I6-FPS: Automating the ICMPv6 Filtering Rules

Wan Nor Ashiqin Wan Ali[1,*], *Abidah* Mat Taib [2], and *Syed Zulkarnain* Syed Idrus[1]

[1]School of Human Development and Techno-Communication (iKOM), Universiti Malaysia Perlis (UniMAP), Malaysia
[1]Centre of Excellence Geopolymer and Green Technology (CEGeoGTech), Universiti Malaysia Perlis (UniMAP), Malaysia DP Sciences, Editorial Department, 91944 Les Ulis Cedex A, France
[2]Faculty of Computer & Mathematical Sciences, Universiti MARA, Malaysia

Abstract. Enterprises are required to utilize Internet Control Message Protocol version 6 (ICMPv6) when IPv6 is deployed. In IPv4, Internet Control Message Protocol (ICMP) is aggressively filtered by a network administrator while in IPv6, ICMPv6 messages cannot be aggressively filtered due to the function of ICMPv6 message. ICMPv6 security risks increase when ICMPv6 threats and vulnerabilities are exploited. Thus, it is very crucial for enterprises to address the issues. In practice, network researchers must review several resources to identify ICMPv6 related attacks occurring due to the exploitation of ICMPv6 vulnerabilities. Overlooking any of these issues will jeopardize the security of ICMPv6. While conducting the attack scenarios testing, IPv6-Filtering Prototype System (I6-FPS) was developed to overcome the deficiency and limited filtering tools that supported IPv6 filtering rules (ip6table). I6-FPS is used to automate and simplify the writing of ip6table and it was developed using PHP5 and Shell script languages. This research revealed that I6-FPS is significant in the initial phase of securing IPv6 deployment as well as focusing on the ICMPv6 filtering rules. The I6-FPS has the potential to be enhanced and developed over time by including more functions to that system in generating specific filtering ip6table rules.

1 Introduction

In practice, network and security administrators are responsible to strengthen the network security from any intruders. The probability of a network to be intruded is getting higher from the unintended use of IPv6 [1]. In scenarios in which IPv6-enabled devices are deployed on enterprise networks that are intended to be IPv4-only, native IPv6 support and/or IPv6 transition/coexistence technologies could be leveraged by local or remote attackers for a number of (illegitimate) purposes. In general, most of the aforementioned security implications can be mitigated by enforcing security controls on native IPv6 traffic and on IPv4-tunneled IPv6 traffic. Among such controls, is the enforcement of filtering policies to block undesirable traffic is crucial. A prototype is developed to assist the IPv6 beginners and unwell-trained administrators to the IPv6 filtering policies and how to implement those policies into filtering rules.

The IPv6 filtering rules prototype is a part of firewall filtering rules where it helps user to generate and activate the filtering rules. User can study on how IPv6 filtering rules work in Linux Operating System. However, having a threat model that only represent on how to mitigate from the identified threats and vulnerabilities is insufficient due to the possibility that unwell-trained administrator and IPv6 beginners are applying inadequate control measure to mitigate ICMPv6 attacks. The basic control measure that can be applied to mitigate the ICMPv6 attacks is by enforcing filtering rules. Nevertheless, the unwell-trained administrator requires more time to learn the filtering rules. Therefore, in facilitating the generation of filtering rules, I6-FPS is developed using the PHP5 and shell script languages.

2 Related Works

An IPv6 filtering rules prototype is proposed while conducting the testing due to the deficiency for available IPv6 filtering tools. The IPv6 filtering rules prototype is a part of firewall filtering rules where it helps user to generate and activate the filtering rules. User can study on how IPv6 filtering rules work in Linux Operating System.

2.1 Firewall Filtering Rules

Firewall is a hardware-based or software-based network equipment to secure network from any threats and attacks. Firewall implements packet filtering, that uses a set of predefined rules, to filter incoming and outgoing packets [2]. Thus, defining of the filtering rules must be coherent to react with the firewall. Moreover, the firewall is one of the vital tools in securing private network, and it is broadly enforced to secure enterprise network as well [3]. Firewall also acts as a fundamental component based on filtering rules to secure network from illegal traffics, as well as threats and attacks [4].

[*] Corresponding author: wannorashiqin@gmail.com

Moreover, the firewall is the current use of security measure, which directly becomes crucial in any applications [5].

They also stated that packet filtering firewall is enforced broadly because it is simple, forwards quicker, more efficient and clear to the users. This packet filtering firewall works based on tuples and does not check data segment in a packet. Furthermore, the packet filter rules inside a firewall are only based on the network security's needs, even though it becomes more and more difficult to manage [5]. Hence, there is a high possibility for mistakes to occur in the enterprise network. The firewall filtering rules must be checked, managed and controlled properly in order to avoid it from making mistakes. Rapid growth of technology in firewall filtering rules has contributed to the development of software-based or hardware-based firewall filtering rules. Firestarter [6],[7] and Firewall Builder [8],[9] are the two well-known software-based firewall filtering rules that are being used.

2.2 Firestarter

Firestarter is considered as a present Linux firewall which simplifies the security management inside Linux OS. Firestarter comes with several beneficial features such as it provides GUI, and it is simple. It is also the appropriate software to be used by desktop, gateway or server, and it allows users to define whitelist, blacklist IP addresses as well as inbound and outbound policy. Not only that, Firestarter provides Internet connection for sharing activity. It is also free, and it is open source software which can be downloaded as an individual or host firewall. Firestarter uses Netfilter (iptables/ipchains) system built-in inside Linux kernel, which provides real-time monitoring for network traffics. The Firestarter provides a functional, secured and simple GUI software-based designed for an advanced firewalling technology [7].

Firestarter is an appropriate tool for a user who needs to learn Linux-based firewall. Firestarter works in a faster way since iptables is quite complex for new beginners in Linux filtering rules. The Firestarter solves the problem for average users to enforce proper firewall filtering rules without taking a longer time in designing complex iptables rules [6]. However, Firestarter does not have a command prompt to manipulate the rules, thus limits users to learn iptables to increase their network security. Hence, there is a suggestion for Firestarter to be enhanced and included with a console version to allow users to learn Linux-server rules as well.

2.3 Firewall Builder

Firewall Builder provides three main features which are simplicity, time saving and flexibility [9]. Firewall builder's features such as search-and-replace, shared objects, drag-and-drop GUI have simplified the task of configuring firewall rules. It also supports several firewall platforms such as Private Internet eXchange (PIX), Berkeley Software Distribution (BSD) packet filter and Cisco ASA which makes it more flexible. Firewall Builder has simple GUI and supporting multiple platforms, which allows users to focus on filtering their traffic instead of wasting time to learn and search for filtering rule commands. However, the Firewall Builder designed for large target user such as organization, academic institution and government. A comparison table was constructed to summarize and compare the Firestarter, Firewall Builder and a designed prototype for this research that is known as I6-FPS shown in Table 1:

Criteria	**Firestarter**	**Firewall Builder**	**Research Design Prototype (I6-FPS)**
Download version	i. Open source software ii. Available free of charge	i. Trial open source for 30 days. ii. Need to buy a license to keep using the software.	i. Open source software ii. Available free of charge
Graphical User Interface (GUI)	i. Easy to use Graphical User Interface (GUI)	i. Easy to use Graphical User Interface (GUI) ii. More complex GUI elements compare to Firestarter	i. Easy to use Graphical User Interface (GUI)
Supporting Operating System	i. Linux	i. Linux ii. Windows	i. Linux ii. Can be accessed in Windows, but rules cannot be activated.
Manageme nt tool supports	i. iptables	i. iptables ii. ip6tables iii. Access Control List (ACL)	i. ip6tables
Target Users	i. Linux user	i. Large organization network	i. IPv6 beginner, enterprises and students

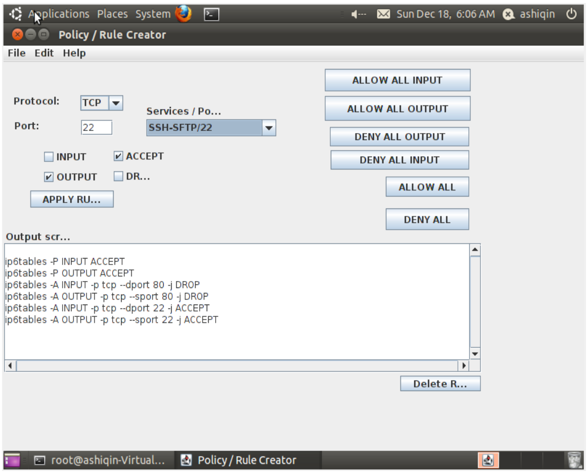
Table 1. Comparison Table (Firestarter, Firewall Builder and Research Designed Prototype)

3 Development of I6-FPS

This section discusses the steps involved in I6-FPS development. This I6-FPS is able to be used in the experimentation phase which indirectly eases the researcher in executing the ICMPv6 rules for each node represented in the testing.

3.1 Planning

In the planning phase, some preliminary studies regarding tools in generating the filtering rules have been conducted. A filtering rules prototype was developed to demonstrate on how filtering policies can be converted into filtering rules. The prototype was developed using JAVA programming languages. Figure 1 shows the standalone prototype (rule creator) that is capable to convert the filtering policies into ip6tables rules. The rule creator is developed to proof the concept on how possible the filtering policies can be converted into filtering rules.



Protocol: TCP
Port: 22
Services / Po... SSH-SFTP/22
□ INPUT ☑ ACCEPT
☑ OUTPUT □ DR...

Output scr...

```
ip6tables -P INPUT ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -A INPUT -p tcp --dport 80 -j DROP
ip6tables -A OUTPUT -p tcp --sport 80 -j DROP
ip6tables -A INPUT -p tcp --dport 22 -j ACCEPT
ip6tables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Fig 1. Filtering Rules Prototype (Rule Creator)

The rule creator as shown in Figure 1 is developed to initiate the development of I6-FPS even though the rule creator cannot activate the filtering rules inside the Linux OS. Each of the tasks that are involved in I6-FPS development is described in details in Context Diagram, Sequence Diagram and Level 0 Diagram.

3.2 Analysis

The analysis phase defines the processes involve in gathering and interpreting facts about the filtering rules, diagnosing the problems of the existing filtering system and recommending the development of I6-FPS. Context Diagram and Level 0 Diagram show the flow in the development of I6-FPS.

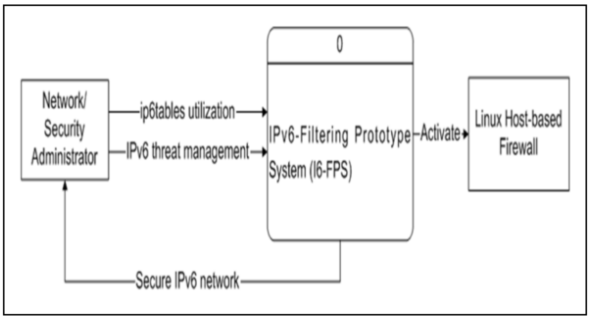


Fig 2. Context Diagram

Referring to Figure 2, there are two entities involved in the system development which are network or security administrator and Linux host-based firewall. The two entities are linked together with the main process known as I6-FPS. The network or security administrator manages the IPv6 threat and utilizes the ip6tables using I6-FPS. Then, I6-FPS activates the Linux host-based firewall for each of the network administrator's nodes. As a result, the network or security administrator can manage their IPv6 network securely. The context diagram has been zoomed in and narrowed down by expanding the main process which is called Level 0.

3.3 Design

The development of I6-FPS is commenced once the required software is installed. A sequence diagram as in Figure 3 was illustrated to show how I6-FPS works.

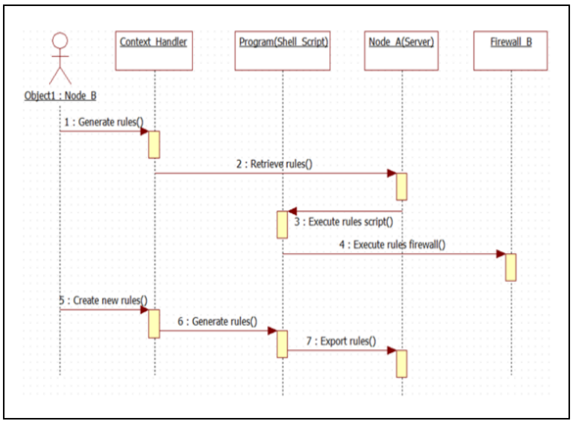


Fig 3. Sequence Diagram of I6-FPS

The object, which is Node_B, will perform two activities such as generating rules and creating new rules. For the generating rules activity, Node_B will generate the rules using the context handler. Then, the context handler will retrieve the rules from the server. The server will execute the rules script to the host firewall. Node_B can also create new rules if there is no existence rule for the current case via the context handler. The rules can be generated by a shell script and then will be exported to the server. The flow of the context handler's functions can be referred to Level 0 diagram.

The Level 0 for I6-FPS development consists of six processes, which have been expanded from the main process as in Figure 4. However, there is still the same number of entity involved. Figure 4 shows the expansion of the main process including; a) Check ip6tables rules, b) Generate ip6tables rules, c) Select ip6tables rules action, d) Check ip6tables status, e) Report an attack, and f) View report.

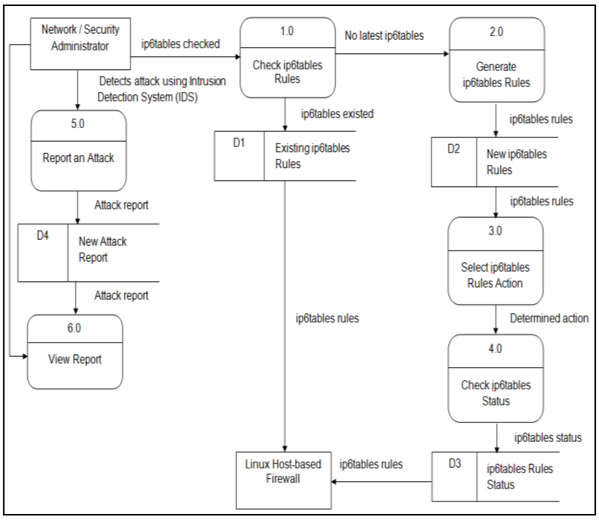


Fig 4. Level 0

3.3.1 *Process 1.0: Check ip6tables Rules*

The entity "Network/Security Administrator" checks whether the computer already has ip6tables or not. If the required ip6tables already existed, user might proceed to Process 2.0 or Process 3.0, but if the required ip6tables is not yet there, the entity will move to Process 2.0.

3.3.2 *Process 2.0: Generate ip6tables Rules*

Process 2.0 produces the new required ip6tables rules. Then, the new ip6tables rules will be stored in the database and the entity "Network/Security Administrator" will move to Process 3.0.

3.3.3 *Process 3.0: Select ip6tables Rules*

After the new required ip6tables rules are stored, then the entity "Network/Security Administrator" will select the action whether to activate or deactivate those ip6tables rules. The selected action will determine whether ip6tables in the entity's "Linux Host-based Firewall" will be activated or deactivated. After that, the entity needs to check the ip6tables rules' status as in Process 4.0.

3.3.4 *Process 4.0: Check ip6tables Rules*

Process 4.0 happens when the entity "Network/Security Administrator" desires to check the ip6tables' status whether the selected action is successful or not.

3.3.5 *Process 5.0: Report an Attack*

The entity "Network/Security Administrator" can report the new attack when the entity finds or detects the attack. Report of the attack will be stored in the database once the entity submits the report.

3.3.6 *Process 6.0: View Attack*

The entity "Network/Security Administrator" can view all the reported attack, and if the entity has any solutions to mitigate the attack, he will generate new ip6tables rules as in Process 2.0. I6-FPS development processes proceed with a flowchart before it moves to the encoding phase.

3.4 Implementation

3.4.1 *Identifying System Requirement*

I6-FPS was developed in Ubuntu Virtual Machine. The Ubuntu Virtual Machine and software needed in the system are installed. The specification of the computer used for the development is Intel(R) Core(TM) i5-2300 CPU @ 2.80GHz. The software installation is done in the root privilege mode in order to ensure that all the software can be installed successfully. There are three steps in completing the software installation for the system development which are installing MySQL 5, installing Apache2, and installing PHP5 [10].

Figure 5 shows the flowchart of I6-FPS. Identically, every system needs a user to log in to authenticate the user of the system. The user will enter the name and password before can use the system. After the user's information has been authenticated by the system, the user can validate the ip6tables inside their machine, checking whether the ip6tables rules exist or not. If no ip6tables rules exist yet, the user will generate the new ip6tables rules using I6-FPS. Finally, if the user does not have anything to do with I6-FPS such as Process 5.0 and Process 6.0 as in Figure 5, the user can log out from the system.

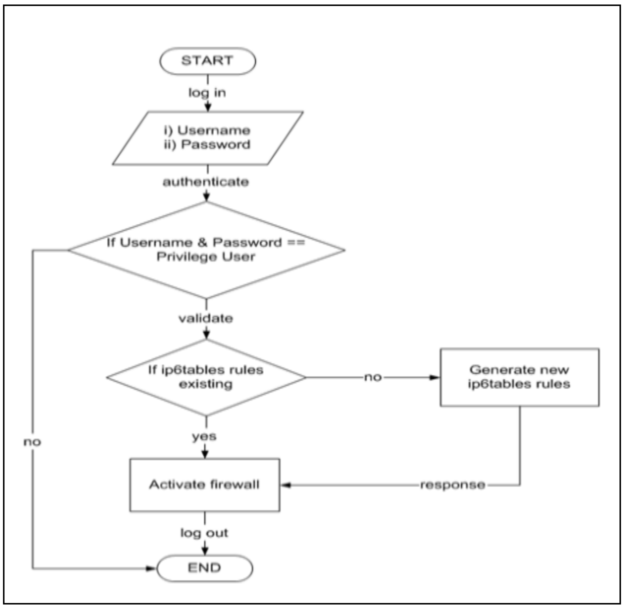
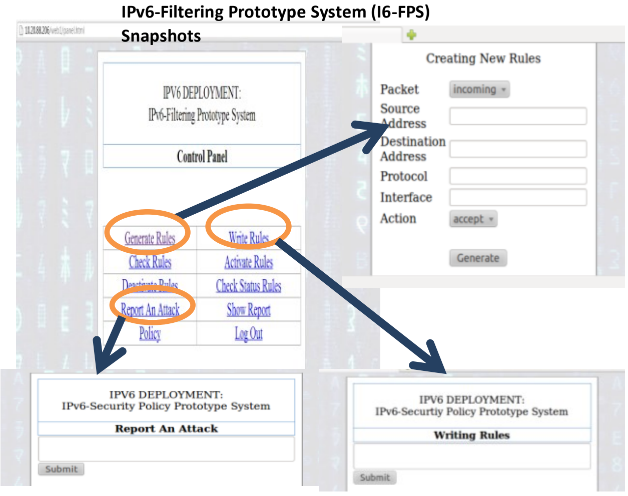


Fig 5. Flowchart of I6-FPS

As has been noted before, I6-FPS is used in the experimentation to ease the user in managing the ip6tables rules efficiently. The system can be accessed through the Local Area Network (LAN). Figure 6(a) and Figure 6(b) show the snapshot of I6-FPS front page.

4 Proposed I6-FPS

Based on Figure 6(a) and Figure 6(b), I6-FPS is used to facilitate the filtering rules generation associated safeguards in the ICMPv6 threat model [11]. The filtering rules used in the testing are translated by adapting the ICMPv6 selective filtering policy [12]. The prototype also facilitates users to write the ip6tables rules with Graphical User Interface (GUI) in the threat experimentation.



IPv6-Filtering Prototype System (I6-FPS)
Snapshots

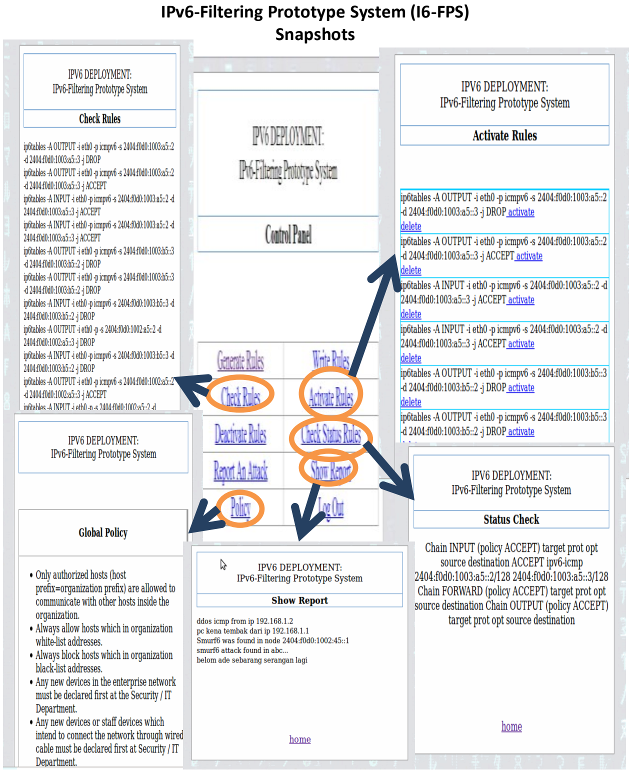
IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Control Panel

Creating New Rules
Packet: incoming
Source Address:
Destination Address:
Protocol:
Interface:
Action: accept

IPV6 DEPLOYMENT:
IPv6-Security Policy Prototype System
Report An Attack

IPV6 DEPLOYMENT:
IPv6-Securtiy Policy Prototype System
Writing Rules

Fig. 6(a) The Snapshots of I6-FPS



IPv6-Filtering Prototype System (I6-FPS)
Snapshots

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Check Rules

```
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j DROP
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP
ip6tables -A OUTPUT -i eth0 -p -s 2404:f0d0:1002:a5::2 -d 2404:f0d0:1002:a5::3 -j DROP
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1002:a5::2 -d 2404:f0d0:1002:a5::3 -j ACCEPT
ip6tables -A INPUT -i eth0 -p -s 2404:f0d0:1002:a5::2 -d
```

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Control Panel

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Activate Rules

```
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j DROP activate
delete
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT activate
delete
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT activate
delete
ip6tables -A INPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:a5::2 -d 2404:f0d0:1003:a5::3 -j ACCEPT activate
delete
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP activate
delete
ip6tables -A OUTPUT -i eth0 -p icmpv6 -s 2404:f0d0:1003:b5::3 -d 2404:f0d0:1003:b5::2 -j DROP activate
```

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Global Policy

- Only authorized hosts (host prefix=organization prefix) are allowed to communicate with other hosts inside the organization.
- Always allow hosts which in organization white-list addresses.
- Always block hosts which in organization black-list addresses.
- Any new devices in the enterprise network must be declared first at the Security / IT Department.
- Any new devices or staff devices which intend to connect the network through wired cable must be declared first at Security / IT Department.

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Show Report

```
ddos icmp from ip 192.168.1.2
pc kena tembak dari ip 192.168.1.1
Smurf6 was found in node 2404:f0d0:1002:45::1
smurf6 attack found in abc...
belom ade sebarang serangan lagi
```

home

IPV6 DEPLOYMENT:
IPv6-Filtering Prototype System
Status Check

```
Chain INPUT (policy ACCEPT) target prot opt source destination ACCEPT ipv6-icmp 2404:f0d0:1003:a5::2/128 2404:f0d0:1003:a5::3/128 Chain FORWARD (policy ACCEPT) target prot opt source destination Chain OUTPUT (policy ACCEPT) target prot opt source destination
```

home

Fig. 6(b) The Snapshots of I6-FPS

5 Conclusions

This research is conducted to automated the writing of the filtering rules (ip6tables) using I6-FPS. The I6-FPS is quite similar to the Firestarter in Linux. Firestarter is used in configuring firewall rules and setting policies for IPv4 only. Thus, this research designed I6-FPS as filtering prototype for IPv6 deployment. Similar to Firestarter, I6-FPS is not a firewall; rather, it is a fronted system to configure rules. Dissimilar to Firestarter, it has been designed to support ip6tables rules. This prototype is also accessible by any users in the network and they do not need to install this prototype in their own machine.

Several attack scenarios testing were conducted to test the associated safeguards for ICMPv6 threats and vulnerabilities using the use of I6-FPS as in [12]. I6-FPS was tested in the attack scenario testing to show that it is capable in generating and activating ip6tables rules. A usability testing was also conducted by distributing questionnaires to assess whether I6-FPS can ease and help the users in writing IPv6 filtering rules (ip6tables).

The I6-FPS is crucial for the enterprise network when deploying IPv6 because it is simple and easy to use for beginners in the enterprise network or individual user to learn firewall scripting (ip6tables rules). The I6-FPS is indirectly contribute in securing the IPv6 deployment. In contrast, without having the I6-FPS, the enterprise networks are exposed and threatened with the potential of ICMPv6 attacks which are probably exploited from ICMPv6 threats and vulnerabilities.

Future works are conducted to enhance the I6-FPS by adding Rule-Based and Case-Based Reasoning approaches. With those approaches, I6-FPS will work more efficient and optimistically will be able to imitate the function of the firewall. Furthermore, the usability testing was conducted if demonstrated that I6-FPS is an easy and simple prototype. Hence, it can be clearly considered that having a GUI prototype system helps users to handle the ip6tables filtering rules.

This research work was supported by Ministry of Education (MOE), Malaysia and Universiti Malaysia Perlis (UniMAP), Malaysia under Research Short Term Grant (STG), Project code: “9001-00527”.

References

1. Gont, T., Liu W. (2014). Security Implications of IPv6 on IPv4 Networks. Technical Report: Internet Engineering Task Force (IETF). ISSN: 2070-1721.
2. Benelbahri, M. A., & Bouhoula, A. (2007). Tuple based approach for anomalies detection within firewall filtering rules. Paper presented at the 12th IEEE Symposium on Computers and Communications (ISCC), 2007.
3. Abbes, T., Bouhoula, A., & Rusinowitch, M. (2008). An inference system for detecting firewall filtering rules anomalies. Paper presented at the Proceedings of the 2008 ACM Symposium on Applied Computing.

4. Benelbahri, M. A., Bouhoula, A., & Trabelsi, Z. (2007). XML based open tool for anomalies detection in firewall filtering rules. Paper presented at the 4th International Conference on Innovations in Information Technology (IIT), 2007.
5. Wang, Y.-G., Ge, Y.-m., & Yang, J.-X. (2008, 12-14 Dec. 2008). Research on packet filter rules of the firewall based on visual prolog. Paper presented at the International Conference on Computer Science and Software Engineering, 2008.
6. Jack, W. (2009). Review: Firestarter firewall for Linux. Retrieved May 12, 2012, from http://www.techrepublic.com/blog/products/review-firestarter-firewall-for-linux/667.
7. Tomas, J. (2004). Firestarter manual. Retrieved October 4, 2012, from http://firestarter.sourceforge.net/manual/introduction.php.
8. Kurland, V. (2010). What's new in Firewall Builder 3.0. Linux J., 2010(189), 4.
9. NetCitadel. (2012). Firewall Builder. from http://www.fwbuilder.org.
10. Timme, F., & Schmalfeld, C. (2011). Installing Apache2 with PHP5 and MySQL support on Ubuntu 11.10 (LAMP). Retrieved October 22 2012, from http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-ubuntu-11.10-lamp.
11. Taib, M. A., Ali, W. N. A. W. & Shaari, N. S. (2013). “ICMPv6 Vulnerability: The Importance of Threat Model and SF-ICMP6”, International Journal of Mobile Computing and Multimedia Communications (IJMCMC), 5(2), 78-100, April-June 2013.
12. Ali, W. N. A. W., Taib, A. H. M., Hussin, N. M., & Othman, J. (2012). IPv6 attack scenarios testbed. Paper presented at the IEEE Symposium on Humanities, Science and Engineering Research (SHUSER), 2012.