246.49.163 มีการร้องขอมากเกินกำหนดจึงถูกปฏิเสธการร้องขอแต่ทำให้ไม่สามารถร้องขอข้อมูลได้แต่ไม่กระทบถึงเครื่อง IP 161.246.49.167
- หากเกิดการโจมตีทั้งสองเครื่องโปรแกรมก็สามารถตรวจสอบและจัดการปฏิเสธการร้องขอโดยไม่กระทบกันได้ดังเช่น แถวที่ 33 และ 35 ในภาพที่ 4.11 ซึ่งเป็นการตรวจพบการโจมตี
- ผลการทำงานของโปรแกรมจะเกิด history file และผลของไฟร์วอลล์แสดงได้ดังภาพที่ 4.12 และภาพที่ 4.13 และ 4.14 ตามลำดับดังนี้



ภาพที่ 4.12 แสดงข้อมูลของ history file ที่เกิดขึ้นจากการทำงานของโปรแกรม



ภาพที่ 4.13 แสดงการทำงานของไฟร์วอลล์ในช่วงการเริ่มต้นการปฏิเสธการโจมตี



ภาพที่ 4.14 แสดงการทำงานของไฟร์วอลล์ในช่วงการยกเลิกปฏิเสธการโจมตี

จากภาพที่ 4.13 เป็นภาพการทำงานของไฟร์วอลล์ที่เกิดขึ้นขณะโปรแกรมทำงานซึ่งจะเห็นว่าในกรอบสี่เหลี่ยมที่ 1 เป็นการเพิ่มกฎของไฟร์วอลล์มีหมายเลขกฎคือ 00400 ซึ่งมีความหมายว่าเครื่อง IP 161.246.49.163 ถูกปฏิเสธการร้องขอข้อมูล port 8080จาก Server IP 161.246.49.147 ส่วนในกรอบสี่เหลี่ยมที่ 2 เป็นการเพิ่มขึ้นของกฎซึ่งเป็นการปฏิเสธการให้บริการในลักษณะเดียวกันคือเครื่อง IP 161.246.49.167 ถูกปฏิเสธการร้องขอข้อมูล port 8080 จากเครื่อง Server IP 161.246.49.147 เช่นกัน

จากภาพที่ 4.14 เป็นภาพแสดงการยกเลิกปฏิเสธการโจมตีโดยกรอบสี่เหลี่ยมที่ 1 แสดงการยกเลิกการโจมตีของเครื่อง IP 161.246.49.163 และในกรอบสี่เหลี่ยมที่ 2 แสดงการยกเลิกการโจมตีของเครื่อง IP 161.246.49.167

4.3 สรุปผลการทดสอบโปรแกรม

จากการทดสอบโปรแกรมในหัวข้อที่ 5.2 จะพบว่าโปรแกรมสามารถใช้งานได้ตามวัตถุประสงค์ของโครงงานคือ สามารถตรวจสอบการโจมตีที่เกิดจากการร้องขอข้อมูลของ client ซึ่งรับข้อมูลการต้องขอจาก access.log ของโปรแกรม Squid โดยการเทียบรูปแบบที่กำหนดไว้ใน Pattern file หลังจากทราบว่าSquid ถูกโจมตี โปรแกรมสามารถจำกัดการโจมตีโดยส่งคำสั่งเพื่อปฏิเสธการร้องขอที่ทำให้เกิดการโจมตีนั้นได้ โปรแกรมยังสามารถตรวจสอบการสิ้นสุดของการโจมตีและส่งคำสั่งไปยังไฟร์วอลล์เพื่อยกเลิกการปฏิเสธการร้องขอข้อมูลได้ ผลของการทำงานของโปรแกรมจะถูกจัดเก็บลงใน history.log ได้

ดังนั้นเครือข่ายที่ต้องการป้องกันการโจมตีที่อาจจะเกิดขึ้นจากภายในในลักษณะเดียวกันนี้สามารถนำโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชไปใช้ประโยชน์ได้ นอกจากนี้โปรแกรมยังให้ความสะดวกแก่ผู้ดูแลระบบโดยไม่จำเป็นต้องตรวจสอบพฤติกรรมของผู้ใช้งานภายในเพื่อป้องกันการโจมตีลักษณะที่โปรแกรมสามารถจัดการได้รวมทั้งไม่จำเป็นต้องดูแลจัดการกำหนดกฎที่ไฟร์วอลล์เพื่อป้องกันปัญหาที่เกิดขึ้นได้

4.4 ข้อเสนอแนะ

โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชสามารถนำไปพัฒนาต่อให้เกิดความสมบูรณ์และเกิดประโยชน์ให้มากขึ้นได้ โดยเพิ่มความสามารถดังต่อไปนี้

- เพิ่ม interface ให้ผู้ใช้สามารถกำหนดรูปแบบใน pattern file ให้ง่ายขึ้น

- พัฒนาโปรแกรมให้มีประสิทธิภาพมากขึ้นในเรื่องของการตรวจจับซึ่งจากเดิมใช้การตรวจจับแบบ misuse detection เป็นแบบ anomaly detection เพื่อให้โปรแกรมสามารถวิเคราะห์รูปแบบการร้องขอออบเจ็กต์ได้เองเพื่อความสะดวกของผู้ดูแลระบบและความรวดเร็วในการป้องกันการโจมตีที่อาจจะเกิดขึ้นขณะใดก็ได้
- พัฒนาโปรแกรมให้มีการทำงานที่รวดเร็วขึ้นเช่นการใช้วิธีการ search แบบ binary search แทนแบบ sequential search และปรับปรุงการโปรแกรมใช้ thread เพื่อความรวดเร็วของโปรแกรม
- โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชที่พัฒนาขึ้นเป็นการปฏิเสธการร้องขอในลักษณะการปฏิเสธการร้องขอหนึ่งครั้งต่อหนึ่งออบเจ็กต์ซึ่งควรปรับปรุงให้มีการตรวจสอบการปฏิเสธการร้องขอหนึ่งครั้งต่อหนึ่งเครื่อง
- โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชไม่มีส่วนของการตรวจสอบความผิดพลาดที่เกิดจากการกำหนดรูปแบบใน Pattern file และในส่วนของการกำหนดการเรียกใช้โปรแกรมภายนอกระบบในส่วนของคำสั่งที่เกี่ยวของกับไฟร์วอลล์ดังนั้นเพื่อประสิทธิภาพของโปแกรมควรเพิ่มเติมส่วนการป้องกันความผิดพลาดในส่วนนี้ด้วย
- ปรับปรุงประสิทธิภาพการทำงานของโปรแกรมโดยเปลี่ยนเทคนิคการโปรแกรมจากการวนรับข้อมูลการร้องขอเพื่อพิจารณาการทำงานของโปรแกรมในส่วนต่างๆมาเป็นการใช้ Thread เพื่อแยกส่วนของโปรแกรมทำงานออกจากกันซึ่งจะทำให้การใช้ทรัพยากรของระบบน้อยลง

บรรณานุกรม

พลสิทธิ์ พูลศิริ. 2544. "การพัฒนาโปรแกรมสำหรับหาอายุและขนาดของเว็บออบเจ็กต์จาก Squid's Log File" โครงงานการพัฒนาระบบงานวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

วรกุล เมืองสุวรรณ. 2543. "การพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บ." โครงงานการพัฒนาระบบงานวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Duane Wessels. 2000. SQUID Frequency Asked Question. [Online]. Available: http://www.squid-cache.org/Doc/FAQ/FAQ.html.

Rebecca Bace Bace. 2000. **Technology series Intrusion Detection.** Indianapolis: Macmillan Technical Publishing

Rebecca Bace and Peter Mell. NIST Special Publication on Intrusion Detection System. [Online]. Available: http://www.snort.org/docs/nist-ids.pdf.

ภาคผนวก

ภาคผนวก ก

การติดตั้งและใช้งานโปรแกรมตรวจจับการบุกรุกสำหรับเว็บแคช

1. การติดตั้งโปรแกรมตรวจจับการบุกรุกสำหรับเว็บแคช

ก่อนที่โปรแกรมจะสามารถทำงานได้ต้องมีการคอมไพล์โปรแกรมจาก source code โดยมีขั้นตอนการติดตั้งโปรแกรมดังต่อไปนี้

1.1 คัดลอก source code ทั้งหมดของโปรแกรมลงใน directory ที่ต้องการ

1.2 คอมไพล์ source code โดยใช้คำสั่งใช้ make –f makefile_ids ดังนี้

```
#make -f  makefile_ids
```

1.3 เมื่อเสร็จขั้นตอนนี้แล้วให้ใช้คำสั่ง ls เพื่อตรวจสอบไฟล์ ids ซึ่งเป็นไฟล์ที่สามารถเรียกใช้งานได้แล้ว

```
# ls
```

1.4 หลังจากคอมไพล์แล้วใช้คำสั่ง make –f makefile_ids clean เพื่อลบไฟล์ที่เป็นออบเจ็กต์ไฟล์ต่างๆออก

```
# make –f makefile_ids clean
```

2. การใช้งานโปรแกรมตรวจจับการบุกรุกสำหรับเว็บแคช

หลังจากติดตั้งโปรแกรมตามขั้นตอนแรกเสร็จเรียบร้อยแล้วตอนนี้พร้อมที่จะใช้โปรแกรมตามได้ขั้นตอนต่อไปนี้

2.1 เนื่องจากโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชต้องทำงานร่วมกับโปรแกรม Squid และโปรแกรม IP Firewall ของระบบปฏิบัติการฟรีบีเอสดี ดังนั้นต้องติดตั้งโปรแกรมทั้งสองและสั่งให้โปรแกรมทำงาน

2.2 กำหนดให้เครื่อง Client เรียกใช้งานโปรแกรม Squid ซึ่งอาจจะใช้วิธี Transparency หรือกำหนดใน Browser เช่นการกำหนดใน Internet Explorer ดังขั้นตอนต่อไปนี้

1) เปิดหน้าต่างของโปรแกรม Internet Explorer
2) เลือก Tools บน menu -> Internet Options -> Connections -> LAN Settings
3) เลือก Use a proxy server จากนั้นพิมพ์หมายเลขของเครื่อง server และ port ที่ได้กำหนดไว้ในการ config ของ Squid จากนั้นกด OK ถึงตอนนี้ก็สามารถเรียกใช้งานได้แล้ว

2.3 ก่อนใช้งานต้องกำหนดรูปแบบการโจมตีใน pattern file ของโปรแกรมโดยกำหนดที่ pattern.conf ซึ่งเป็น file ที่เก็บไว้ที่เดียวกับ source file การกำหนดรูปแบบสามารถดูตัวอย่างได้ดังภาพที่ 4.3

2.4 สั่งให้โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชทำงานโดยกำหนดให้รับข้อมูลจาก access.log ของ โปรแกรม Squid ซึ่งจะมี 2 กรณีดังนี้คือ

2.4.1 สั่งให้โปรแกรมทำงานแบบ Foreground Process การสั่งให้ทำงานในลักษณะ Foreground Process จะมีการแสดงการทำงานของโปรแกรมทำให้เห็นเหตุการณ์ต่างๆ ที่เกิดขึ้นด้วยโดยใช้คำสั่ง

tail –f –n 1 **pathของaccess.log** | **./ pathของโปรแกรม ids หมายเลข IPของเครื่อง Server**

ความหมายของคำสั่งคือ

- tail –f –n 1 **pathของaccess.log** : หมายถึงการตัดตอนข้อมูลเพียงบรรทัดสุดท้ายบรรทัดเดียวจากไฟล์ที่กำหนดเช่น tail –f –n 1 /usr/local/squid/logs/access.log หมายถึงให้แสดงข้อมูลบรรทัดสุดท้ายของ access.log จำนวน 1 บรรทัด
- **./ pathของโปรแกรม ids** : หมายถึง pathของโปรแกรมตรวจจับการโจมตีสำหรับเว็บแคช
- **หมายเลข IPของเครื่อง Server** : หมายถึงหมายเลขของเครื่อง Server

ตัวอย่าง

```
# tail –f –n 1 /usr/local/squid/logs/access.log | ids 161.246.49.147
```

หมายถึงให้สั่งให้โปรแกรมตรวจจับการโจมตีสำหรับเว็บแคชทำงานรับข้อมูลจาก access.log มีหมายเลข IP ของเครื่อง Server คือ 161.246.49.147

หมายเหตุ ในกรณีที่ต้องการยกเลิกการทำงานของโปรแกรมมีสามารถสั่งได้โดยกด Ctrl + C

2.4.2 สั่งให้โปรแกรมทำงานแบบ Background Process หากต้องการให้โปรแกรมทำงานเป็น Background Process สามารถใช้ & ใส่ไว้หลังคำสั่งได้ดังนี้

```
tail –f –n 1 /usr/local/squid/logs/access.log | ids 161.246.49.147 &
```

หมายเหตุ ในกรณีที่ต้องการยกเลิกการทำงานของโปรแกรมมีสามารถสั่งได้โดยใช้คำสั่ง Kill process โดยตรวจสอบหมายเลขของ Process โดยใช้คำสั่งดังนี้

```
# ps –ax | grep ids
```

คำสั่งดังกล่าวจะแสดง Process และหมายเลข Process จากนั้นใช้ Kill หมายเลข Process ดัง ตัวอย่างต่อไปนี้

```
# kill 616
```

ประวัติผู้เขียน

ชื่อ-นามสกุล นางสาวภัทราวดี เหมทานนท์

ประวัติการศึกษา คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

Email hamtanon@hotmail.com