รายงานความก้าวหน้าโครงงานทางวิศวกรรม

(Senior Project Progress Report)

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

4 พฤศจิกายน 2552

ชื่อโครงการ (ภาษาไทย) ไฟร์วอลล์ชนิดปรับตัวตามรูปแบบการโจมตี

(ภาษาอังกฤษ) iWall (Intelligent Firewall)

โดย

ชื่อ – นามสกุล		**เลขประจำตัว**	**ลายมือชื่อ**
1.นายวีกิจ	สัจจะมโนรมย์	4931243221	_____
2.นายสิขเรศ	ศุภปัญญา	4931252921	_____
3.นายอภิชิต	หาญบรรจง	4931255821	_____

อาจารย์ที่ปรึกษาโครงงาน

อาจารย์ที่ปรึกษาโครงการ	**ลายมือชื่อ**
อาจารย์ ดร.เกริก ภิรมย์โสภา	_____

บทนำ

รายงานความก้าวหน้าของโครงงานวิศวกรรม “ไฟร์วอลล์ชนิดปรับตัวตามรูปแบบการโจมตี (iWall)” ฉบับนี้ถูกเขียนขึ้นเพื่อนำเสนอความก้าวหน้าของโครงงานดังกล่าวในช่วงเวลาที่ผ่านมาตั้งแต่เริ่มส่งหัวข้อและนำเสนอหัวข้อโครงการผ่านไปเมื่อเทอมการศึกษาภาคต้นของปีการศึกษา 2552

โครงงานวิศวกรรม “ไฟร์วอลล์ชนิดปรับตัวตามรูปแบบการโจมตี (iWall)” เป็นการสร้างไฟร์วอลล์ที่มีความสามารถในการป้องกันการโจมตีแบบ Distributed Denial-of-Service (DDoS) ทาง HTTP Port หรือ Port 80 ซึ่งไฟร์วอลล์จะปรับตัวตามรูปแบบการโจมตีของ DDoS ที่พบ โดยจะนำเสนอถึง อุปกรณ์ที่ใช้ วิธีการติดตั้ง การติดต่อสื่อสารระหว่างอุปกรณ์ พื้นฐานความรู้และอัลกอริทึมที่ใช้ และการทดสอบประเมินผลโครงงานนี้

รายงานความก้าวหน้าของโครงงานวิศวกรรมฉบับนี้จะนำเสนอโดยกล่าวควบคู่ไปกับแผนการดำเนินงานที่ได้นำเสนอไปในข้อเสนอโครงการฉบับก่อนหน้าโดยจะนำเสนอจากภาพรวมของโครงงาน ปัญหาและวิธีการแก้ปัญหา ขั้นตอนการปฏิบัติในแต่ละงานอย่างละเอียดประกอบไปกับการวัดผลในแต่ละส่วนของโครงงาน

สารบัญ

ภาพรวมของโครงงาน

“ไฟร์วอลล์ชนิดปรับตัวตามรูปแบบการโจมตี (iWall)” คือ ระบบไฟร์วอลล์ (Firewall) ที่ป้องกันการโจมตีจากภายนอกเครือข่ายขององค์กร ภายใต้หลักการ “ศึกษา เรียนรู้ และป้องกัน” ซึ่งเป็นการผสมผสานความรู้ด้านการทำเหมืองข้อมูล (Data Mining) ปัญญาประดิษฐ์ (Artificial Intelligence) การเรียนรู้ของเครื่อง (Machine Learning) และความมั่นคงของคอมพิวเตอร์ (Computer Security) แล้วนำมาประยุกต์ใช้ในการพัฒนาไฟร์วอลล์ให้มีความสามารถในการป้องกันการโจมตีแบบ Distributed Denial-of-Service (DDoS) ผ่านทาง HTTP Port หรือ Port 80 โดยระบบไฟร์วอลล์สามารถเรียนรู้และปรับเปลี่ยนกฎของไฟร์วอลล์ (Firewall Rule) ให้เหมาะสมกับการโจมตีที่พบ ซึ่งระบบไฟร์วอลล์จะใช้เครื่อง Server จำนวน 2 เครื่อง ได้แก่ Rule Server และ Filter Server โดยแบ่งออกเป็น 5 Module แยกตามหน้าที่การทำงาน ได้แก่ Firewall Module, Traffic Anomaly Identification Module, Bandwidth Control Module, Log Analyzer Module และ Rule Generator Module

สรุปความก้าวหน้า

ในหัวข้อนี้จะแบ่งเป็นสามส่วน ได้แก่ ขั้นตอนที่ทำเสร็จแล้ว ขั้นตอนที่กำลังทำ และขั้นตอนที่จะปฏิบัติต่อหลังจากนี้โดยแสดงรายละเอียดดังนี้

ขั้นตอนที่ทำเสร็จแล้ว

ในระหว่างเดือนกันยายน 52 ถึงเดือนตุลาคม 52 ได้ปฏิบัติงานตามแผนงานที่วางไว้ทั้งหมด 4 ขั้นตอนดังนี้

1. วิเคราะห์ปัญหา

การโจมตีแบบ Denial-of-Service (DoS) เป็นการโจมตีระบบเครือข่ายโดยการทำให้ระบบเครือข่ายหรือ Server ไม่สามารถให้บริการได้ตามปกติ ซึ่งการโจมตีแบบ DoS จะเน้นไปที่การใช้ทรัพยากรของระบบเครือข่ายให้หมด ถือเป็นการโจมตีจุดอ่อนหรือข้อจำกัดของระบบ โดยสามารถทำได้หลายรูปแบบ ดังนี้

1.SYN Flood Attack คือการส่ง Packet TCP/SYN โดยใช้ IP ที่ไม่มีอยู่จริง

2.Mail Bomb คือการส่ง Mail ที่มีขนาดใหญ่เป็นจำนวนมากเข้าไปเพื่อให้เนื้อที่ใน Mail box เต็ม

3.Smurf Attack คือการส่งปลอม IP address เป็นของเครื่องเป้าหมายแล้วจึงส่ง Packet ping เข้าไปหา

4.Broadcast Address เพื่อให้กระจาย Packet เข้าไปทุกเครื่องแล้วหลังจากนั้นเมื่อทุกเครื่องได้รับแล้วจึงตอบ Packet ไปหาเครื่องเป้าหมายซึ่งอาจเกิด Buffer Overflow ได้

5.Fraggle Attack เหมือนกับ Smurf Attack แต่เปลี่ยนเป็นใช้ Packet ของ UDP แทน

6.Ping of Death คือการส่ง Packet Ping ที่มีขนาดใหญ่เกินกว่าปกติเข้าไปที่เครื่องเป้าหมาย

7.Teardrop Attack คือการส่ง Packet ที่ไม่สามารถประกอบได้ไปให้เครื่องเป้าหมายเพื่อให้เกิดความสับสน

8.ICMP Flood Attack คือการส่ง Packet Ping เข้าไปที่เครื่องเป้าหมายเป็นจำนวนมาก

9.UDP Flood Attack คือการส่ง Packet ของ UDP จำนวนมากไปที่เครื่องเป้าหมาย

การโจมตีแบบ Distributed Denial-of-Service (DDoS) มีหลักการการโจมตีลักษณะเดียวกันกับ DoS แต่จะโจมตีโดยใช้เครื่องคอมพิวเตอร์จำนวนมาก แทนที่จะใช้เครื่องคอมพิวเตอร์เครื่องเดียวเหมือน DoS ซึ่งการโจมตีแบบ DDoS โดยทั่วไปจะใช้ Botnet ในการโจมตี ซึ่ง Botnet คือเครือข่ายของเครื่องคอมพิวเตอร์ที่ถูกควบคุมจากระยะไกล โดยอาศัยโปรแกรมที่ฝังตัวไว้ตามเครื่องคอมพิวเตอร์เหล่านั้น เครื่องคอมพิวเตอร์เหล่านี้เรียกว่า Zombie เมื่อ Zombie ได้รับคำสั่งจากผู้ควบคุม Botnet จะทำการโจมตีไปยังเป้าหมายในช่วงเวลาเดียวกัน

ไฟร์วอลล์ในปัจจุบันสามารถป้องกันการโจมตีแบบ DDoS ได้อย่างมีขีดจำกัด เพราะกฎของไฟร์วอลล์เป็นกฎที่ถูกตั้งมาแบบตายตัวหรือถูกกำหนดโดยผู้ใช้เท่านั้น ทำให้ไฟร์วอลล์ไม่สามารถป้องกันการโจมตีที่อยู่นอกเหนือรูปแบบที่ถูกกำหนดไว้ได้ การสร้างไฟร์วอลล์ที่สามารถปรับตัวได้ตามรูปแบบของการโจมตีจึงมีความสำคัญมาก

2. วิเคราะห์ความต้องการ

iWall ต้องสามารถตรวจจับและป้องกันการโจมตีแบบ DDoS ผ่านทาง HTTP Port รูปแบบที่ iWall เคยพบและไม่เคยพบมาก่อนได้ และกฎของ iWall ต้องสามารถปรับเปลี่ยนตามการโจมตีที่พบ รวมทั้งปรับเปลี่ยนการป้องกันตามขีดจำกัดของระบบเครือข่ายได้ เช่น ควบคุม Traffic ให้เหมาะสมกับ Bandwidth ของระบบเครือข่ายองค์กร

iWall ถูกพัฒนาบนระบบปฏิบัติการ Linux เนื่องจาก Linux เป็นซอฟต์แวร์ประเภท Open Source และมีซอฟต์แวร์ประเภท Open Source จำนวนมากที่ทำงานบนระบบปฏิบัติการ Linux ทำให้มีซอฟต์แวร์หลากหลายระดับความสามารถ จึงสามารถเลือกใช้ซอฟต์แวร์ที่สอดคล้องและเหมาะสมต่อการพัฒนา iWall ได้

การโจมตีแบบ DDoS รูปแบบที่ iWall ไม่เคยพบมาก่อนจะถูกวิเคราะห์หาความสัมพันธ์และรูปแบบของการโจมตีจาก Log ของ Traffic ที่ได้ในแต่ละช่วงเวลา

3. ศึกษาหาข้อมูลที่เกี่ยวข้อง

3.1 Netfilter / iptables

Netfilter และ iptables คือแม่แบบในการทำงานของ Packet Filtering Firewall ซึ่งสนับสนุนการทำงานโดย kernel ของ Linux เวอร์ชัน 2.4.X และ 2.6.X ในขณะที่สำหรับในเวอร์ชัน 2.2.X จะใช้ “ipchains” และในเวอร์ชัน 2.0.X จะใช้ “ipfwadm” ซึ่งจะมีรูปแบบการใช้คำสั่งที่แตกต่างกันออกไป

iptables สามารถรองรับการทำงานได้ทั้ง Stateless Packet Filtering ซึ่งจะตรวจสอบเฉพาะส่วน Header ของแพ็กเก็ตและ Stateful Packet Filtering ซึ่งจะตรวจสอบข้อมูลได้ทั้งส่วนที่เป็น Header และข้อมูล (Payload) ซึ่งทำให้สามารถตรวจสอบการทำงานได้ละเอียดมากขึ้น นอกจากนี้ iptables ยังสนับสนุนการทำงานของ NAT/NAPT (Network Address and Port Translation) อีกด้วย

Option พื้นฐานของ iptables มีดังนี้

-A เพิ่มกฎใหม่เข้าไปต่อท้ายชุดของกฎที่มีอยู่แล้ว

-L เรียกดูชุดของกฎในปัจจุบัน

-m conntrack อนุญาตให้กฎการคัดกรอง (Filter rules) เข้าคู่กันตามสถานะของการเชื่อมต่อ (ctstate)

--ctstate แบ่งออกเป็น 4 สถานะ คือ NEW, RELATED, ESTABLISHED และ INVALID

-m limit กำหนดจำนวนครั้งสูงสุดในการเข้าคู่กันของกฎ

--limit กำหนดจำนวนครั้งสูงสุดในการเข้าคู่กันของกฎภายในระยะเวลา

-p ระบุ Protocol ที่ใช้ในการติดต่อ

--dport ระบุ Port ปลายทาง

-j ระบุรูปแบบการทำงานเมื่อตรงตามเงื่อนไข ซึ่งมี 4 รูปแบบดังนี้

1.ACCEPT ยอมรับ Packets

2.REJECT ปฏิเสธ Packets และแจ้งไปยังต้นทาง

3.DROP ปฏิเสธ Packets

4.LOG บันทึก Log และดำเนินการตรวจสอบชุดของกฎต่อไป

--log-prefix เพิ่มข้อความหน้า Log ที่ทำการบันทึก

--log-level กำหนดระดับของการบันทึก Log

-I เพิ่มกฎใหม่เข้าไปในชุดของกฎตามตำแหน่งที่กำหนด

-R เปลี่ยนกฎในชุดของกฎตามที่กำหนด

-D ลบกฎในชุดของกฎตามที่กำหนด

-s ระบุ Address/mask ต้นทาง

-d ระบุ Address/mask ปลายทาง

คำสั่งของ iptables ที่ใช้ในโครงงานนี้แบ่งออกเป็น 7 ส่วน ดังนี้

1.Allowing Established Sessions

```
# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

2.Allowing Incoming Traffic

```
# iptables -A INPUT --dport 80 -j ACCEPT –s 192.168.1.1
```

3.Blocking Incoming Traffic

```
# iptables -A INPUT --dport 80 -j DROP –s 192.168.1.1
```

4.Inserting Rules

```
# iptables -I INPUT 1 --dport 80 -j DROP –s 192.168.1.1
```

5.Replacing Rules

```
# iptables -R INPUT 1 --dport 80 -j DROP –s 192.168.1.1
```

6.Deleting Rules

```
# iptables -D INPUT --dport 80 -j DROP –s 192.168.1.1
```

7.Logging

```
# iptables -I INPUT 5 -m limit --limit 5/min -j LOG --log-prefix "iptables
denied: " --log-level 7
```

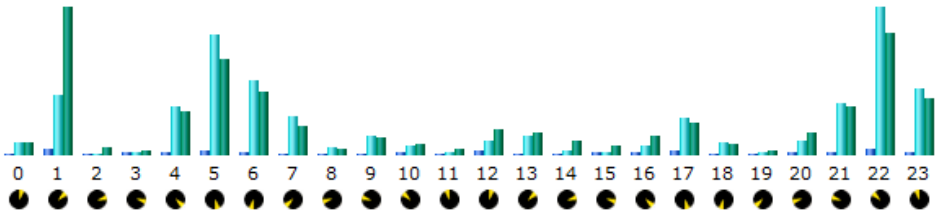
3.2 AWStats

AWStats เป็นเครื่องมือ Open Source ที่เขียนด้วยภาษา PERL สำหรับวิเคราะห์ข้อมูลจากอินเทอร์เน็ต เช่น Web, Streaming Media, Mail หรือ FTP Server เป็นต้น การทำงานของ AWStats ทำโดยการรับ Log มาวิเคราะห์และแสดงผลการวิเคราะห์ในรูปของ HTML ซึ่งผลการวิเคราะห์ส่วนใหญ่จะถูกแสดงในรูปของตารางและกราฟ โดยผลการวิเคราะห์นี้สามารถสร้างได้ผ่าน Command Line

Attribute ที่ได้จากการวิเคราะห์ Log โดย AWStats ที่สนใจในโครงงานนี้ได้แก่

1.ข้อมูล Traffic แยกตามชั่วโมง

Hours



Hours	Pages	Hits	Bandwidth	Hours	Pages	Hits	Bandwidth
00	3	54	113.44 KB	12	16	63	244.88 KB
01	28	260	1.37 MB	13	7	85	213.84 KB
02	3	3	65.58 KB	14	7	16	127.20 KB
03	9	9	40.04 KB	15	10	10	92.27 KB
04	9	214	408.56 KB	16	8	38	180.38 KB
05	21	531	911.86 KB	17	18	163	298.52 KB
06	11	327	597.08 KB	18	7	53	106.68 KB
07	7	167	270.00 KB	19	1	11	31.73 KB
08	3	35	58.11 KB	20	10	62	216.51 KB
09	6	85	158.87 KB	21	11	230	466.17 KB
10	8	43	105.72 KB	22	23	649	1.13 MB
11	2	11	52.55 KB	23	12	293	531.41 KB

รูป 1 แสดงรายละเอียดข้อมูล Traffic แยกตามชั่วโมง

2.ข้อมูล Traffic แยกตาม Hosts

Hosts (Top 10) - Full list - Last visit - Unresolved IP Address

Hosts : 0 Known, 157 Unknown (unresolved ip) 148 Unique visitors	GeoIP Country	GeoIP City	GeoIP Region	Info.	Pages	Hits	Bandwidth	Last visit
88.163.130.195	France	Paris	Unknown	?	20	208	1.28 MB	03 Nov 2009 - 01:59
91.54.26.156	Germany	Unknown	Unknown	?	13	13	57.84 KB	01 Nov 2009 - 17:11
217.94.63.151	Germany	Ronnenberg		?	9	56	145.90 KB	02 Nov 2009 - 12:29
94.236.23.242	Great Britain	Unknown	Unknown	?	9	9	196.73 KB	01 Nov 2009 - 16:25
81.227.178.214	Sweden	Tallebo		?	8	8	35.59 KB	02 Nov 2009 - 03:38
201.22.249.205	Brazil	Curitiba		?	6	50	49.82 KB	02 Nov 2009 - 01:10
79.237.121.71	Germany	Unknown	Unknown	?	4	4	17.80 KB	02 Nov 2009 - 18:55
65.55.230.216	United States	Unknown	Washington	?	4	4	13.91 KB	02 Nov 2009 - 17:22
194.8.75.155	Great Britain	Unknown		?	4	4	52.97 KB	02 Nov 2009 - 14:08
131.173.84.108	Germany	Unknown		?	4	35	73.83 KB	02 Nov 2009 - 17:27
Others					159	3021	5.76 MB	

รูป 2 แสดงรายละเอียดข้อมูล Traffic แยกตาม Hosts

3.ผู้เยี่ยมชมที่เป็น Robots หรือ Spiders

Robots/Spiders visitors (Top 10) - Full list - Last visit			
30 different robots*	Hits	Bandwidth	Last visit
Yahoo Slurp	903+22	671.78 KB	03 Nov 2009 - 06:51
MSNBot	195+130	1.48 MB	03 Nov 2009 - 06:48
Java (Often spam bot)	88	525.31 KB	02 Nov 2009 - 23:17
Unknown robot (identified by 'bot*')	50+2	148.39 KB	02 Nov 2009 - 13:36
Googlebot	34+5	82.55 KB	03 Nov 2009 - 06:21
BSpider	17+4	71.46 KB	03 Nov 2009 - 05:03
Unknown robot (identified by empty user agent string)	17+4	371.61 KB	03 Nov 2009 - 06:30
Voila	17+4	66.46 KB	03 Nov 2009 - 04:41
BaiDuSpider	15+2	123.42 KB	03 Nov 2009 - 06:53
Sogou Spider	8+6	21.05 KB	03 Nov 2009 - 03:52
Others	44+36	280.73 KB	

รูป 3 แสดงผู้เยี่ยมชมที่เป็น Robots หรือ Spiders

4.ข้อมูล Traffic แยกตามระยะเวลาเยี่ยมชม

Visits duration		
Number of visits: 159 - Average: 92 s	Number of visits	Percent
0s-30s	146	91.8 %
30s-2mn	3	1.8 %
2mn-5mn	1	0.6 %
5mn-15mn	5	3.1 %
15mn-30mn		
30mn-1h	2	1.2 %
1h+	1	0.6 %
Unknown	1	0.6 %

รูป 4 แสดงรายละเอียดข้อมูล Traffic แยกตามระยะเวลาเยี่ยมชม

3.3 HTTP

เกณฑ์วิธีขนส่งข้อความหลายมิติ หรือ HTTP (Hyper Text Transport Protocol) คือโพรโทคอลในระดับชั้นโปรแกรมประยุกต์เพื่อการแจกจ่ายและการทำงานร่วมกันกับสารสนเทศของสื่อผสม ใช้สำหรับการรับทรัพยากรที่เชื่อมโยงกับภายนอก ซึ่งนำไปสู่การจัดตั้งเวิลด์ไวด์เว็บ โดย HTTP/1.1 เป็นรุ่นที่ใช้กันอย่างกว้างขวางในปัจจุบัน

HTTP เป็นมาตรฐานในการร้องขอและการตอบรับระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย ซึ่งเครื่องลูกข่ายคือผู้ใช้ปลายทาง (end-user) และเครื่องแม่ข่ายคือเว็บไซต์ เครื่องลูกข่ายจะสร้างการร้องขอ HTTP ผ่านทางเว็บบราวเซอร์ เว็บครอว์เลอร์ หรือเครื่องมืออื่น ๆ ที่จัดว่าเป็น ตัวแทนผู้ใช้ (user agent) ส่วนเครื่องแม่ข่ายที่ตอบรับ ซึ่งเก็บบันทึกหรือสร้าง ทรัพยากร (resource) อย่างเช่นไฟล์ HTML หรือรูปภาพ จะเรียกว่า เครื่องให้บริการต้นทาง (origin server) ในระหว่างตัวแทนผู้ใช้กับเครื่องให้บริการต้นทางอาจมีสื่อกลางหลายชนิด อาทิ Proxy, Gateway และTunnel นอกจากนี้ HTTP ไม่ได้จำกัดว่าจะต้องใช้ชุดเกณฑ์วิธีอินเทอร์เน็ต (TCP/IP) เท่านั้น แม้ว่าจะเป็นการใช้งานที่นิยมมากที่สุดบนอินเทอร์เน็ตก็ตาม โดยแท้จริงแล้ว HTTP สามารถ "นำไปใช้ได้บน Protocol อินเทอร์เน็ตอื่น ๆ หรือบนเครือข่ายอื่นก็ได้" HTTP คาดหวังเพียงแค่การสื่อสารที่เชื่อถือได้ นั่นคือ Protocol ที่มีการรับรองเช่นนั้นก็สามารถใช้งานได้

ปกติเครื่องลูกข่าย HTTP จะเป็นผู้เริ่มสร้างการร้องขอก่อน โดยเปิดการเชื่อมต่อด้วยเกณฑ์วิธีควบคุมการขนส่งข้อมูล (TCP) ไปยัง Port เฉพาะของเครื่องแม่ข่าย (Port 80 เป็นค่าปริยาย) เครื่องแม่ข่าย HTTP ที่เปิดรอรับอยู่ที่ Port นั้น จะเปิดรอให้เครื่องลูกข่ายส่งข้อความร้องขอเข้ามา เมื่อได้รับการร้องขอแล้ว เครื่องแม่ข่ายจะตอบรับด้วยข้อความสถานะอันหนึ่ง ตัวอย่างเช่น "HTTP/1.1 200 OK" ตามด้วยเนื้อหาของมันเองส่งไปด้วย เนื้อหานั้นอาจเป็นแฟ้มข้อมูลที่ร้องขอ ข้อความแสดงข้อผิดพลาด หรือข้อมูลอย่างอื่นเป็นต้น

ทรัพยากรที่ถูกเข้าถึงด้วย HTTP จะถูกระบุโดยใช้ตัวระบุแหล่งทรัพยากรสากล (URI) (หรือเจาะจงลงไปก็คือ ตัวชี้แหล่งในอินเทอร์เน็ต (URL)) โดยใช้ http: หรือ https: เป็นแผนของตัวระบุ (URI scheme)

3.4 HTTP Header

HTTP Header แบ่งออกเป็น 2 ประเภท ได้แก่ Request และ Response แต่ในโครงงานนี้สนใจเฉพาะ Inbound Traffic จึงคำนึงถึง HTTP Header เฉพาะประเภท Request เท่านั้น โดยจะนำ HTTP Heer ดังต่อไปนี้มาทำการวิเคราะห์

ตาราง 1 แสดงรายละเอียดของ HTTP Header ที่สนใจในโครงงานนี้

ส่วนหัว	คำอธิบาย	ตัวอย่าง
Authorization	รหัสรับรองการยืนยันตนของผู้ใช้	Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Connection	ประเภทของการเชื่อมต่อที่ตัวแทนผู้ใช้ต้องการ	Connection: close
Content-Length	ขนาดของเนื้อหาที่ร้องขอในรูปแบบ Octets	Content-Length: 348
Content-Type	ชนิด MIME ของเนื้อหาที่ร้องขอ (ใช้กับ POST และ PUT)	Content-Type: application/x-www-form-urlencoded
Date	วันที่และเวลาที่ข้อความนั้นถูกส่งไป	Date: Tue, 15 Nov 1994 08:12:31 GMT
Expect	แสดงพฤติกรรมที่ต้องการจากเครื่องแม่ข่ายบางชนิด	Expect: 100-continue
Proxy-Authorization	รหัสรับรองการยืนยันตนสำหรับการเชื่อมต่อผ่านพร็อกซี	Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

Range	ร้องขอข้อมูลทรัพยากรบางส่วน	Range: bytes=500-999
User-Agent	สายอักขระแสดงชื่อของโปรแกรมตัวแทนผู้ใช้	User-Agent: Mozilla/5.0 (Linux; X11)
Via	แจ้งให้เครื่องแม่ข่ายทราบว่าการร้องขอส่งผ่านมาจากพร็อกซีใด	Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
Warn	คำเตือนทั่วไปเกี่ยวกับปัญหาที่อาจเกิดขึ้นในเนื้อหาทรัพยากร	Warn: 199 Miscellaneous warning

3.5 Algorithms

3.5.1 Traffic Baseline Algorithm

Algorithm นี้แบ่งการสถานะการทำงานออกเป็น 2 ส่วน

1.Learning Mode ในส่วนนี้ Algorithm จะทำการคำนวณหาค่า Baseline ที่จะใช้เปรียบเทียบกับปริมาณ Traffic ในอนาคต

2.Dynamic Mode ในส่วนนี้ Algorithm จะทำการปรับตัวตามเงื่อนไขในปัจจุบัน และปรับค่า Baseline ให้เหมาะสม

3.5.1.1 Learning Mode

Algorithm จะเริ่มทำงานในส่วนนี้เป็นอันดับแรก โดยจะทำการจัดเรียงปริมาณ Traffic และนำค่า Percentile Rank ที่ 75 และ 25 แล้วหาค่ามัธยฐาน จากนั้นจึงหาขอบบนและขอบล่างโดยนำค่ามัธยฐานบวกและลบด้วยส่วนต่างระหว่างค่า Percentile Rank ที่ 75 และ 25 ตามลำดับ แล้วตัดข้อมูลที่อยู่นอกเหนือขอบบนและขอบล่าง นำข้อมูลที่เหลือมาหาค่าเฉลี่ยจะได้ Traffic Baseline เริ่มต้น

3.5.1.2 Dynamic Mode

เมื่อ Algorithm หาค่า Traffic Baseline เริ่มต้นได้แล้ว จะเริ่มทำงานในส่วนที่สองนี้ โดยผู้ใช้ต้องระบุขอบเขตซึ่งเป็นตัวกำหนดช่วงของปริมาณ Traffic ที่สามารถยอมรับได้ (Tolerance) โดยมีการทำงานดังนี้

1.ในกรณีที่ปริมาณ Traffic อยู่ในช่วงที่สามารถยอมรับได้ ค่า Traffic Baseline จะมีค่าเท่ากับค่าเฉลี่ยของปริมาณ Traffic ในปัจจุบันและปริมาณ Traffic ในอดีต

2.ในกรณีที่ปริมาณ Traffic มีค่าเกินกว่าหรือต่ำกว่าช่วงที่สามารถยอมรับได้ ค่า Traffic Baseline จะมีค่าเท่าเดิม

3.5.2 Traffic Anomaly Identification Algorithm

Algorithm นี้จะทำการตรวจจับความผิดปกติของ Inbound Traffic โดยเปรียบเทียบปริมาณ Traffic ในปัจจุบันกับ Traffic Baseline ถ้าปริมาณ Traffic ในปัจจุบันมีค่ามากกว่า Traffic Baseline เกินช่วงที่สามารถยอมรับได้ ถือว่า Traffic ในช่วงนั้นมีความผิดปกติ แต่ถ้าปริมาณ Traffic มีค่าอยู่ในช่วงที่สามารถยอมรับได้หรือมีค่าน้อยกว่า Traffic Baseline จะถือว่า Traffic ในช่วงนั้นปกติ

เมื่อตรวจพบว่า Traffic มีความผิดปกติ จะใช้เกณฑ์ในการตัดสินว่าเครื่องต้นทางใดมีส่วนทำให้เกิดความผิดปกติ โดยตรวจสอบจากปริมาณ Traffic ที่เกิดจากแต่ละเครื่องต้นทางในช่วงเวลานั้น ว่าเครื่องต้นทางใดทำให้ปริมาณ Traffic สูงเกินช่วงที่สามารถยอมรับได้ ถือว่าเครื่องต้นทางเหล่านั้นทำให้เกิดความผิดปกติ

3.5.3 Bandwidth Control Algorithm

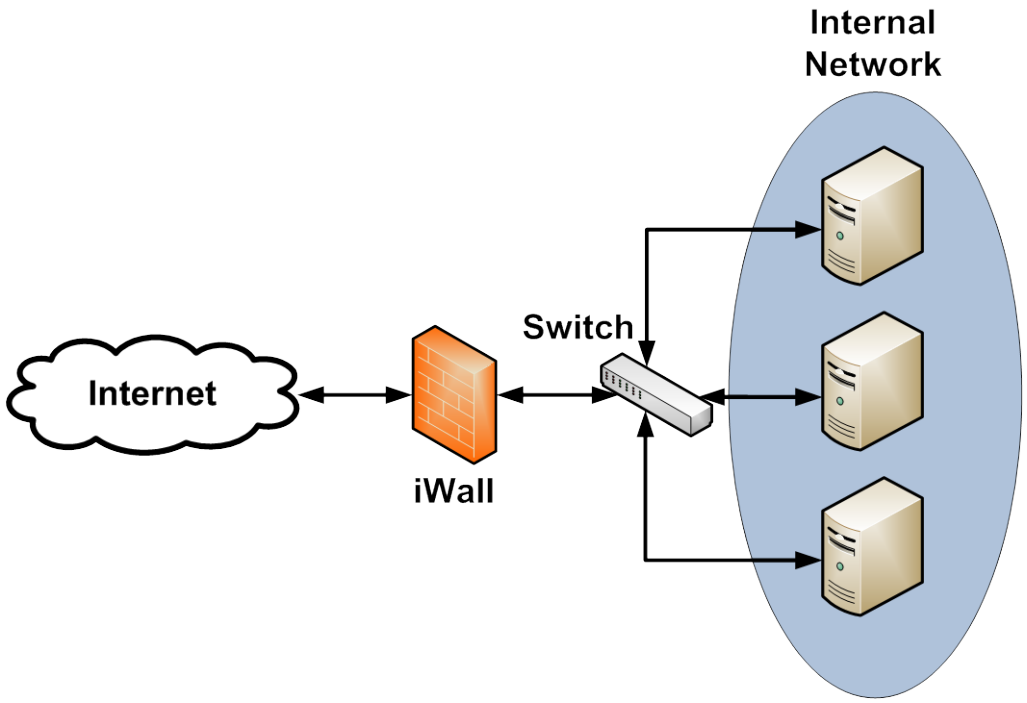
Algorithm นี้ใช้เกณฑ์ในการคัดกรอง Packet ที่ผ่านเข้าไปในเครือข่ายขององค์กร โดยเปรียบเทียบ Bandwidth ในปัจจุบันกับ Maximum Bandwidth ซึ่งถูกกำหนดโดยผู้ใช้ ถ้า Bandwidth ในปัจจุบันมีค่าเกิน Maximum Bandwidth จะคัดเลือก Packet ให้ผ่านเข้าไปในเครือข่ายขององค์กรตามหลักการ First-come, First-served กล่าวคือ Packet ที่มาถึงปลายทางก่อนจะได้รับการให้บริการ ส่วน Packet ที่มาถึงปลายทางในภายหลังจะถูกคัดออก

3.5.4 Blocking Algorithm

Algorithm นี้จะควบคุมเวลาในการ Block แต่ละ Source IP Address ตามระดับความรุนแรง โดยความรุนแรงนี้วัดจากอัตราส่วน Severity ซึ่งเป็นอัตราส่วนระหว่างปริมาณ Traffic ของแต่ละ Source IP Address กับ Traffic Baseline ถ้าอัตราส่วน Severity ของ Source IP Address ใดมีค่าสูง Source IP Address นั้นจะถูก Block เป็นเวลานานมากขึ้น

4. ออกแบบสถาปัตยกรรม iWall

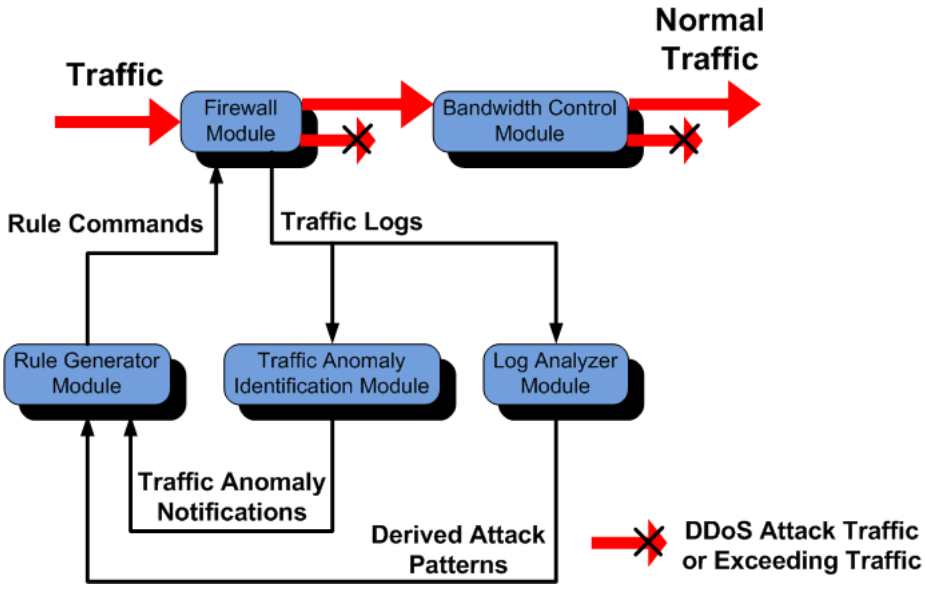
4.1 สถาปัตยกรรมเชิงกายภาพของ iWall



รูป 5 สถาปัตยกรรมเชิงกายภาพของ iWall

iWall ประกอบไปด้วยเครื่อง Server 1 เครื่อง ทำหน้าที่ป้องกันการโจมตีแบบ DDoS ผ่านทาง HTTP Port โดยสังเกตปริมาณ Traffic วิเคราะห์ Log และปรับเปลี่ยนกฎให้เหมาะสม

4.2 สถาปัตยกรรมเชิงตรรกะของ iWall



รูป 6 สถาปัตยกรรมเชิงตรรกะของ iWall

4.2.1 Firewall Module

เป็นไฟร์วอลล์ทำงานบนระบบปฏิบัติการ Linux คือ Netfilter / iptables ซึ่งเป็นไฟร์วอลล์ใน Kernel ของ Linux โดยทั่วไปแล้ว iptables จะทำงานบน Open Systems Interconnection (OSI) Layer 4 (Stateful Firewall) แต่สามารถติดตั้ง Patch เพื่อทำให้ใช้งานบน OSI Layer 7 (Application Inspection) ได้

Netfilter / iptables เป็นจุดเชื่อมต่อระหว่างเครือข่ายภายในองค์กรและเครือข่ายภายนอก โดยทำหน้าที่คัดกรองแพ็กเก็ตที่ส่งจากภายนอกเครือข่ายขององค์กรก่อนปล่อยให้เข้าสู่เครือข่ายภายในองค์กร

4.2.2 Traffic Anomaly Identification Module

ทำหน้าที่คอยเฝ้าระวังความผิดปกติของ Traffic ที่ผ่าน Firewall Module เข้ามาได้ โดยเปรียบเทียบกับ Traffic Baseline ซึ่ง Traffic Baseline ได้จาก Traffic Baseline Algorithm โดยวิเคราะห์ตาม Traffic Anomaly Identification Algorithm ถ้าพบความผิดปกติของ Traffic ในปัจจุบัน จะส่ง Source IP Address ที่ทำให้เกิดความผิดปกติ พร้อมทั้งส่งค่าที่ได้จากการคำนวณหาระยะเวลาในการ Block แต่ละ Source IP Address ตาม Blocking Algorithm ไปบอก Rule Generator Module

4.2.3 Bandwidth Control Module

ทำหน้าที่จำกัด Bandwidth ของ Inbound Traffic ให้เหมาะสมกับขีดจำกัดของเครือข่ายภายในองค์กรตาม Bandwidth Control Algorithm

4.2.4 Log Analyzer Module

ทำหน้าที่นำ Traffic Log ที่ได้จาก Traffic Anomaly Identification Module มาวิเคราะห์ด้าน Application Anomaly โดยใช้เครื่องมือ AWStats วิเคราะห์ Log แล้วนำผลที่ได้มาทำ Data Mining หาความสัมพันธ์ จากนั้นจึงส่งผลการวิเคราะห์ไปแจ้ง Rule Generator Module

4.2.5 Rule Generator Module

ทำหน้าที่สร้างกฎตามรูปแบบชุดคำสั่งของ iptables โดยกฎถูกสร้างขึ้นจากผลการวิเคราะห์ที่ได้จาก Traffic Anomaly Identification Module และ Log Analyzer Module

ขั้นตอนที่กำลังปฏิบัติอยู่

1. ศึกษาภาษา PERL

เนื่องด้วยโครงงานนี้ใช้ AWStats เป็นเครื่องมือเสริมในการวิเคราะห์ Traffic Log ของไฟร์วอลล์ และ AWStats เป็น Open Source ที่เขียนด้วยภาษา PERL และจากการศึกษาเบื้องต้นของภาษา PERL คาดว่าภาษา PERL เหมาะสมในการนำมาพัฒนา iWall จึงเลือกภาษา PERL มาเป็นภาษาที่ใช้ในการพัฒนา Module ต่างๆทั้งหมด

ภาษา PERL (Practical Extraction and Report Language) เป็นภาษาที่มีรากฐานการพัฒนามาจากภาษา C โดยเหตุผลที่ทำให้ภาษา PERL เป็นที่นิยมคือ

1.ไม่เสียค่าใช้จ่าย

2.ติดตั้งมาแล้วกับ Unix Standard หรือ Linux

3.มีรากฐานมาจากภาษา C ซึ่งเป็นที่นิยม

4.มีความสามารถพิเศษด้านการติดต่อระบบเพื่อการดูแลจัดการระบบ

5.มี Function สำเร็จรูปมาให้

6.มี Site Reference จำนวนมาก

ภาษา PERL มีตัวแปร 4 ชนิด ได้แก่

1.Scalar สามารถเก็บข้อมูลได้ 1 อย่าง โดยจะเป็น Number, String หรือ Reference ก็ได้

2.Array เป็นเสมือนกลุ่มของ Scalar ที่ถูกเรียงไว้

3.Hash หรือ Associative Array เป็นเสมือนตู้ล็อกเกอร์สำหรับเก็บ Scalar กุญแจที่จะใช้ไขตู้ล็อกเกอร์จะเรียกว่า Keys

4.File Handle เป็นตัวแปรที่ใช้สำหรับ I/O โดยเฉพาะ อาจจะใช้สำหรับรับการสั่งงานจากผู้ใช้ผ่านทาง Standard Input หรือใช้สำหรับแสดงผลออกทาง Standard Output

2. สร้างข่ายงานประสาทเทียม

โครงงานนี้ใช้ข่ายงานประสาทเทียม (Artificial Neural Network) ในการวิเคราะห์ Attribute ที่ได้จากการวิเคราะห์ Log ของ AWStats และวิเคราะห์ Traffic Log

ข่ายงานประสาทเทียม คือ โมเดลทางคณิตศาสตร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (connectionist) แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาข่ายงานไฟฟ้าชีวภาพ (bioelectric network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ นิวรอน (neurons) และ จุดประสานประสาท (synapses) ตามโมเดลนี้ ข่ายงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาท จนเป็นเครือข่ายที่ทำงานร่วมกัน

ข่ายงานประสาทแบบป้อนไปหน้า (feedforward) ประกอบด้วยเซตของบัพ (node) ซึ่งอาจจะถูกกำหนดให้เป็นบัพอินพุต (input nodes) บัพเอาต์พุต (output nodes) หรือ บัพอยู่ระหว่างกลางซึ่งเรียกว่า บัพฮิดเดน (hidden nodes) มีการเชื่อมต่อระหว่างบัพ (หรือนิวรอน) โดยกำหนดค่าน้ำหนัก (weight) กำกับอยู่ที่เส้นเชื่อมทุกเส้น เมื่อข่ายงานเริ่มทำงาน จะมีการกำหนดค่าให้แก่บัพอินพุต โดยค่าเหล่านี้ อาจจะได้มาจากการกำหนดโดยมนุษย์ จากเซนเซอร์ที่วัดค่าต่างๆ หรือผลจากโปรแกรมอื่นๆ จากนั้นบัพอินพุต จะส่งค่าที่ได้รับ ไปตามเส้นเชื่อมขาออก โดยที่ค่าที่ส่งออกไปจะถูกคูณกับค่าน้ำหนักของเส้นเชื่อม บัพในชั้นถัดไปจะรับค่า ซึ่งเป็นผลรวมจากบัพต่างๆ แล้วจึงคำนวณผลอย่างง่าย โดยทั่วไปจะใช้ฟังก์ชันซิกมอยด์ (sigmoid function) แล้วส่งค่าไปยังชั้นถัดไป การคำนวณเช่นนี้จะเกิดขึ้นไปเรื่อยๆ ทีละชั้น จนถึงบัพเอาต์พุต

<u>ขั้นตอนที่จะปฏิบัติต่อหลังจากนี้</u>

1. สร้าง Traffic Anomaly Identification Module, Bandwidth Control Module, Log Analyzer Module และ Rule Generator Module
2. เชื่อมโยงการติดต่อระหว่าง Module แต่ละ Module
3. ทดลองและหาระยะเวลาที่เหมาะสมสำหรับการทดสอบประสิทธิภาพในการป้องกันการโจมตีแบบ DDoS ผ่านทาง HTTP Port
4. ทดลองและปรับเปลี่ยน Parameter ของ Module ต่างๆให้ได้ประสิทธิภาพในการป้องกันการโจมตีแบบ DDoS ผ่านทาง HTTP Port สูงที่สุด

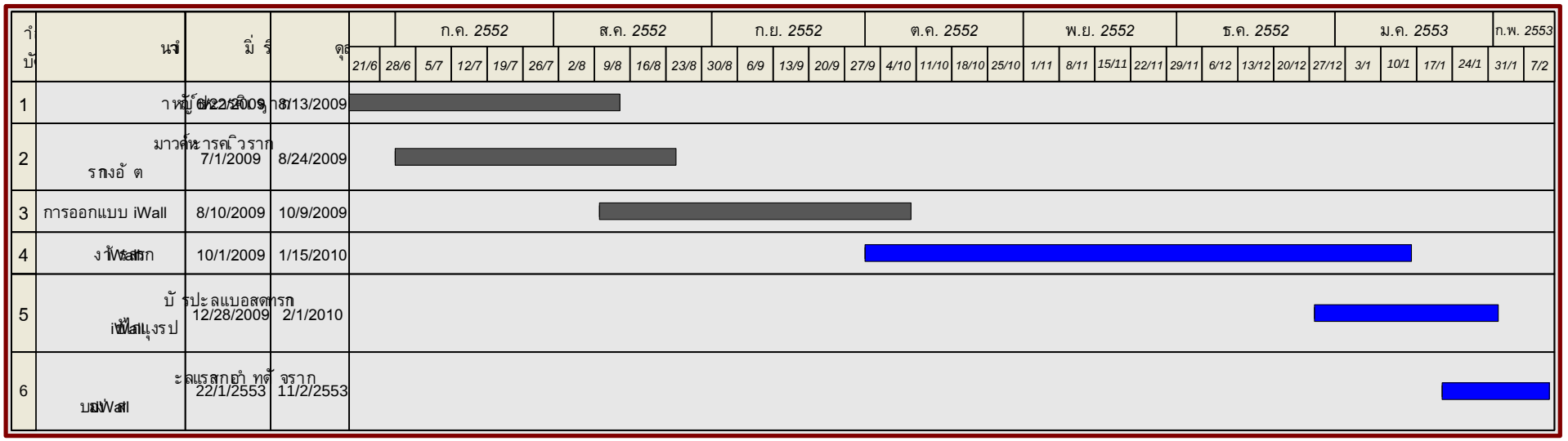
ปัญหาที่พบในระหว่างการทำโครงงาน

1. การกำหนดขอบเขตของโครงงานกว้างเกินไป ทำให้ไม่สามารถทำโครงงานได้เสร็จตามกำหนด แก้ไขโดยการลดขอบเขตของโครงงานให้แคบลงจนเหมาะสมกับกำหนดการ
2. แหล่งข้อมูลการป้องกัน DDoS มักปกปิดข้อมูลบางส่วนเป็นความลับ ทำให้ค้นหาวิธีการป้องกันและเทคนิคต่างๆได้น้อยลง แก้ไขโดยการศึกษาหลายแหล่งข้อมูลแล้วนำมาประยุกต์ใช้ร่วมกัน
3. ชุดข้อมูลทดสอบมาจากแหล่งเดียว ทำให้ไม่สามารถทดสอบการทำงานของ iWall ได้อย่างมีประสิทธิภาพ แก้ไขโดยการค้นหาชุดข้อมูลทดสอบเพิ่มเติมจากแหล่งข้อมูลต่างๆ
4. AWStats มีข้อจำกัด ทำให้การป้องกัน DDoS ยังมีข้อจำกัดอยู่ แก้ไขโดยการศึกษาข้อจัดกัดนั้น และปรับเปลี่ยนหรือเพิ่มเติมการทำงานในบางส่วนให้ดีขึ้น
5. Attribute บางอย่างที่ได้จาก AWStats ไม่ตรงตามรูปแบบที่ต้องการ ทำให้ไม่สามารถดึงไปใช้ในการทำ Data Mining ได้ แก้ไขโดยการเปลี่ยนรูปแบบของข้อมูลให้เหมาะสมต่อการทำ Data Mining

การเปลี่ยนแผนการดำเนินงานและขอบเขต

แผนการดำเนินงาน

การดำเนินงานเป็นไปตามแผนการดำเนินงานเดิมที่ได้เสนอไปในข้อเสนอโครงงาน ซึ่งในขณะนี้กำลังดำเนินงานอยู่ในขั้นการสร้าง iWall



รูป 7 แผนภาพแสดงขั้นตอนและระยะเวลาในการดำเนินงาน (Gantt chart)

ขอบเขต

ขอบเขตของโครงงานมีการเปลี่ยนแปลงโดยมีการกำหนดขอบเขตให้ชัดเจนขึ้นดังนี้

โครงการ “ไฟร์วอลล์ชนิดปรับตัวตามรูปแบบการโจมตี (iWall)” เป็นโครงการที่พัฒนาขึ้นเพื่อให้ไฟร์วอลล์ของระบบมีความมั่นคงและปลอดภัยมากยิ่งขึ้น โดยมีขอบเขตการทำงานดังนี้

1.iWall สนใจ Inbound Traffic ที่ HTTP Port เท่านั้น

2.iWall สนใจการโจมตีแบบ Distributed Denial-of-Service (DDoS) เท่านั้น

3.มีการจัดเก็บรายการบันทึก (Log) ข้อมูลที่ผ่านไฟร์วอลล์สำหรับนำมาทำ Data Mining เพื่อหารูปแบบการโจมตี

4.มีการปรับปรุงกฎของไฟร์วอลล์โดยอัตโนมัติ

5.iWall เป็น Application Firewall เนื่องด้วยมีการคัดกรอง Packet จาก Source IP Address และ Content ของแต่ละ Packet ในแง่ของ HTTP Header

6.เกณฑ์การวัดประสิทธิภาพของ iWall คือ อัตราส่วนความสำเร็จในการป้องกันการโจมตีแบบ DDoS ผ่านทาง HTTP Port และระยะเวลาที่ใช้ในการตรวจจับและทำการป้องกันการโจมตีนั้น

การประเมินโครงงาน

การประเมินโครงงานสามารถแบ่งได้เป็น 2 ส่วน

1.ในส่วนของแผนการดำเนินงานพบว่าการดำเนินงานเป็นไปตามที่ได้วางแผนไว้

2.ในส่วนของคุณภาพของโครงงาน ถือว่ามีการพัฒนา เพราะสามารถกำหนดขอบเขตและออกแบบได้ชัดเจนมากขึ้น

บรรณานุกรม

"HTTP headers", [online] http://en.wikipedia.org/wiki/List_of_HTTP_headers, 31 ต.ค. 2552

"baseline algorithm", [online] http://www.monitorware.com/en/workinprogress/baseline-algorithm-for-traffic.php, 31 ต.ค. 2552

"awatats", [online] http://awstats.sourceforge.net/, 1 พ.ย. 2552

"awatats", [online] http://www.nltechno.com/awstats/awstats.pl?config=destailleur.fr, 1 พ.ย. 2552

"iptables", [online] http://linux.about.com/od/commands/l/blcmdl8_iptable.htm, 28 ต.ค. 2552

"iptables", [online] https://help.ubuntu.com/community/IptablesHowTo, 28 ต.ค. 2552

"Artificial Neural Network", [online] http://en.wikipedia.org/wiki/Artificial_neural_network, 2 พ.ย. 2552