รีเวิร์สพร็อกซี่ 2 ขั้นตอนเพื่อลดผลกระทบการโจมตีเว็บไซต์แบบ Slowloris

Slowloris mitigation techniques by double reverse proxy servers

วชิรนันท์ ปุ้ม[1], รสสุคลธ์ สุวรรณกูฏ[2], เพ็ญพักตร์ สืบศรี[3], สุพรรณี เขียวไกร[4], ณรงค์ศักดิ์ สุขมา[5]
[1,2]สาขาวิชาคอมพิวเตอร์ธุรกิจคณะวิทยาการจัดการและเทคโนโลยีสารสนเทศมหาวิทยาลัยนครพนม
[3,4]สาขาวิชาการจัดการคณะวิทยาการจัดการและเทคโนโลยีสารสนเทศมหาวิทยาลัยนครพนม
[5]สาขาวิชาวิศวกรรมเครือข่ายและความมั่นคงปลอดภัยสารสนเทศแขนงความมั่นคงปลอดภัยไซเบอร์
คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร

E-mail: wachiranun@npu.ac.th[1], rossukon.koy@npu.ac.th[2], penpuk@npu.ac.th[3] ,
supunnee.kob@npu.ac.th[4] , narongsak.s@mut.ac.th[5]

บทคัดย่อ

งานวิจัยนี้นำเสนอผลการปรับปรุงเว็บไซต์ให้มีความทนทานต่อการถูกโจมตี Denial of ServiceในรูปแบบSlowloris ที่เกิดจากวิธีส่งคำร้องขอแบบปกติเข้ามาจำนวนมาก แต่มีการหน่วงค่าเวลาในการรับข้อมูลไว้เพื่อจุดประสงค์ให้เว็บไซต์ปฏิเสธการให้บริการในที่สุด ทีมวิจัยจึงทำการทดลองเพื่อหาทางลดผลกระทบด้วยการใช้เครื่องรีเวิร์สพร็อกซี่จำลองสถานการณ์ส่งคำร้องเพื่อทดสอบและค้นหาขีดจำกัดในแต่ละองค์ประกอบของระบบเว็บไซต์ โดยจากผลวิจัยนี้จะพบว่าการใช้เครื่องรีเวิร์สพร็อกซี่ สามารถช่วยลดผลกระทบจากการโจมตีแบบนี้ได้ดีในระดับที่น่าพอใจทั้งยังช่วยเพิ่มความเร็วให้กับเว็บไซต์ทำให้ตอบสนองดีขึ้น และการทดลองได้แสดงให้เห็นข้อจำกัดต่างๆอันเป็นประโยชน์ต่อผู้ดูแลระบบสามารถนำไปวางแผนระบบตั้งแต่ขั้นตอนออกแบบเว็บไซต์ รวมถึงโครงสร้างพื้นฐานสารสนเทศที่จำเป็นเพื่อนำไปปรับใช้ป้องกันการโจมตีแบบ DDOS ที่มีจุดประสงค์ให้ระบบปฏิเสธการเข้าถึง หรือหยุดการให้บริการต่อไปได้

คำสำคัญ: เว็บไซต์, การโจมตีให้ระบบหยุดบริการ , ระบบฐานข้อมูล , การรักษาความปลอดภัยเว็บไซต์

Abstract

This paper describes Slowloris mitigation techniques that is a DDOS attack type which hacker often like used to attack web services with minimal bandwidth. This research has major implemented base on open-source solution. By paper shows prevention value nearly get from the high-end vendors' security devices. In previous test time, Web services was downed in 10 seconds but after we applied solution as this paper, Web services passed though test period 10 minutes. Also, double reverse proxy servers in this Lab return well web response times for first minute reach to test finished too. System administrator can keep these practices for internal design IT infrastructure for ensure web servers tiers can support high workload as well.

Keywords:
computer security, slowloris, ddos , security network , cybersecurity

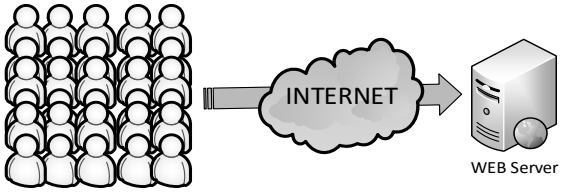
1. คำนำ

ในปัจจุบันนี้เว็บไซต์คือช่องทางหลักในการให้บริการและสื่อสารของทุกหน่วยงานทั้งภาครัฐ และ เอกชน ด้วยความเป็นที่นิยมและสะดวกในการใช้สื่อสารข้อมูลเพราะสามารถเข้าถึงได้จากอุปกรณ์ต่าง ๆที่ต่อกับอินเตอร์เน็ตได้จากทุกที่ทุกเวลา แต่ในขณะเดียวกันด้วยความง่ายในการเข้าถึงนี้เองทำให้มีกลุ่มผู้ไม่ประสงค์ดีพยายามเข้าถึงระบบโดยไม่ได้รับอนุญาต รวมถึงการโจมตีให้ระบบหยุดให้บริการด้วยวิธีการต่าง ๆ อันส่งผลต่อภาพลักษณ์ของหน่วยงาน ซึ่งอาจมีจุดประสงค์แตกต่างกันไปตามเวลา และสถานการณ์ เช่น เพื่อหวังผลทางการเมือง เพื่อหวังผลทางธุรกิจ เพื่อข่มขู่เรียกค่าไถ่หรือเพื่อแอบซ่อนการโจมตีที่แท้จริง [1] อีกทั้งจากการศึกษาสถิติภัยคุกคามผ่านช่องทางอินเตอร์เน็ตที่ได้มีการบันทึกนับตั้งแต่ปี ค.ศ. 1980 [2]จะพบว่าจำนวนการคุกคามในทุกรูปแบบมีแนวโน้มแต่จะเพิ่มขึ้นอย่างเนื่องทุกปี

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

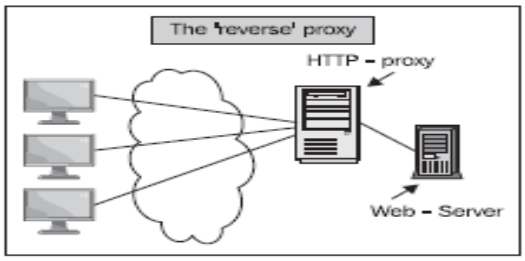
2.1 โครงสร้างระบบบริการเว็บไซต์

ระบบเดิมไม่ได้มีการออกแบบให้ระบบมีการป้องกันการโจมตีเว็บไซต์ ตามรูปที่ 1 โดยระบบเดิมจัดตั้ง web server ให้มีการติดต่อกับอินเตอร์เน็ตโดยตรงจึงอาจทำให้ระบบมีความเสี่ยงที่จะโดนคุกคามหรือถูกโจมตีให้ระบบหยุดให้บริการ รวมถึงมีข้อจำกัดในการให้บริการเมื่อมีการเข้าถึงพร้อมกันจากผู้ใช้งานปริมาณมากในช่วงเวลาเดียวกัน เช่น ช่วงเวลาประกาศผลสอบนักศึกษา จนส่งผลให้ระบบช้าลง



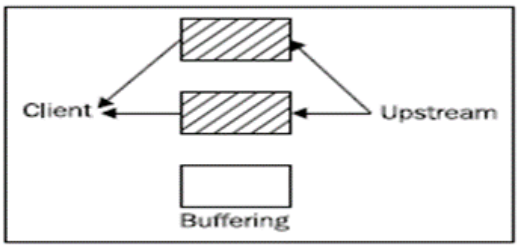
รูปที่ 1 โครงสร้างระบบแบบเดิม

ระบบเว็บไซต์จะมีการตอบสนองช้าจนถึงใช้งานไม่ได้ในบางเวลาของวันดังกล่าว เป็นต้น งานวิจัยนี้ได้ทำการศึกษาเพื่อหาวิธีทำให้ระบบสามารถรองรับการเข้าถึงพร้อมกันจากผู้ใช้งานปริมาณมากในช่วงเวลาเดียวกัน รวมถึงยังสามารถป้องกันการโจมตีเว็บไซต์ที่จะทำให้ระบบปฏิเสธการเข้าถึงหรือหยุดการให้บริการ[3-4] โดยแนวทางวิจัยคือต้องเป็นวิธีการที่เข้าไปยุ่งเกี่ยวกับเครือข่ายที่ให้บริการอยู่ในปัจจุบันให้น้อยที่สุด



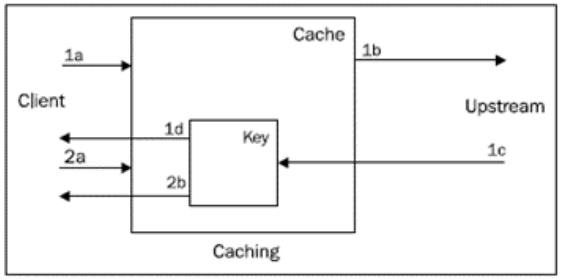
รูปที่ 2 การทำงานของ Reverse proxy [5]

ดังนั้นผู้วิจัยจึงเลือกที่จะใช้ NGINX [5-6] มาใช้ในงานวิจัยเพื่อติดตั้งในรูปแบบ Reverse proxy ตามรูปที่ 2 โดย NGINX นั้นมีความน่าเชื่อและนิยมมากขึ้นอย่างต่อเนื่อง จากผลสำรวจของ Netcraft [7] NGINXได้ถูกนำมาใช้อย่างแพร่หลายในการนำมาทำเป็น Reverse proxy เพื่อช่วยให้ระบบที่อยู่ด้านหลังของ NGINX สามารถรองรับปริมาณ Transaction ที่เพิ่มขึ้น เช่น Buffering , Caching ,compressing



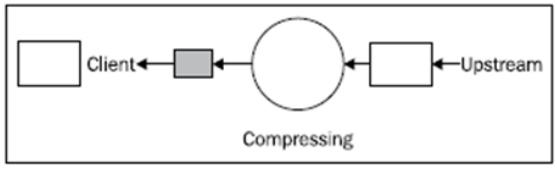
รูปที่ 3 การ Buffering ข้อมูลของ NGINX [8]

การทำงานของ NGINX จะเริ่มจากมีผู้ใช้งานส่งคำร้องขอเข้ามาในระบบ NGINXจะอ่านข้อมูล



รูปที่ 4 การ Caching ข้อมูลของ NGINX

จาก upstream server หรือ web server และเก็บไว้ใน buffer ตัวเองก่อนตามรูปที่ 3 จากนั้นจึงส่งผลนั้นไปที่ผู้ใช้งาน และหากมีการส่งคำร้องขอซ้ำเข้ามาในระบบอีก NGINX จะตรวจสอบว่ามีข้อมูลนั้นใน buffer หรือยัง และหากมีแล้ว NGINX จะส่งข้อมูลเดิมที่มีใน buffer กลับไปให้ผู้ร้องขอทันทีโดยไม่ต้องไปอ่านซ้ำอีกที่ Upstream server โดยยิ่งมีการร้องขอเข้ามาซ้ำ ๆบ่อย ๆในข้อมูลเดิม NGINX จะมี caching ตามรูปที่ 4 ที่จะเก็บข้อมูลนั้นไว้ใน MEMORY อีกทั้งยังมีการใช้ compressing ตามรูปที่ 5 ทำการบีบอัดข้อมูลจากต้นฉบับเดิมที่อ่านมาจาก upstream server ให้เล็กลงก่อนส่งไปให้ผู้ใช้งาน ด้วยการทำงานแบบนี้จึงทำให้การตอบสนองกลับไปยังผู้ใช้งานมีความรวดเร็วที่มากขึ้นกว่าการไม่มี NGINX ทำหน้าที่ reverse proxy นั้นเอง ด้วยค่า buffer พื้นฐานที่มากับ NGINX จะเป็น 4kb หรือ 8Kb ขึ้นอยู่กับระบบปฏิบัติการที่ได้ติดตั้งเมื่อลองคำนวนดูจากค่าbuffer 4Kb คือ 32,768 bytes (8 * 4 * 1024) ต่อ 1 คำร้องขอที่เข้ามาในระบบหาก NGINX server ที่เราใช้งานมีหน่วยความจำ 768 MB ก็จะได้เท่ากับ 805,306,368 bytes (768 * 1024 * 1024) ส่งผลให้หาจำนวน Transaction โดยประมาณที่ระบบสามารถรองรับได้คือ การนำ 32768 มาหารจะได้เท่ากับ 805306368 / 32768 = 24576 คำร้องขอ ดังนั้นคงกล่าวได้ว่าจากหน่วยความจำ จำนวน 768 MB แต่ NGINX สามารถรองรับปริมาณคำร้องได้ถึงประมาณ 25,000 ที่เป็น Active connection [8]



รูปที่ 5 การ Compressing ข้อมูลของ NGINX

2.2 รายการซอฟต์แวร์

รายการซอฟต์แวร์แสดงในตารางที่ 1 ทางนักวิจัยเลือกใช้ซอฟต์แวร์ส่วนใหญ่เป็นซอฟต์แวร์โอเพนซอร์ส คือซอฟต์แวร์ที่เผยแพร่ด้วยสัญญาอนุญาต (license) ที่ประกันสิทธิในการศึกษา เผยแพร่ แก้ไข และใช้งานซอฟต์แวร์ได้อย่างอิสระ [9]

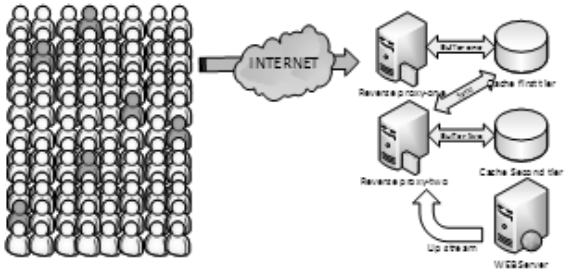
server	software /License type	OS name /License type
NGINX proxy server one	nginx 1.6.2 /2-clause BSD-like license	UBUNTU 15.04.1 /GPLv2
NGINX proxy server two	nginx 1.6.2 /2-clause BSD-like license	UBUNTU 15.04.1 /GPLv2
Web server	apache 2.4.10 / Apache License,V 2	UBUNTU 15.04.1 /GPLv2
TEST Laptop	Webserver Stress Tool 8.0 /Freeware	window 7 / NPU license

ตารางที่ 1 รายการซอฟต์แวร์

2.3 ปรับปรุงโครงสร้างเพื่อเพิ่มประสิทธิภาพระบบ

ผู้วิจัยได้ทำการวิเคราะห์การทำงานของ Reverse proxy ในด้านประสิทธิภาพของ Reverse proxy แบบปกติทำให้ระบบเร็วขึ้นจริง แต่ยังมีช่วงที่ระบบมีการตอบสนองช้าในช่วงแรก จนถึงจุดที่ buffer หรือ caching ทำงานระบบจึงจะทำงานเร็วขึ้น ทางผู้วิจัยจึงนำข้อสงสัยนี้มาทำการทดลองว่าหากเราสามารถลดเวลาในการอ่านข้อมูลตั้งแต่ครั้งแรกที่ NGINX ต้องไปอ่านจาก upstream server ได้หรือไม่

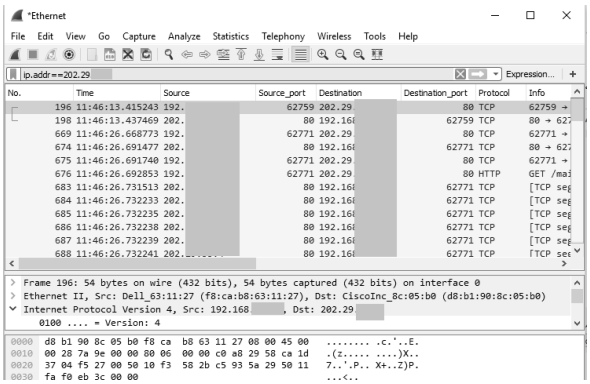
ดังนั้นผู้วิจัยจึงได้ทำการออกแบบให้มีระบบมี reverse proxy 2 ขั้นตอน ตามรูปที่ 6



รูปที่ 6 โครงสร้างระบบแบบใหม่

2.4 ปรับปรุงเพื่อป้องกันการโจมตี

ผู้วิจัยได้มุ่งเป้าวิจัยไปที่การจำกัดจำนวนคำร้องขอที่เกิดจากผู้ใช้งานในเวลาปกติ เพื่อให้มั่นใจว่าการปรับเพิ่มระบบป้องกันนี้ ไม่มีผลกระทบต่อการใช้งานปกติของผู้ใช้ทั่วไปด้วยวิธีการใช้เครื่องมือ Wireshark [10] ซึ่งเป็นเครื่องมือที่ได้รับการยอมรับโดยทั่วไป เพื่อหาจำนวนคำร้องขอเพื่อเข้าถึงเวบไซต์จากการใช้งานโดยปกติของผู้ใช้งานคอมพิวเตอร์ 1 เครื่องในการเข้าถึงเวบไซต์มหาวิทยาลัยนครพนม



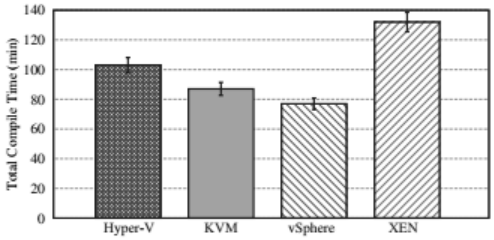
รูปที่ 7 ตรวจจับด้วย Wireshark

จากการตรวจจับด้วยWireshark ทีมผู้วิจัยพบว่าจำนวนสูงสุดที่สามารถเกิดขึ้นได้ภายใน 30 นาทีคือ 100 คำร้องขอ ตามรูปที่ 7

3.การทดลอง

3.1 รายการอุปกรณ์ทดสอบและการออกแบบการทดลอง

การทดลองนี้กระทำในสภาพแวดล้อมเครือข่ายจำลองโดยการเลือกใช้ hypervisors สร้างเครื่องแม่ข่ายจำลอง 3 เครื่องด้วย vSphere ESXi 6.0 [11]โดยเลือกจากการเปรียบเทียบประสิทธิภาพของ Total compile workloads ที่ใช้เวลาน้อยที่สุด[12]



รูปที่ 8 Total Compile Time (min) [11]

3.2 การทดสอบความสามารถในการป้องกันผลกระทบการโจมตีเว็บไซต์

ผู้วิจัยได้ใช้ความสามารถของ NGINX ในส่วนของการจำกัดปริมาณคำร้องมากสุดที่จะเข้าถึงเวปไซต์จำนวน 100 คำร้องขอ ภายในเวลา 30 นาที ที่มาจากผู้ใช้งานเดียวกัน เพื่อจำกัดการเข้าถึงที่ผิดปกติ เพื่อป้องกันการโจมตีเว็บไซต์แบบ Slowloris ด้วย NGINXตามรูปที่ 9

```
# these limits apply to the whole virtual server
limit_conn ips 10;
 # only 1000 simultaneous connections to the same server_name
limit_conn servers 1000;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
client_max_body_size 10m;
client_body_buffer_size 2m;
client_body_in_single_buffer on;
# proxy buffers
proxy_buffers 8 24k;
proxy_buffer_size 2k;
proxy_connect_timeout 90;
proxy_send_timeout 90;
proxy_read_timeout 90;
proxy_set_header Host $host;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_pass   http://myapp;
proxy_cache_valid 200 302 60m;
proxy_cache_valid 404 1m;
 ##### # Configuring Timeouts  #####
send_timeout 30;
open_file_cache max=1000 inactive=20s;
#####caching of static content
open_file_cache_valid 60s;
open_file_cache_min_uses 5;
open_file_cache_errors off;
```

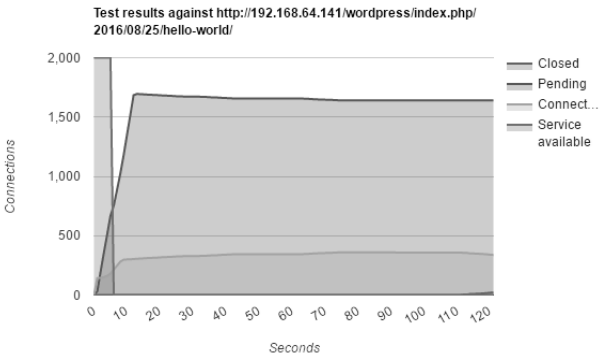
รูปที่ 9 พารามิเตอร์ที่เพิ่มใน NGINX

4. ลำดับและขั้นตอนการทดลอง

ผู้วิจัยได้ออกแบบการทดลองส่งคำร้องขอในรูปแบบ Slowloris ด้วย slowhttptest [13]

4.1 แบบเดิมที่ไม่มี reverse proxy

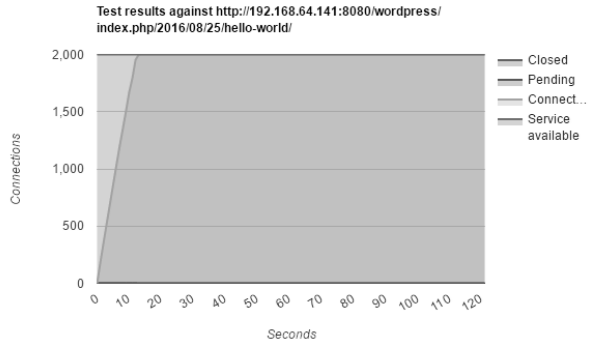
แบบเดิมที่ไม่มี reverse proxy วางด้านหน้า Web server แสดงผลการทดลองตามรูปที่ 10



รูปที่ 10 เวบไซต์หยุดให้บริการเมื่อวินาทีที่10

4.2 แบบที่มี reverse proxy

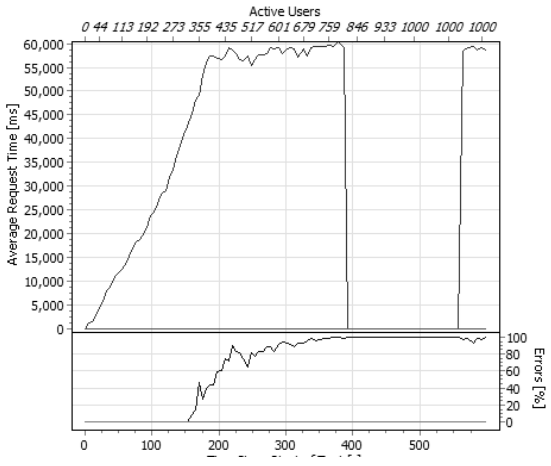
แบบที่มี reverse proxy ด้านหน้าWeb server พร้อมมีการจำกัดการเข้าถึงแสดงผลการทดลองตามรูปที่ 11



รูปที่ 11 เวบไซต์ถูกโจมตีแต่กลับมาให้บริการตามปกติ

4.3 ทดสอบอัตราการตอบสนองของระบบที่ไม่มีรีเวิร์สพร็อกซี่

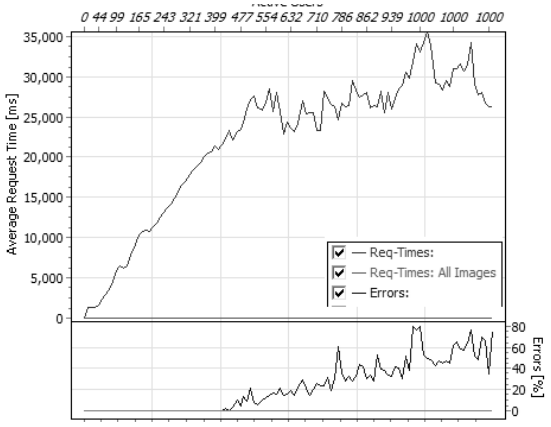
พบว่าระบบจะหยุดให้บริการตั้งแต่วินาทีที่ 400 และไม่สามารถให้บริการได้จนหมดเวลาทดลอง



รูปที่ 12 ระบบไม่มีรีเวิร์สพร็อกซี่

4.4 ทดสอบอัตราการตอบสนองของระบบระบบที่มีรีเวิร์สพร็อกซี่

ระบบสามารถให้บริการได้อย่างต่อเนื่องและรักษาเสถียรภาพของการให้บริการที่ดีจนหมดเวลาทดลอง



รูปที่ 13 ระบบที่มีรีเวิร์สพร็อกซี่

5. สรุป

กลุ่มแรก คือ กลุ่ม Web server ที่ไม่มีเครื่องรีเวิร์สพร็อกซี่ จะมีปัญหาในการตอบสนองจากการโจมตีตั้งแต่ช่วงแรกจนหยุดให้บริการไปในที่สุด กลุ่มที่สอง คือ กลุ่มที่มีเครื่องรีเวิร์สพร็อกซี่ และระบบการป้องกันพบว่ามีอัตราการตอบสนองที่ดี และไม่พบอาการช้าที่เกิดจากปริมาณคำร้องขอที่สะสมเพิ่มมากขึ้น อีกทั้งระบบ

ยังคงสามารถให้บริการได้ต่อไปจนจบการทดลอง ดังนั้นจากผลการทดลองพบว่าการใช้เครื่องรีเวิร์สพร็อกซี่ นั้นให้ผลการตอบสนองที่ดีขึ้นอีกทั้งยังช่วยทำให้ระบบสามารถเพิ่มความทนทานของระบบจากการถูกโจมตีเว็บไซต์แบบ Slowlorisได้อย่างดี

ในอนาคตต้องการศึกษาวิจัยในด้าน Cyber security รวมถึงผลกระทบจากปัจจัยเกี่ยวข้องด้านประสิทธิภาพอื่น การโจมตีแบบ DDOSรูปแบบอื่น ชนิดของหน่วยประมวลผล ขนาดหน่วยความจำ ชนิดของหน่วยเก็บข้อมูล รวมถึงความเร็วในการเขียนอ่านข้อมูล เพื่อให้ครอบคลุมความต้องการผู้ดูแลระบบต่อไป

เอกสารอ้างอิง

[1] P. Sim, T. Cruz, J. Proen, and E. Monteiro, Cyber Security: Analytics, Technology and Automation, vol. 78. 2015.

[2] Joseph Migga Kizza,Computer ,Network Security and Cyber Ethics FOURTH EDITION, 2014

[3] S. a Crosby and D. S. Wallach, “Denial of Service via Algorithmic Complexity Attacks,” Usenix Secur. 2003, pp. 29–44, 2003.

[4] R. Abramov and A. Herzberg, “TCP Ack storm DoS attacks,” Comput. Secur., vol. 33, pp. 12–27, 2013.

[5] D. Sarkar, Nginx 1 Web Server Implementation Cookbook. 2011.

[6] W. U. Nginx and L. Apache, “Nginx : the High-Performance Web Server and Reverse Proxy What should Canonical have named,” Most, pp. 16–19, 2011.

[7] Netcraft, “Web Server Survey by Netcraft on 16th September 2015,” Netcraft News, 201. [Online]. Available:http://news.netcraft.com/archives/category/web-server-survey/

[8] D. Aivaliotis, Mastering Nginx. 2013.

[9] NECTEC ,งานวิจัยซอฟต์แวร์พื้นฐานและทั่วไป (RDI-1)”โครงการโอเพนซอร์ส,” [Online].Available: http://www.nectec.or.th/rd/rd-opensource-th.html

[10] R. Shimonski, The Wireshark Field Guide: Analyzing and Troubleshooting Network Traffic. 2013.

[11] “VMware vSphere Hypervisor 6.0 Download Center,” (n.d.) [Online]. Available:https://my.vmware.com/en/web/vmware/evalcenter?p=free-esxi6

[12] J. Hwang, S. Zeng, and T. Wood, “A component-based performance comparison of four hypervisors,” Integr. Netw. Manag. …, pp. 269–276, 2013.

[13] Sergey Shekyan, “slowhttptest,” May 3 ,2016. [Online]. Available: https://github.com/shekyan/slowhttptest/wiki