OPEN ACCESS
ISSN 2356-5462
socj.telkomuniversity.ac.id/ijoict

Intl. Journal on ICT
Vol. 2, Issue. 2, December 2016. pp. 11-26
doi:10.21108/ijoict.2016.22.77

Improving Network Security - A Comparison between nDPI and L7-Filter

G. B. Satrya [#1], F. E. Nugroho [*2], T. Brotoharsono [#3]

[#] *Telkom Applied Science School, Telkom University*
Telekomunikasi St No.1, Bandung, West Java, Indonesia

[1] gbs@telkomuniversity.ac.id, [3] tri.brotoharsono@telkomuniversity.ac.id

[*] *Forensics and Security Laboratory, Telkom University*
Telekomunikasi St No.1, Bandung, West Java, Indonesia

[2] faizal.eko@gmail.com

Abstract

The classification of data traffic in a firewall using parameters such as port number, IP address, and MAC address is not sufficient. For example, currently, many applications can be used without a port number meaning they can easily circumvent a firewall. Firewalls inspecting up to only layer four could allow malicious data to pass. Next-generation deep packet inspection (DPI) is a method that can be used for firewalls as a method of classification up to layer seven in data traffic control.

This research recommends the use of nDPI and L7-filter by network administrators on existing open source firewalls. Eleven internet applications were used to test and analyze nDPI and L7-filter which are capable of detecting traffic based on the data signature. nDPI and L7-filter were tested for accuracy and speed of total time of execution in live performance networks. We conclude that the development of next-generation deep packet inspection is important for the future of system and network security.

Keywords: firewall, deep packet inspection, nDPI, L7-filter, data signature, accuracy, speed.

I. Introduction

A firewall is sometimes called a packet filter (*pf*) in UNIX/Linux (Rash, 2007) (Ciampa, 2011), CISCO call it an access list (Cisco, 2003), Juniper call it a firewall filter (Garrett, Drenan, & Morris, 2002) and Sonicwall refer to it as access rules (SonicWALL, 2015). A firewall is a system, either hardware or software, which controls the flow of incoming and outgoing data traffic on a network by analyzing network data packets of the traffic and directing that traffic according to a set of rules (Rash, 2007)(Ciampa, 2011). Firewalls are generally used as a barrier between networks that have different security policies. For example, a firewall installed to limit a company's internal network, limits access into the network from outside.

In securing computer networks the first step is to make various security policies that articulate the rules, requirements, standards, and recommendations specific to the environment (Liu et al., 2006) (Casado et al., 2016). The development of network security policy is important and needs to be addressed seriously. Making good decisions at this stage will save finances and prevent future issues over security on the network. Vice versa, making incorrect or hasty decisions would create an insecure network infrastructure. It is important, therefore, for proper controls to be configured correctly in order to provide the optimum level of network security policy for the organization (Liu et al., 2006) (Casado et al., 2016).

Received on Aug 2016. Accepted on Nov 2016

Attacks can be unpredictable and cannot always be avoided (Ciampa, 2011) (Oriyano, 2014). An organization must protect itself by considering its vulnerability, assessing how an attacker could penetrate its defense, and then taking proactive steps to defend against an attack. Ciampa said that to create a secure network it is not always necessary to purchase and install the latest expensive software (Ciampa, 2011). By understanding the system vulnerabilities and threats that might exploit the system good security can be achieved. Some effective solutions are vulnerability scanning and penetration testing (Oriyano, 2014) (G. Satrya & Shin, 2015) (G. B. Satrya, Cahyani, & Andreta, 2015). DPI is a software that is able to perform vulnerability scanning up to layer 7 of the OSI layer (Scarfone & Hoffman, 2009).

DPI is a packet filter which examines the payload of a packet when it passes an inspection point (Scarfone & Hoffman, 2009) (Deri, Martinelli, Bujlow, & Cardigliano, 2014) (Ou, 2013). DPI has been developed by many different vendors. For example Cisco has ASA (Cisco, 2003), Juniper has Screenos (Garrett et al., 2002), and Fortinet has Fotigate (Fabbri & Volpe, 2013). The present research concerns open source DPI known as nDPI (Deri et al., 2014) (Bujlow, Carela-Español, & Barlet-Ros, 2015) (ntop, 2013) and L7-filter (Bober, 2009) (L7-Filter, 2008). By testing the accuracy and speed of nDPI (using Aho-Corasick string matching process) and L7-filter (using regular expression matching process), this research aims to provide recommendation for network and system administrator in optimizing existing open source firewalls. Those optimization can be carried out by modifying the kernel module on Linux (especially using iptables), using nDPI and L7-filter as stateless firewalls. Firewalls were tested using eleven well used internet applications including BitTorrent, eDonkey, Skype, Yahoo Messenger, Facebook, Twitter, Google, and YouTube. In terms of accuracy, which is measured by specificity and sensitivity, nDPI can provide better performance than L7-filter in live performance testing.

The remainder of the paper is organized as follows. Section II explains the related work and theory of open source firewalls. Section III describes system design and scenarios for assessment. Section IV contains the firewall assessment with eleven applications used for evaluation and analysis. Section V presents the results and discussion. Finally, Section VI presents the conclusions and potential future work from this paper.

II. Related Work & Literature

A. Generation of Firewall

The development of firewalls can be divided into three generations (Rash, 2007) (Ciampa, 2011) (Scarfone & Hoffman, 2009). In the early generations, starting in the 1980s, firewalls provided filtering of packet data based on criteria such as port, protocol, IP address, and MAC address. In the 1990s, appeared stateful packet inspection firewalls, which could classify the traffic based on the existing state of incoming or outgoing traffic compared with the table state. In that generation, a firewall worked on the OSI model layers 2, 3, and 4. In the current third generation, since around 2000, the firewall has more capabilities with a broader scope, ranging from deep packet inspection (DPI) of all data traffic, intrusion prevention, malware detection, traffic analysis, application control, IPSec and SSL VPN. In this generation, the firewall is working on a model OSI layers 2, 3, 4, and 7, which are not only able to defend the system from attack, but also the content filtering of data traffic (Ayoub, 2012).

B. Deep Packet Inspection

Deri, Luca et al(Deri et al., 2014), states that commercial firewalls that support DPI libraries are very expensive. Therefore, using Linux-based firewalls with iptables or netfilter is more cost effective. Netfilter is a packet filtering framework that has existed since Linux kernel 2.4, with iptables as a front end feature for users. Iptables is a user-space application program which gives commands to the kernel to perform certain actions on incoming, passing and outgoing data packets of the Linux operating system. Iptables is not only used for controlling packet access or filtering data packets, but can also be used for packet manipulation (mangling), and network address translation (NAT), which is the translation of an IP address to another IP address.

Deep packet inspection (DPI) firewalls, also called as application firewall, is a firewall that works using deep packet inspection methods to examine data packet (Gheorghe, 2006) (Bujlow, Carela-Español, & Barlet-Ros, 2014). The firewall will examine and analyze the protocol in the data packet up to the

application layer of OSI model (Deri et al., 2014) (Radisys, 2010) (Thomason, 2012). This type of firewall can be used to stop or allow access to applications running in a network based on the type of the application (Bujlow, Carela-Español, & Barlet-Ros, 2013). For example, it is used to stop instant messaging application even though it was still running on port 80. In addition, the firewall with this method can be used to identify denial of service attacks and malware worms (Radisys, 2010) (Thomason, 2012). Application of DPI on the firewall has also been used in artificial firewall from various companies such as Cisco, SonicWall, Juniper, and Fortinet.

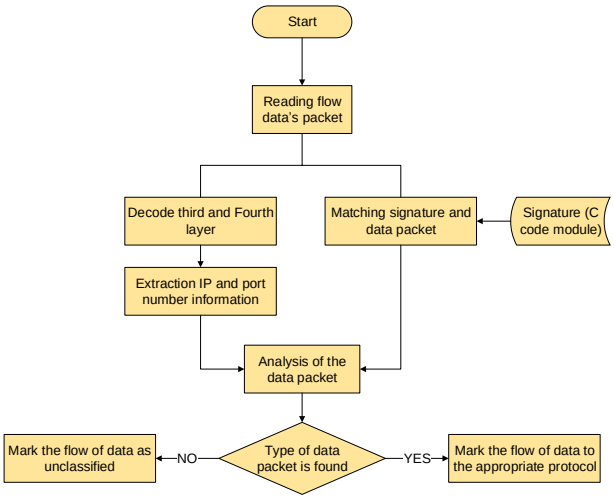


Fig. 1: Execution Process of nDPI

1) nDPI: nDPI is a DPI library that was developed by Ntop (ntop, 2013). nDPI was developed from OpenDPI, an open source version of DPI library created by Ipoque for firewalls. As stated by Deri, Luca et al (Deri et al., 2014)(Bujlow et al., 2015), that DPI was developed by creating an efficient DPI library and supports a high number of protocols. nDPI also has an influence on performance DPI protocol detection and recognition. nDPI uses an Aho-Corasick string matching algorithm to process data packets matching the signature owned by nDPI (Rash, 2007). In Figure 3 below is a flowchart of the L7-filter. Some nDPI capabilities include (ntop, 2013):

- Facilitating the detection of more than 170 protocols
- Having a decoder of SSL certificates, so that it can facilitate the detection on an encrypted connection
- Having the ability to support sub-protocol using a string-based matching process.

nDPI was created for use by applications that require application protocol detection on the flow of communication (see Figure 1). It focuses on Internet traffic, where all available dissectors support standard protocols (such as HTTP and SMTP) and is popular among the Internet community (such as Torrent and Skype). For these nDPI has supported dissection by reverse-engineering the network traffic. Although nDPI can extract specific metadata (such as HTTP URL) of the traffic analyzed, it is not designed for tasks such as lawful interception or a data leak prevention. The main purpose of nDPI is to characterize network traffic.

2) L7-filter: L7-filter is a packet classifier on Linux, originally created as an iptables additional module for detecting data traffic on peer-to-peer sharing applications. This library uses a regular expression matching process (see Figure 2) for matching the protocol used with the data at the application layer in the OSI model (Bober, 2009) (L7-Filter, 2008). L7-filter works best when mounted on the mangle table, chain POSTROUTING in netfilter/iptables (Bober, 2009) (L7-Filter, 2008).

L7-Filter's development began in 2003, due to the majority of applications for controlling the bandwidth used for protocol being expensive. Then, in 2005, developed versions were combined with the Linux kernel and Netfilter. After that, in 2006, a version was developed that could run without having to be combined with the Linux kernel. At present, the development of L7-Filter has been taken over by ClearFoundation, and the latest version of the L7-Filter is version 2:23, which was released in September 2013 (L7-Filter, 2008) (Wildgoose, 2015). In Figure 2 below is a flowchart of the L7-filter.

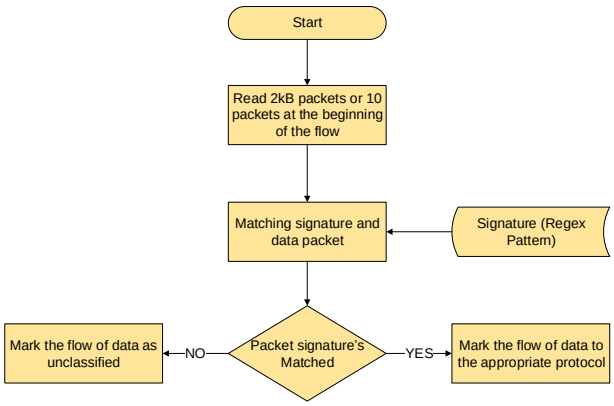


Fig. 2: Execution Process of L7-Filter

III. System Design

A. *Proposed Module*

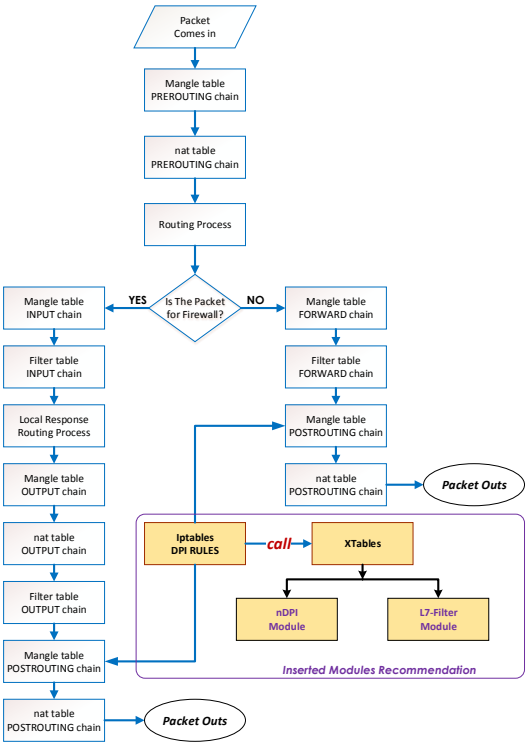


Fig. 3: Flow of data traffic in the netfilter

Netfilter/iptables consists of different tables. For example; filter, NAT, and mangle. Each table has a set of rules incorporated in particular chains. Rules which are in chains on a table will be used to manipulate the data packets that pass through the table and chain, if the results of packet analysis by the netfilter is in accordance with the rule. The rule itself is executed by using the module iptables in the kernel, namely XTables, then looking at the XTables module associated with the rule and running it. In Figure 3 is a modification of the module XTable on iptables. As explained in DPI sub section, that Deri (Deri et al., 2014) and Bujlow (Bujlow et al., 2015) have developed DPI called nDPI at the application layer. Testing the accuracy and speed of nDPI and L7-filter adds to the contribution of this research in order to

make netfilter/iptables, that previously did not support DPI analysis, to be able to support the inspection up to the application layer as depicted in Figure 3. In light of these recommendations, the network and system administrators will be able to optimize the existing open source firewalls through modifications in the kernel module on Linux.

The difference between our research and Deri's (Deri et al., 2014) and Bujlow's (Bujlow et al., 2015) are as follow: First, the data retrieval processes were carried out using live performance testing. In this case, every firewall was tested using real time data traffic where as the two authors (Deri et al., 2014)(Bujlow et al., 2015) considered the result of replay of saved traffic. Second, the scenarios in our experiments have been built in such a way that the actions taken in each scenario were the same in terms of activities such as accessed links, total time for execution, and the amount of tests in a scenario. On the other hand, Deri's (Deri et al., 2014) and Bujlow's (Bujlow et al., 2015) did not compute the total time for execution of the firewalls.

Based on Figure 3 above, the research was conducted by two stages. The first was to insert the module nDPI XTables in the netfilter package. The second was to insert L7-filter module in XTables on the netfilter package. Iptables can not perform deep packet inspection. So this research combines the capabilities of nDPI and of L7-filter on the XTables module in the netfilter package. By testing separately between nDPI with iptables and L7-filter with iptables, sensitivity and specificity testing was carried out on eleven applications that are described in section IV.

B. Network Configuration

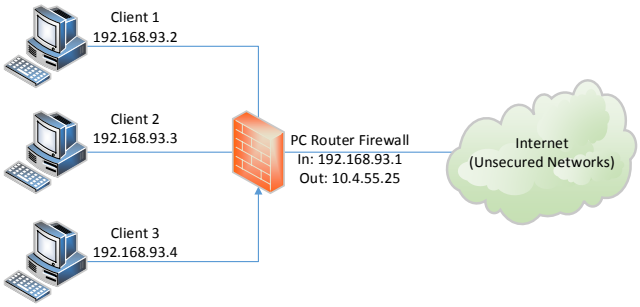


Fig. 4: Network topology

The network used in the testing of nDPI firewall and L7-filter firewall consisted of a firewall and three virtual computers connected to the firewall. nDPI firewall and L7-filter firewall were used interchangeably as corresponding test scenarios were run. A personal computer with an Intel Core i7-3770 3.40Ghz, 4GB RAM and a 25GB hard disk was used to create four virtual computers with a minimum of 512MB of RAM each. The virtual computers were named Client 1, Client 2, Client 3 and PC Router Firewall (see Figure 4). Netfilter/iptables version 1.4.4 was used as a firewall application. L7-Filter version 2.23 (released September 2013) was used to be combined with Nefilter/iptables. The Source code nDPI used was compatible with the version nDPI ndpi-netfilter, revision 7340 (released in February 2014). ndpi-netfilter application (released October 2012) was a link between nDPI with netfilter/iptables, created by users github ewildgoose (Wildgoose, 2015).

C. Firewall configuration & Applications

The eleven Access services used to test the firewall nDPI and L7-Filter firewall were: BitTorrent, Encrypted BitTorrent, eDonkey, eDonkey Obfuscated Mode, Skype, Yahoo Messenger, Facebook, Twitter, Google, YouTube and YouTube with https. In the following section results are discussed. The firewalls were tested as follows:

- Denying access by the firewall was tested by adding rules to accept access to all data traffic except for data traffic from the eleven applications being tested.
- Allowing access by the firewall was tested by adding rules to deny access to all data traffic except for data traffic from the eleven applications.

Each application was tested on each of the three client computers that were connected to the firewall. Tests were performed five times to ensure accuracy. In each testing scenario, as well as the application being tested another application would be running to simulate non-target data traffic which had not been incorporated into the rule. In addition, applications were accessed within ten minutes of each other.

IV. Firewall Security Assessment

A. Assessment Methodology

Parameters used to measure accuracy in testing nDPI and L7-Filter as deep packet inspection firewalls were sensitivity and specificity. Sensitivity and specificity are parameters that were originally used in the medical world for measuring how good and reliable a classification process is, such as the classification of people who have certain illnesses (Zhu, Zeng, Wang, et al., 2010). Sensitivity measures the proportion of true positives, that is cases where the firewall correctly blocked data traffic, while specificity measures the proportion of true negatives, that is data traffic that was correctly allowed to pass through the firewall. Accuracy is a term that encompasses both sensitivity and specificity (Zhu et al., 2010) (Alcock & Nelson, 2013). Calculation of sensitivity and specificity value are defined as follows:

$$Sensitivity = \frac{\sum(true\_positive)}{\sum(true\_positive + false\_negative)} \times 100\% \quad (1)$$

$$Specificity = \frac{\sum(true\_negative)}{\sum(true\_negative + false\_positive)} \times 100\% \quad (2)$$

In order to get the sensitivity and specificity values in this research, the data packets were classified in the following categories:

(i) True Positive: Data packet services that should be included in the firewall rule and were targeted by the firewall rule.
(ii) True Negative: Data packet services that should not be included in the firewall rule and were not targeted by the firewall rule.
(iii) False Positive: Data packet services that should not be included in the firewall rule but were targeted by the firewall rule.
(iv) False Negative: Data packet services that should be included in the firewall rule that were not targeted by the firewall rule.

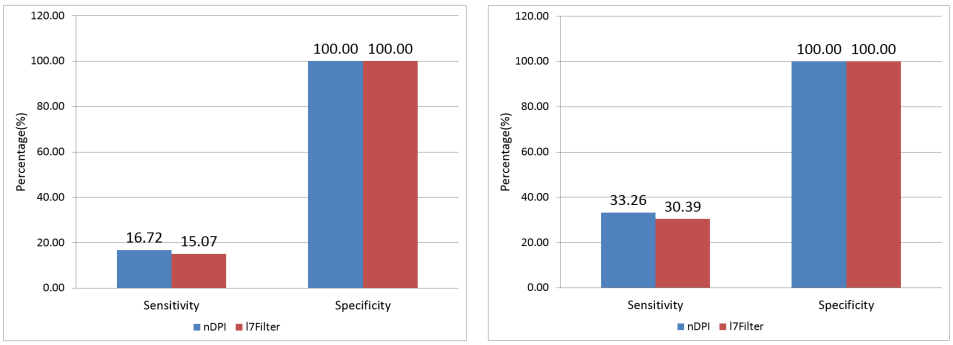
B. Testing System Performance

1) BitTorrent Scenario: At the time of deny of access to BitTorrent, using nDPI firewall or L7-filter firewall, new torrent downloads through BitTorrent client stopped. However, for torrents that had been in the download or seeding process when the deny rule was applied, the torrent data traffic was still running. In addition, low data sensitivity in both firewalls also indicated that there was more BitTorrent data traffic not being detected and stopped.

Upon denying access to BitTorrent, using either nDPI firewall or L7-Filter firewall, the new torrent downloads process through BitTorrent client would stop. However, torrents that were already being downloaded or were in the seeding process when the deny rule was applied continued running. Low sensitivity data in both firewalls indicated that more data traffic of BitTorrent not being detected and stopped (see Figure 5a).

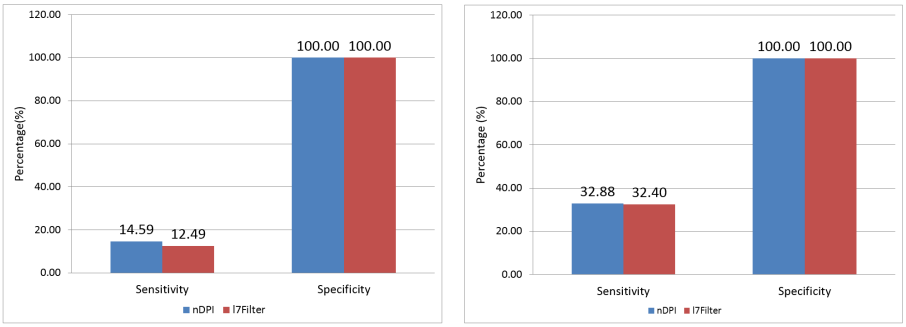
At the time of deny of access to all data traffic except BitTorrent, using nDPI or L7-filter, the running torrent download process on BitTorrent client stopped. However, for new torrents executed after the access allow rule of BitTorrent was activated, the download process through BitTorrent client still ran. The value of data sensitivity on both firewalls was higher than the value of sensitivity of BitTorrent access deny, which indicated there were more BitTorrent packets detected and forwarded by the firewall (see Figure 5b).

2) Encrypted BitTorrent Scenario: At the denying of access to BitTorrent using encrypted mode, torrent download processes that had previously been running did not stop, either upon denying of access performed using nDPI firewall or L7-filter firewall. However, for new torrent downloads that were executed



(a) Access denial of data traffic Bittorrent (b) Access approval of data traffic Bittorrent

Fig. 5: Data traffic BitTorrent



(a) Access denial of data traffic Encrypted Bittorrent (b) Access approval of data traffic Encrypted Bittorrent

Fig. 6: Data traffic Encrypted BitTorrent

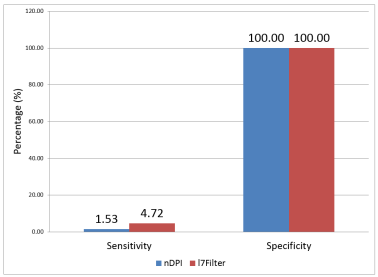
after denying of access applied, the process could not run. The sensitivity data value associated with BitTorrent in this scenario also indicated that there were more BitTorrent data packets undetected and stopped by the firewall, and showed that the sensitivity values in both firewalls was lower than in the unencrypted BitTorrent access deny.

On denying access to BitTorrent in encrypted mode, torrent download processes that were already running did not stop with either firewall. Yet, for torrent downloads that were carried out after the deny rule was applied, the processes stopped. The sensitivity value associated with BitTorrent in this scenario indicated that there were BitTorrent data packets not detected and stopped by the firewall. It also showed that the sensitivity values of both firewalls were lower than in the access deny scenario for unencrypted BitTorrent (see Figure 6a).

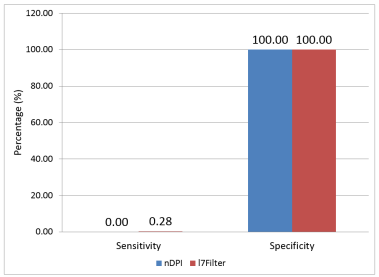
At the denying of access to all data traffic except BitTorrent, the torrent download process that was running on the BitTorrent client stopped, for both nDPI firewall and L7-Filter. But, the new torrent download process, still ran even through both firewalls. The sensitivity value in both firewalls was higher than the value of sensitivity in the encrypted BitTorrent access deny, which indicated there were more BitTorrent packets detected and forwarded by the firewall (see Figure 6b).

3) eDonkey Scenario: Access deny to eDonkey using DPI firewall and L7-filter firewall caused some of the downloading and searching files processes via eDonkey client to stop. Nevertheless, most of the download processes through eDonkey client could still run. This was showed by the low data sensitivity, in other words, there were more eDonkey data packets that were not stopped by both firewalls, although in this case the sensitivity value of L7-filter was higher than nDPI.

Denying access to eDonkey by using nDPI and L7-Filter firewall caused some downloads and file search processes to stop. Nonetheless, most download processes through eDonkey client continued running. This was represented by the low sensitivity value. In other words, there were eDonkey data packets not terminated by both firewalls, though in this case the sensitivity value of L7-Filter was higher than nDPI (see Figure 7a).

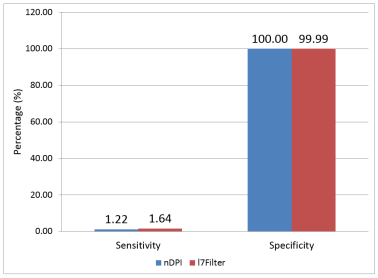


(a) Access denial of data traffic eDonkey

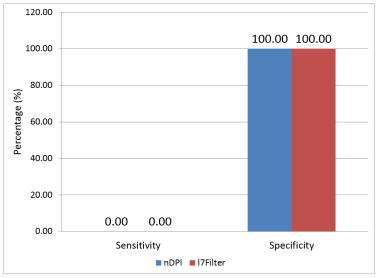


(b) Access approval of data traffic eDonkey

Fig. 7: Data traffic eDonkey



(a) Access denial of data traffic Obfuscated Mode eDonkey



(b) Access approval of data traffic Obfuscated Mode eDonkey

Fig. 8: Data traffic Obfuscated Mode eDonkey

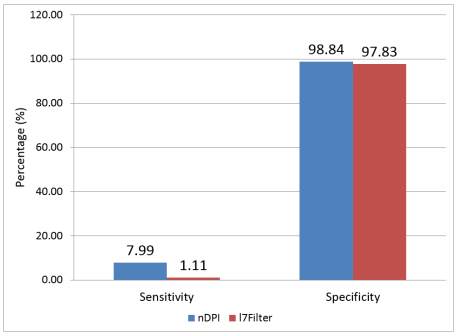
Denying access to data traffic other than eDonkey, using nDPI firewall and L7-filter firewall, caused eDonkey client to be unable to access the eDonkey network. However, based on sensitivity data, for L7-filter (but not for nDPI) there were eDonkey data packets forwarded by L7-filter firewall even though overall the sensitivity value in both firewalls was lower than when access to eDonkey was denied (see Figure 7b).

4) Obfuscated Mode eDonkey Scenario: At the denying of access to eDonkey, either using nDPI firewall or L7-filter firewall, a small portion of download processes on the eDonkey client stopped. In this scenario, the sensitivity data indicated that data packets stopped by the firewall were fewer than the ones that passed. There were more eDonkey data packets stopped by L7-filter firewall than by nDPI. In addition, for the sensitivity values of eDonkey access deny in this scenario, on both nDPI firewall and L7-filter firewall, values were lower than when the eDonkey client was not using obfuscated mode.

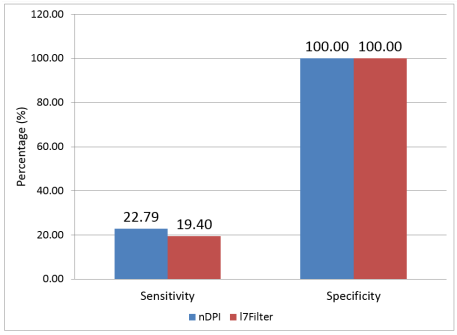
When denying access to eDonkey, using either nDPI or L7-Filter firewall, a small portion of download processes on eDonkey client stopped. In this scenario, the sensitivity data indicated that the data packets terminated by the firewall were fewer than those that passed, although more eDonkey data packets were stopped by the L7-Filter firewall. In addition, the sensitivity value of was lower than when eDonkey client was not using obfuscated mode (see Figure 8a).

On access deny besides eDonkey data traffic, eDonkey client could not access the eDonkey network. This happened on denying of eDonkey access using nDPI firewall and L7-filter firewall. Sensitivity data of this access reception also indicated that there were no eDonkey data packets forwarded by both firewalls.The sensitivity data of the L7-filter firewall in this scenario was lower than the sensitivity data of access reception on eDonkey without obfuscated mode (see Figure 8b).

5) Skype Scenarios: On denying of access to Skype, either using nDPI firewall or L7-filter firewall, the Skype client still connected to Skype and used it as instant messaging application. However, when exiting and re-entering the Skype application, Skype was stopped by the nDPI firewall while L7-filter firewall allowed the Skype application to be connected. Moreover, based on sensitivity data in this scenario, there were Skype data packets terminated by nDPI firewall and L7-filter firewall. The nDPI sensitivity values were higher than L7-filter firewall.

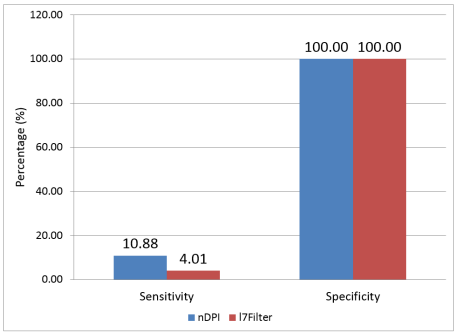


(a) Access denial of data traffic Skype

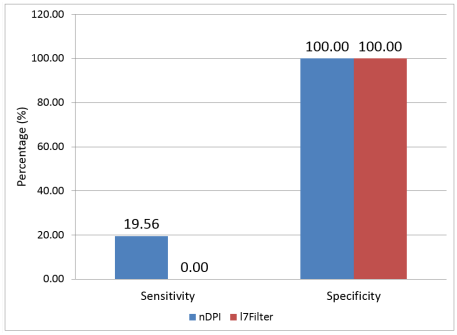


(b) Access approval of data traffic Skype

Fig. 9: Data traffic Skype



(a) Access denial of data traffic Yahoo



(b) Access approval of data traffic Yahoo

Fig. 10: Data traffic Yahoo

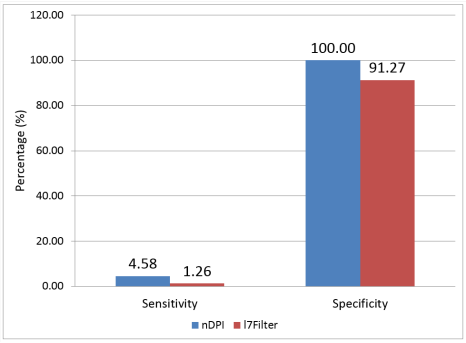
Upon denying access to Skype, using either nDPI firewall or L7-Filter firewall, Skype client application could still be connected and used as an instant messaging application. However, when the user logged out and then signed back in to Skype, Skype was denied by nDPI firewall, however on L7-filter firewall, Skype could still be connected. Based on sensitivity data in this scenario, there were Skype data packets terminated by nDPI firewall and L7-Filter firewall. The sensitivity value of nDPI firewall was higher than the L7-Filter firewall (see Figure 9a).

On access deny of all data traffic except Skype using nDPI firewall or L7-filter firewall, the Skype application could not be connected and used as an instant messaging application. But, based on data sensitivity in this scenario, there were Skype data packets transmitted by both firewalls. The sensitivity values of the nDPI firewall were higher than the L7-filter firewall. Furthermore, based on data sensitivity, sensitivity values of both firewalls was higher than when they were used to terminate access for Skype data packets (see Figure 9b).

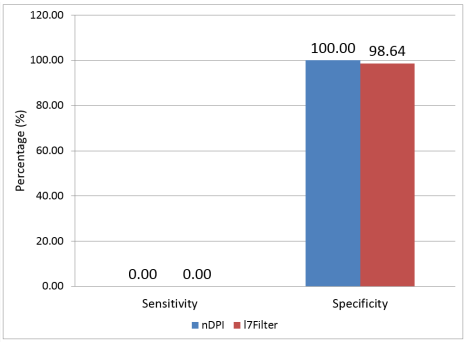
6) Yahoo Messenger Scenario: Deny of access to Yahoo Messenger using nDPI firewall and L7-filter firewall resulted in it not being possible to connect to Yahoo Messenger application within one to two minutes after the deny rule was enforced. Nevertheless, low values of sensitivity data on both firewalls indicated that there were more packets not detected by the firewall. Additionally, in this scenario the sensitivity values of nDPI firewall were higher than L7-filter firewall.

Denying access to Yahoo Messenger using nDPI firewall or L7-Filter firewall, caused the Yahoo Messenger application to disconnect within 1-2 minutes of the deny rule being applied. Nevertheless, the low sensitivity values for both firewalls indicated there were some undetected packets by the firewalls. Additionally, in this scenario the sensitivity value of nDPI firewall was higher than L7-Filter firewall (see Figure 10a).

For access deny for all data traffic except Yahoo Messenger, Yahoo Messenger application was still able to connect to the network, both using nDPI firewall and L7-filter firewall. Yet, based on sensitivity data, there were data packets not forwarded by nDPI firewall, with higher sensitivity values for L7-filter firewall than nDPI firewall on the Yahoo Messenger access deny scenario (see Figure 10b).

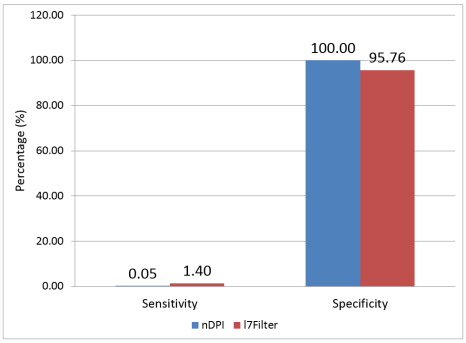


(a) Access denial of data traffic Facebook

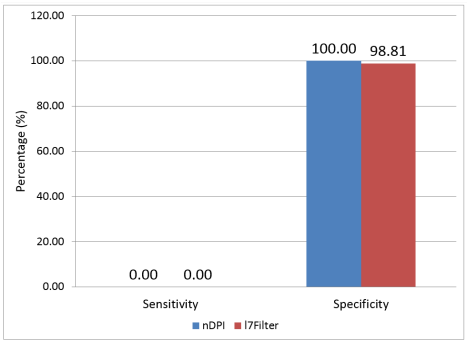


(b) Access approval of data traffic Facebook

Fig. 11: Data traffic Facebook



(a) Access denial of data traffic Twitter



(b) Access approval of data traffic Twitter

Fig. 12: Data traffic Twitter

7) Facebook Scenario: When access deny to Facebook was imposed by nDPI firewall and L7-Filter, if access to Facebook had been made before the rule was applied it was still able to run. However, when accessing Facebook after deny rule was applied, access to Facebook website could not be achieved. In addition, the sensitivity data for nDPI firewall was higher than the sensitivity data for L7-filter firewall, although both had low sensitivity values.

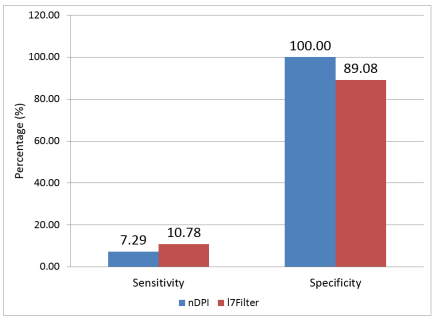
When access to Facebook was denied using nDPI firewall and L7-Filter firewall, access to Facebook conducted before the rule was applied continued. However, when accessing Facebook after the deny rule was applied, access to the Facebook website could not be achieved. Besides that, sensitivity data of the nDPI firewall was higher than for the L7-filter firewall, although both still had low sensitivity values (see Figure 11a).

Upon denying of all data traffic except Facebook, access to Facebook could not be achieved. This applied when both nDPI firewall and L7-filter firewall were used. Moreover, the sensitivity data also showed that there were no Facebook data packets forwarded through either firewalls (see Figure 11b).

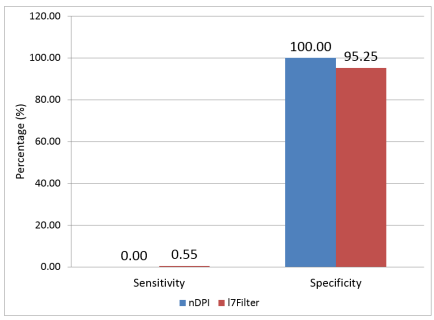
8) Twitter Scenario: When denying of access to Twitter was made, access to Twitter was still able to be achieved, both before and after the access deny was enacted. This applied to when both nDPI firewall and L7-filter firewall were used to stop access to Twitter. However, based on sensitivity data, there were Twitter data packets stopped by both firewalls, with sensitivity values of L7-filter firewall higher than nDPI firewall.

Upon denying access to Twitter, access to Twitter could still be achieved, whether the access was made before or after the deny rule was applied. This applied to both the nDPI firewall and the L7-Filter firewall. Based on sensitivity data, there were Twitter data traffic packets stopped by both firewalls, and the sensitivity value of L7-filter was higher than nDPI firewall (see Figure 12a).

At the denying of all data traffic except the Twitter website, access to Twitter site could not be gained. This happened when both nDPI firewall and L7-filter firewall were used. Besides this, the sensitivity data also showed that there were no Twitter data packets forwarded through either firewall (see Figure 12b).

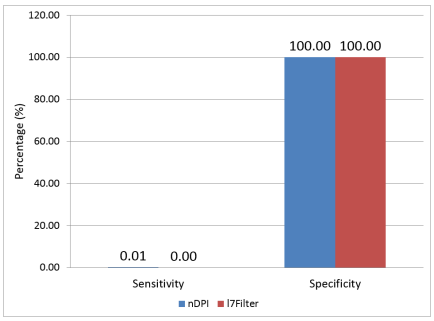


(a) Access denial of data traffic Google

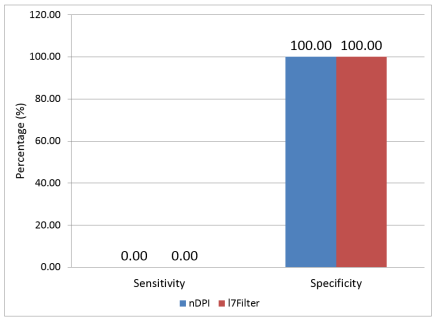


(b) Access approval of data traffic Google

Fig. 13: Data traffic Google



(a) Access denial of data traffic YouTube



(b) Access approval of data traffic YouTube

Fig. 14: Data traffic YouTube

9) Google Scenario: At the point of deny of access to the Google website, access to Google was still able to be gained, using nDPI firewall and L7-filter firewall, if access was made before the access deny was imposed. But, when access was gained after access deny was implemented on both firewalls access could not be gained. However, based on data sensitivity, there were Google data packets stopped by both firewalls, with the sensitivity values of L7-filter firewall higher than nDPI firewall.

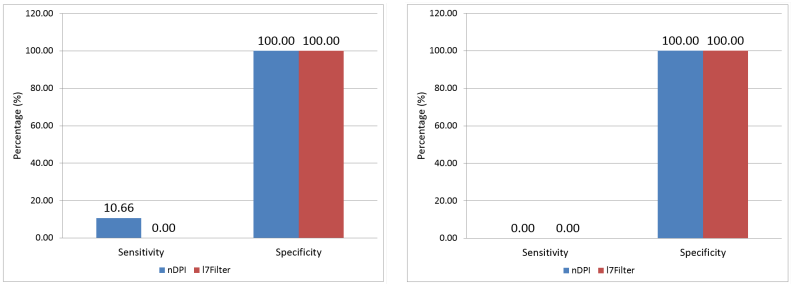
Upon denying access to the Google site on either nDPI firewall or L7-Filter firewall, access to Google still could be gained if it had been accessed before the deny rule was applied. But, it could not be achieved if the access took place after the deny rule was applied.Based on sensitivity data, there were data traffic packets stopped by both firewalls, with the sensitivity value of L7-filter firewall higher than nDPI firewall (see Figure 13a).

At the denying of all data traffic except to the Google web site, access to Google sites could not be achieved. This happened when both the nDPI firewall and L7-filter firewall were used. Furthermore, the sensitivity data indicated there were Google data packets forwarded by L7-filter firewall, but no Google data packets terminated by nDPI firewall (see Figure 13b).

10) YouTube Scenario: At the time of access deny to the YouTube website, access to YouTube was still able to be achieved, using nDPI firewall and L7-filter firewall. This applied to both access to YouTube performed before and after deny rule was enforced. There were YouTube data packets forwarded by nDPI firewall based on sensitivity data in this scenario but not by L7-filter.

Denying access to the YouTube site did not cause access to YouTube to be blocked for either the nDPI firewall or the L7-Filter firewall. This applied to both access that took place before and after the deny rule was applied. Nevertheless, there were YouTube data packets which were forwarded by nDPI firewall based on the sensitivity data in this scenario (see Figure 14a).

At the denying of all data traffic except YouTube website, access could not be gained. This happened when both nDPI firewall and L7-filter firewall were used. Sensitivity data of nDPI firewall and L7-filter firewall in this scenario indicated that there was no data detected and passed by either firewalls (see Figure 14b).



(a) Access denial of data traffic YouTube (https)

(b) Access approval of data traffic YouTube (https)

Fig. 15: Data traffic YouTube (https)

11) YouTube (https) Scenario: At the denying of access to the YouTube website, if YouTube had been accessed with https before deny rule was applied, then access was still able to be gained, when either nDPI firewall or L7-filter was used. However, for accessing YouTube with https after deny rule was applied, access could not be gained using nDPI firewall while it was still able to be done using L7-filter firewall. In addition, the sensitivity data showed that there were no YouTube data packets stopped by L7-filter firewall.

When denying access to the YouTube site with https, access continued when it had been gained before the rule was applied, for both firewalls. However, accessing YouTube with https after the access deny rule was not possible for the nDPI firewall, but was possible for the L7-filter firewall. Sensitivity data showed that there were no packets of YouTube data traffic terminated by the L7-Filter firewall (see Figure 15a).

At the denying of all data traffic except YouTube, accessing the site using https could not be done. This happened both when nDPI firewall and L7-filter firewall were used. Sensitivity data indicated that there were no YouTube data packets forwarded through nDPI firewall or L7-filter firewall (see Figure 15b).

V. Result and Analysis

A. Data Retrieval Result

Results were taken based on the data logs from the firewalls for the eleven scenarios, as well as a log of the netstat program and PCAP file on the client. Firewall and netstat log data was matched to obtain information about the program on the client of origin and destination of the data packet. In addition the IP address data and port number that was accessed and used were recorded, as well as data packet information regarding the firewall rule and which used or did not use deep packet inspection techniques.

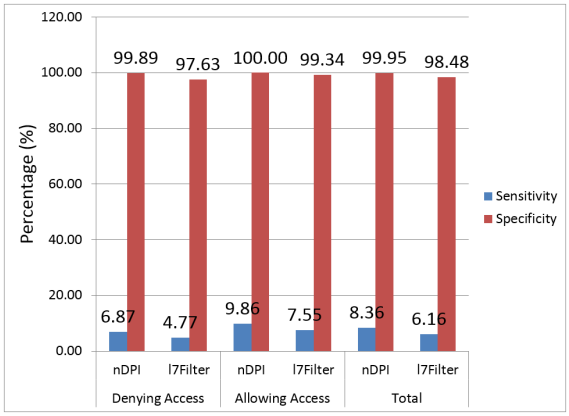


Fig. 16: Average value of sensitivity and specificity

Based on the data about the packet data traffic, the sensitivity and specificity of the firewall could be calculated for each scenario, as previously discussed. Execution time of the firewall rule was also calculated for nDPI and L7-filter. This is the checking time of a rule, or in other words how long a rule took to be implemented. It is calculated by taking the time the rule was implemented and subtracting the time the rule was actioned.

B. Access Scenario Result

$$T_X = T_{fin} - T_{act} \tag{3}$$

Description:

- T_X = Total time for Execution
- T_{fin} = The time the rule was actually implemented
- T_{act} = The time the rule was actioned

This section displays the overall data of access scenarios. Based on average data of sensitivity and specificity for all scenarios, it can be seen that the sensitivity and specificity values of nDPI firewall were higher than those for L7-filter firewall. In addition, by using equation (3) the average value of sensitivity and specificity for the access reception scenarios both in nDPI firewall and L7-filter firewall were higher than the average value of sensitivity and specificity in access deny scenarios.

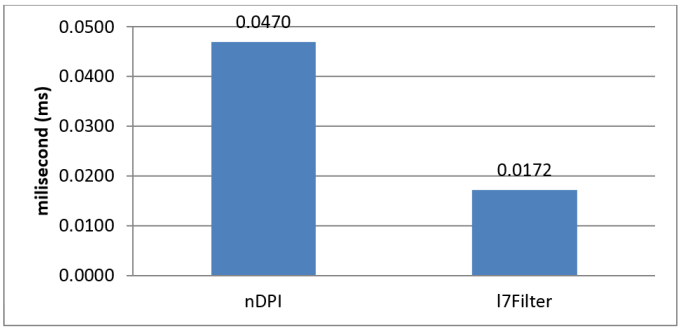


Fig. 17: Average time for execute the rules

C. Analysis and Validation

In Figure 18 and 19, it can be noted that nDPI firewalls had higher sensitivity and specificity values than the L7-filter firewall in live performance testing. This is because, although both firewalls had the same basic way of classifying, matching data in the packet with the packet's signature data, signature data that exists on nDPI had more data in accordance with data packets on each access scenario. Besides that, the thing that made nDPI superior in specificity value was a data packet classification error in L7-Filter. As evidence is the denying of access to the website `http://www.reddit.com/search?q=facebook.com` when testing access deny to Facebook was being run. However, the packet data signature of L7-filter for eDonkey data traffic was more appropriate than nDPI, based on the test results of eDonkey access scenarios and eDonkey obfuscated mode access.

Sensitivity data in the non-encrypted BitTorrent access scenario was higher than in the encrypted BitTorrent access scenario, as well as in non-obfuscated mode eDonkey access than in the obfuscated mode eDonkey access scenario. This was due to the packets data encryption process, in particular the protocol and concealment processes on obfuscated mode. Data packets using the protocol did not match the signature data which was available on nDPI and L7-Filter. However, the sensitivity of YouTube access scenarios was lower than YouTube (https) on the nDPI firewall, because, the signature on nDPI for YouTube, is designed to better recognize accessing YouTube site with https.

For sensitivity data of all scenario test results, the values obtained were less than 50%. In other words, there were more data packets that should have been detected and dealt with, but were not detected by the nDPI firewall or the L7-filter firewall. This is because in the nDPI and L7-filter systems, if there are a number of data packets in the flow of data packets, with different limits on the number of data

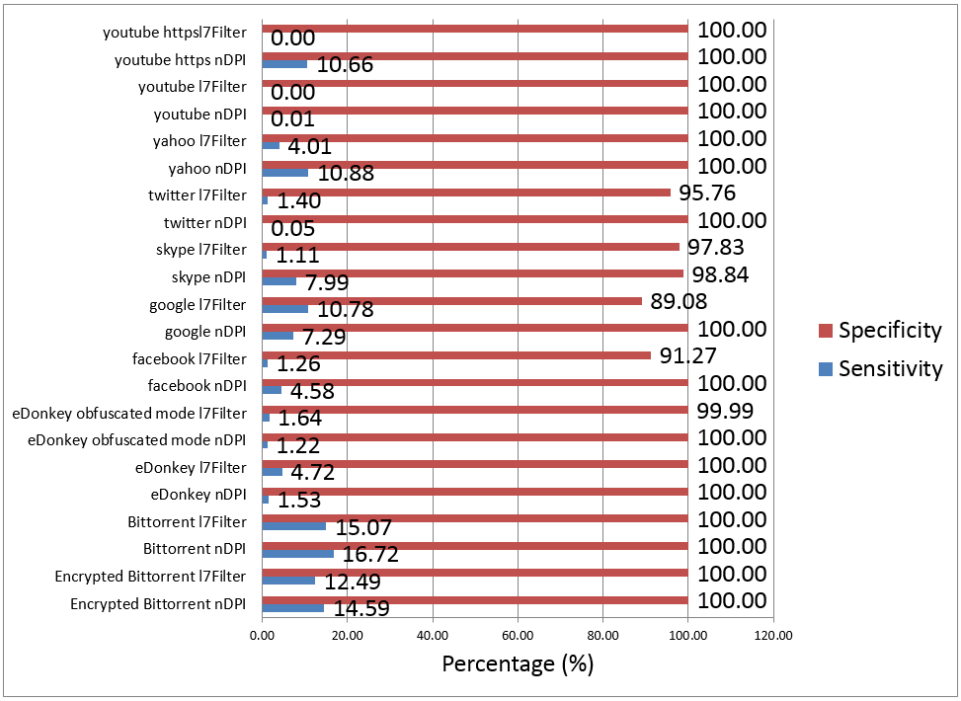


Fig. 18: Overall result of denying access

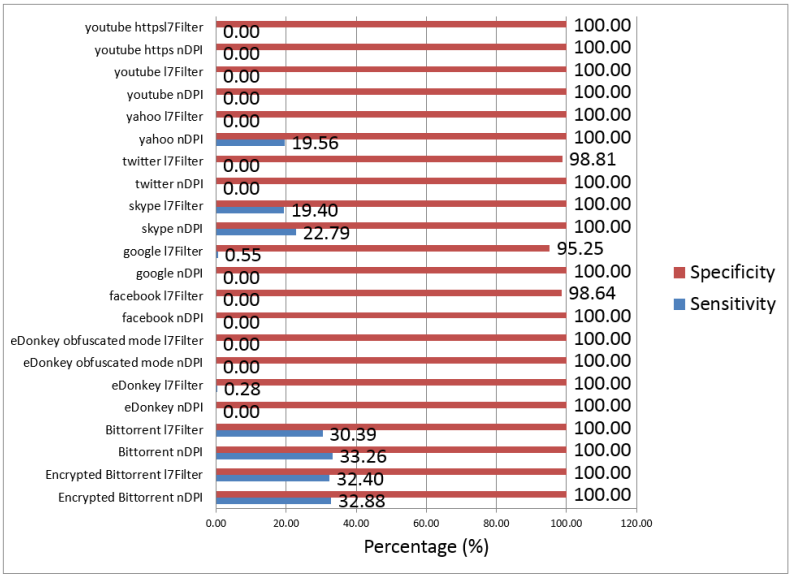


Fig. 19: Overall result of denying access

packets, then the entire flow of data packets will be classified in the same category. Thus, if there is a data packet that should be forwarded by firewall, but that packet is not forwarded by nDPI firewall or L7-filter firewall, then the other data packets in the same flow will not be forwarded. Similarly, for the access deny scenarios, if there was a data packet that should have been stopped, but was not stopped, then the other data packets in the same flow would not be stopped. This led to more data packets being stopped than should have been stopped, so sensitivity was less than in the receiving access scenario.

When the data of the firewall rule execution is observed based on live performance testing, nDPI firewall had an execution average value longer than the rule execution in L7-filter firewall. This is because the classification process on nDPI not only uses signature comparison of data packets, but also a long examination of the payload in a packet, or the size of a field, depending on the type of package on

the search function that is being executed. In addition, it is also a factor causing the overall sensitivity and specificity values for nDPI firewall to be higher than the L7-filter firewall.

VI. Conclusion

Based on the results of the testing and analysis these are the conclusions. This research managed to successfully incorporate deep packet inspection modules nDPI and L7-filter into the XTables module of netfilter in live performance testing. This increases netfilter's performance by enabling it to inspect data packets up to application layer. The research also shows that nDPI is more accurate than L7-Filter, although the average execution speed of L7-Filter was faster by 0.0298ms. For value sensitivity compared with L7-Filter, nDPI was 2.1% more sensitive when used to deny access and 2.31% more sensitive when used to allow access to services. As for specificity nDPI was 2.26% more specific when used to deny access, and 6.66% more specific when used for allowing access. Different networks and system administrators using open source firewalls have their own methods and rules for securing internal networks. However this research can help all in creating tough and secure systems. Further research should be conducted on different Internet applications.

Acknowledgment

This work was supported by Forensics and Security Laboratory, Informatics Engineering Telkom School of Computing, Telkom University, Bandung, Indonesia. Thank you for anonymous person who did not want to be mentioned in this paper, thank you for your cooperation and suggestion.

References

Alcock, S., & Nelson, R. (2013). Measuring the accuracy of open-source payload-based traffic classifiers using popular internet applications. In *Local computer networks workshops (lcn workshops), 2013 ieee 38th conference on* (pp. 956–963).

Ayoub, D. (2012). *Why protection and performance matter : The benefits of multi-core reassembly-free deep packet inspection* (Tech. Rep.).

Bober, A. (2009). Introduction to layer 7-filter [Computer software manual].

Bujlow, T., Carela-Español, V., & Barlet-Ros, P. (2013). *Comparison of deep packet inspection (dpi) tools for traffic classification* (Tech. Rep.). Universitat Politècnica de Catalunya.

Bujlow, T., Carela-Español, V., & Barlet-Ros, P. (2014). *Extended independent comparison of popular deep packet inspection (dpi) tools for traffic classification* (Tech. Rep.). Universitat Politècnica de Catalunya.

Bujlow, T., Carela-Español, V., & Barlet-Ros, P. (2015). Independent comparison of popular dpi tools for traffic classification. *Computer Networks*, *76*, 75–89.

Casado, M., Amidon, K. E., Balland III, P. J., Gude, N., Pettit, J., Pfaff, B. L., ... Wendlandt, D. J. (2016, January 14). *Network operating system for managing and securing networks.* (US Patent 20,160,013,969)

Ciampa, M. (2011). *Security+ guide to network security fundamentals*. Cengage Learning.

Cisco, I. (2003). Security configuration guide, release 12.2. *CISCO, San Jose, CA*.

Deri, L., Martinelli, M., Bujlow, T., & Cardigliano, A. (2014). ndpi: Open-source high-speed deep packet inspection. In *Wireless communications and mobile computing conference (iwcmc), 2014 international* (pp. 617–622).

Fabbri, R., & Volpe, F. (2013). *Getting started with fortigate*. Packt Publishing Ltd.

Garrett, A., Drenan, G., & Morris, C. (2002). *Juniper networks field guide and reference*. Addison-Wesley Professional.

Gheorghe, L. (2006). *Designing and implementing linux firewalls with qos using netfilter, iproute2, nat and l7-filter*. Packt Publishing Ltd.

L7-Filter. (2008). *L7-filter kernel version howto.* http://l7-filter.sourceforge.net/... (Accessed: 2015-12-12)

Liu, D., Miller, S., Lucas, M., Singh, A., Davis, J., et al. (2006). *Firewall policies and vpn configurations*. Syngress.

ntop. (2013, 12). ndpi - quick start guide (1st ed.) [Computer software manual]. Via Ponte a Piglieri 8, 56122 Pisa, Italy.

Oriyano, S.-P. (2014). *Ceh: Certified ethical hacker version 8 study guide*. John Wiley & Sons.

Ou, G. (2013). Understanding deep packet inspection (dpi) technology. *Internet: http://www.digitalsociety.org/files/gou/DPI-Final-10-23-09.pdf*.

Radisys. (2010). Dpi: Deep packet inspection motivations, technology, and approaches for improving broadband service provider roi [Computer software manual].

Rash, M. (2007). *Linux firewalls: Attack detection and response with iptables, psad, and fwsnort*. No Starch Press.

Satrya, G., & Shin, S. (2015). Optimizing rule on open source firewall using content and pcre combination. *Journal of Advances in Computer Networks*, *3*(3), 308-314. doi: http://dx.doi.org/10.18178/JACN.2015.3.4.188

Satrya, G. B., Cahyani, N. D., & Andreta, R. F. (2015). The detection of 8 type malware botnet using hybrid malware analysis in executable file windows operating systems. In *Proceedings of the 17th international conference on electronic commerce 2015* (pp. 5:1–5:4). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2781562.2781567 doi: 10.1145/2781562.2781567

Scarfone, K., & Hoffman, P. (2009). Guidelines on firewalls and firewall policy. *NIST Special Publication*, *800*, 41.

SonicWALL. (2015). *Configuring firewall settings.* http://help.mysonicwall.com. (Accessed: 2015-12-12)

Thomason, S. (2012). Improving network security: Next generation firewalls and advanced packet inspection devices. *Global Journal of Computer Science and Technology*, *12*(13-E).

Wildgoose, E. (2015). *Netfilter kernel module for deep packet inspection filtering.* https://github.com/ewildgoose/ndpi-netfilter. (Accessed: 2015-12-12)

Zhu, W., Zeng, N., Wang, N., et al. (2010). Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas® implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, 1–9.