International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:6, No:7, 2012

Improvising Intrusion Detection for Malware Activities on Dual-Stack Network Environment

Zulkiflee M., Robiah Y., Nur Azman Abu, Shahrin S.

***Abstract*—**Malware is software which was invented and meant for doing harms on computers. Malware is becoming a significant threat in computer network nowadays. Malware attack is not just only involving financial lost but it can also cause fatal errors which may cost lives in some cases. As new Internet Protocol version 6 (IPv6) emerged, many people believe this protocol could solve most malware propagation issues due to its broader addressing scheme. As IPv6 is still new compares to native IPv4, some transition mechanisms have been introduced to promote smoother migration. Unfortunately, these transition mechanisms allow some malwares to propagate its attack from IPv4 to IPv6 network environment. In this paper, a proof of concept shall be presented in order to show that some existing IPv4 malware detection technique need to be improvised in order to detect malware attack in dual-stack network more efficiently. A testbed of dual-stack network environment has been deployed and some genuine malware have been released to observe their behaviors. The results between these different scenarios will be analyzed and discussed further in term of their behaviors and propagation methods. The results show that malware behave differently on IPv6 from the IPv4 network protocol on the dual-stack network environment. A new detection technique is called for in order to cater this problem in the near future.

***Keywords*—**Dual-Stack; Malware; Worm; IPv6;IDS

I. Introduction

IPv6 is a new protocol which was invented initially to overcome the issue of IP address depletion in IPv4 network environment. Moreover, this new protocol also offer many other advantages compares to the existing IPv4 protocol [1-3]. Although IPv6 offers a lot of benefits, still many users are reluctant to migrate from IPv4 to IPv6 fully due the level of service offers in IPv6 is still not as good as in IPv4. Since IPv4 addresses are facing depletion, migrating to IPv6 is eventually inevitable [4, 5]. Hence, transition mechanisms have been introduced to promote the migration from IPv4 to IPv6. These IPv4 to IPv6 transition mechanisms are techniques which can be used to communicate nodes from different IP protocols. Unfortunately, these mechanisms allow malware to propagate its attack from IPv4 to IPv6 network environment.

Zulkiflee M. is with the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 75450 Melaka, Malaysia (phone: 606-331-6671; fax: 606-331-6500; e-mail: zulkiflee@utem.edu.my).

Robiah Y. is with the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 75450 Melaka, Malaysia (e-mail: robiah@utem.edu.my).

Nur Azman Abu is with the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 75450 Melaka, Malaysia (e-mail: nura@utem.edu.my).

Shahrin S. is with the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 75450 Melaka, Malaysia (e-mail: shahrinsahib@utem.edu.my).

The malware was originally intended to attack IPv4 network which it uses IPv6 protocol to propagate the malware attack. This issue is becoming worse since many network administrators do not aware of IPv6 existence in their network as the IPv6 has an auto configuration features [6].

Malware is software which is developed and designed to do harm on computers. Malware is becoming a significant threat on computer network nowadays. Surveys show that malware not only just involving financial lost, but it also can cause fatal errors which may cost lives in some cases [7, 8]. Currently, there are 250 malware variants coming into computer network environment every day[9]. These so called new age malwares are innovated from the existing malware. These malwares are modified and some modules are added in order to avoid being detected by existing malware detection techniques.

The objective of this paper is to show the concept that existing IPv4 intrusion detection techniques need to be improvised in order to detect malware activity in dual-stack network environment more efficiently.

In the following sections, some related works will be discussed and followed by the methodology used in this experimental research. The experimental design will be clarified and some result and analysis will be elaborated. Finally, the conclusion and future work will be stated towards the end of this paper.

II. State of Art

A. IPv4 to IPv6 Transition Mechanisms

There are two transition mechanisms as stated in RFC 4213 which was revised from RFC 2893 namely Dual-ip layer and tunneling IPv4 over IPv6. In this study, the dual ip layer also known as dual-stack technique will be used. This technique is the most straight forward technique to be implemented. Each host and router is using both IPv4 and IPv6 protocols. In this technique, the transition process is transparent to the end users. The users will not realize which protocol they are using as both these protocols are enabled all the time. If they are using IPv4 application then the node will be using IPv4 address and protocol and vice versa. The advantage of this technique are stable and easy to be implemented compares to other techniques [10]. Fig. 1 shows how dual-stack implemented on each node.

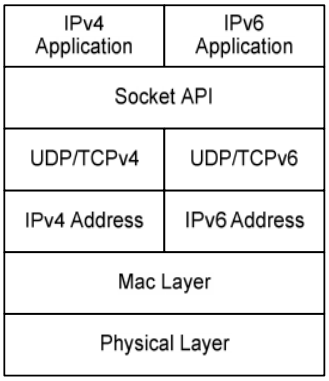

Fig. 1 Dual-Stack transition mechanism architecture

Fig. 1 shows the architecture of dual-stack transition mechanism. If a node communicates in IPv4 then it will use same physical and mac layer addresses. But then it will start use IPv4 address and protocols towards to IPv4 application layer and vice versa with IPv6 protocol.

B. Malware Propagation

Malware are represented by several forms namely virus, Trojan, spyware, adware and worms [11, 12]. Each of these forms has different characteristics once it is infected a network. Their method of propagation also varied including transfer memory sticks, peer-to-peer files download, sharing file and so forth. Based on malware characteristics, worm is capable to be modeled as its propagation did not need any human intervention [13].

The worm scanning methods can be divided into three categories as defined by [14] 1) naïve random scanning, 2) sequential scanning and 3) localized scanning. The naïve random scanning method has pre-defined target IP addresses without concerning about the victim's network configuration. Meanwhile, Sequential scanning method is defined to search for vulnerable hosts through the nearest IP address space based on the infected host configuration. Whereas, localized scanning method will search for vulnerable hosts in the local network by using the network information gained from the infected before the worm initiates its attack.

In this study, two real worms were released on the testbed namely Nimda and Sasser. Although these worms were considered obsolete, these worms are still sufficient enough to be used to prove our concept that native IPv4 malwares still hunt for IPv6 addresses. The main reason these worms were chosen is because these worms targeted victims' IP addresses were not pre-defined but rather randomly assigned based on MAC information from the infected node to launch their attack [15, 16]. Hence, if IPv6 protocol is available then these worms will craft IPv6 packets to launch its attack to IPv6 network. With some modification made on the existing worms, probably IPv6 network can be easily penetrated by this type of worms.

C. Existing Intrusion Detection Technique

Intrusion detection plays important role to accommodate defense line for a network from being exposed by malicious users and incapacity of operating system to provide minimal protection from variety of attack attempts [17]. In this case, intrusion detection can be used as a form of defense line from worm infection. Many studies have been conducted to propose solutions to detect intrusion activities occurred on a network. However, many of these studies were conducted by using IPv4 network traffic dataset. Some features used to detect intrusion in IPv4 perhaps need to be improvised as a same attack used different methods in IPv6 network environment.

In this study, the features which will be emphasized are only 3 namely IP address, Protocol and Number of packets in a time. In Table I, some conducted studies were using these 3 features were used in developed their detection model.

TABLE I
IPV4 DETECTION TECHNIQUES USING THREE FEATURES

Authors	IP address	Protocol Type	Number of Packets
Sangkatsanee, 2011 [18]		✓	✓
Antonis, 2010 [19]			✓
Li, 2009 [20]	✓	✓	
Faizal, 2009 [21]	✓	✓	
Labib, 2002 [22]	✓	✓	

Table I shows some studies conducted in intrusion detection were using IP address, Protocol Type and Number of packets in a time as a part of features used in constructing their detection model based on IPv4 dataset. In this study, these features will be used as comparison of the infection method between IPv4 and IPv6 protocols after a network being infected by a worm.

D. IPv6 Datasets

For the past few years, many studies were conducted on forecasting worm propagation on a network. Most of these studies proposed models based on mathematical theory. Su [23] in her papers has mentioned a few models of worm propagation namely Simple Epidemic Model (SEM), Kermack-Mckendrick Model (KM), Two Factor Model (TFM). These models were novel models produced in epidemiology research. In the same paper, Su also improvised TFM and proposed worm propagation model called Three Layer Worm Propagation Model (TLWM). Meanwhile, Okamura applied Morkovian model method to propose his kill-signal models [24]. Based on our observation, these models were more generic and assuming the worm propagation was not under the influence of neither IPv4 nor IPv6 network environment. Hence, these studies were neglecting IP protocol in constructing their models.

Some other models developed were focusing on IPv6 network environment. Kamra [5] is taken DNS service into its consideration on developing his model. The result was gained from mathematical simulation. In the meantime, Ting [25] was using a testbed in her study to model worm propagation on dual-stack network environment. However, the testbed was just used to identify seed value of worm propagation.

Later, the seed value was used in her simulation to get the propagation model on dual-stack network environment. As the result, the propagation model was not exclusively constructed based on the testbed environment.

Based on our literature findings, most of worm propagation models were constructed by using mathematical simulation. These studies help us on understanding the basic understanding about the overall issue. Not many worm propagation in IPv6 studies were conducted in real life environment due to lack of IPv6 datasets and lack of resources to do so. Hence, this study tries to reduce the gap by using a real hardware and genuine malware to be released to observe its behavior on real dual-stack network environment.

Many researchers agree that a testbed is needed for further IPv6 network investigation [26-28]. Not many IPv6 worm detection studies were conducted in IPv6 network environment. The main reason is because it is not easy to gather a data to be used in analyzing IPv6 traffic pattern. Based on our knowledge, the only anonymity IPv6 data available is at Caida [29]. This data is consisting of a real packet captured in IPv6 network environment. Nevertheless, before this data are used for further analysis, it is better to have a small dataset of IPv6 which can help us have solid fundamental knowledge about IPv6 traffic pattern about malware propagation on dual-stack network. Hence, a fully functional IPv4/IPv6 network testbed have to be designed and implemented to study about worm behavior in dual-stack network environment.

III. Methodology

In the direction of concluding the result, a sequence of work flow as a process of malware analysis has been designed as in Fig. 2.

Fig. 2 shows the process flow of analyzing malware behavior in dual stack network environment. This process flow was innovated from process flow for malware behavior analysis proposed by Zolkipli [30].

Since malware attack is extremely hazardous, a testbed was designed and implemented in order to test the malware behavior on dual-stack network environment. This testbed is used to ensure the malware just propagate in isolated network environment and to avoid malware migration to the real live network. The testbed design for this study can be found in Fig. 3.

Before the process begin, a clean testbed needs to be ready. This stage is the most crucial stage as this affects the final result for this test. Some malware are really hard to be cleaned because after it infected a computer, it may remain in the computer's memory even after the malware was cleaned by using the antivirus software. Hence, it is vital to cleaned thoroughly each node including formatting all computers involved. The switch and router used also need to be cleaned.

All configurations have to be deleted and restored as it just being used for the first time. Before the malware is being released, all nodes connectivity must be tested to ensure everything is working well as it is expected. This process must be executed every time before a new scenario is taken place.

Although process is very time consumption, but the process cannot be neglected as this will influence the final result of this study.

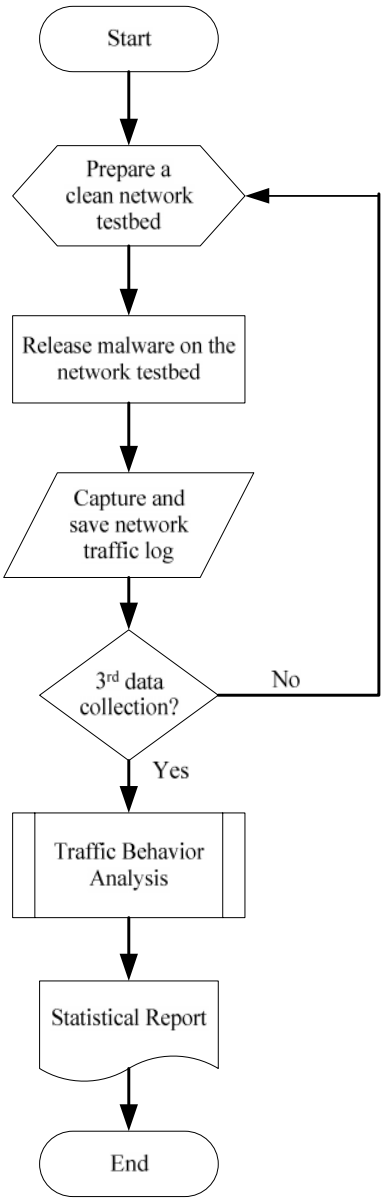

Fig. 2 Malware analysis process flow

After the testbed is ready, sniffer node is started to capture idle network traffic log. After 10 minutes, the sniffer tool is stopped and the log is saved for further analysis. Next step, the malware was released at once. Then, again the sniffer tool is re-activated to capture all packets through the gateway router.

The gateway router is important in this experiment is because it simulates as if this environment is accessible to the other networks. Therefore, the network design is able to deceive the malware to launch its attack to broader scale rather than local area network only. After 10 minutes, the sniffer tool is stopped and then the log is saved for further analysis.

The same scenario using the same malware is repeated three times. Each traffic log is saved and all logs are combined in the analysis stage.

On the traffic behavior analysis stage, all the data are being analyzed both individually and combine all together. The main elements used in this analysis stage are the frequency packet released and protocol used by both IP protocols. This information is extracted from the saved traffic logs.

Finally after all data being analyzed, the findings and outcomes are documented in a statistical report. Some figures have been presented to aid readers to appreciate about the whole idea of this study. A study shows that the IPv4 malware still can survive in IPv6 network environment [13]. However, this study will show whether IPv4 malware could migrate from IPv4 to IPv6 network environment via a transition mechanism. Towards the end of this paper, the result will prove whether this issue is valid or not.

IV. Experimental Design

The objective of this experiment is to analyze how IPv4 malware behave in dual-stack network environment. This experiment was conducted on isolated testbed environment to ensure the malware will not propagate into real live network. In the direction to prove the concept of this study, two malware have been selected to be used in this study, namely, Nimda variant E (Nimda.E), and Sasser variant B (Sasser.B). These worms were selected due to their special feature that can randomly assign targeted victims' IP addresses based on MAC information from the infected node. Each worm was released in different scenario to make certain each worm would not affect each other. Each data from each scenario will be repeated three times as being practiced by previous researchers their study[31, 32]. The duration of data collection of each scenario is ten minutes. The network layout of the designed testbed can be found in Fig. 3.

Fig. 3 is about the network layout of the testbed used in this experiment. This designed was innovated from a study conducted by [33] where almost a similar study was conducted. Each node is configured as stated in the Figure 3. Node 1 will be installed with a packet sniffer tool and this node will capture all traffic through the router gateway link. Switch also being configured so that each packet going through router gateway link is copied and send to Node1 to be collected. Node 2 and Node 3 work as client in a dual-stack network environment. On each scenario, malware is released on Node2. Some loopback is configured in the router to simulate as if this router is connected to other IPv4 and IPv6 networks.

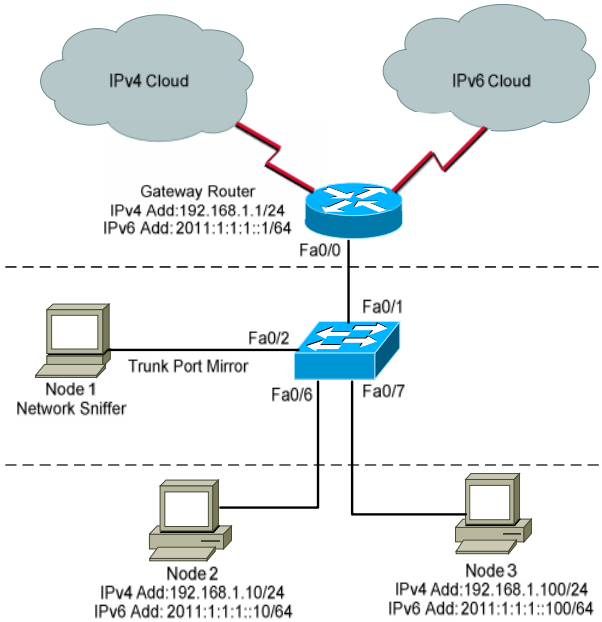

Fig. 3 Testbed network layout

The procedure of this experiment is as the following:

S1. All computers, a switch and a router are cleaned. All configurations into those computers, the router and the switch are restored. The connectivity of each node is tested to ensure everything is working as expected

S2. Leave the computers for two minutes to ensure the network traffic has become stable.

S3. The network sniffer tool is activated to capture idle network traffic pattern.

S4. After 10 minutes, the sniffer tool is stopped and the log is saved.

S5. Just after that, the malware is released from Node 2.

S6. The network sniffer tool is restarted to capture the packet through the gateway link.

S7. The duration of each scenario will last for 10 minutes.

S8. Plug out all cables connected to computer to stop the experiment session and save the network traffic log from Node1 for further analysis.

S9. Before starting the next experiment session, all computers must be formatted. The switch and the router must be reloaded with the original configuration to ensure it is free from malware infection in operating system and in its memory.

V. Result And Analysis

In this analysis, the protocol used in each scenario will be identified and the frequency of all packets going through the gateway within one second will be visualized. Since this experiment is using a dual-stack network environment, both IPv4 and IPv6 protocols will be analyzed. As discussed in the previous section, at this point of time there are 3 scenarios, namely, an idle network, a nimda infection and a sasser infection network. Each scenario also has 3 different datasets.

Each dataset on each scenario will be analyzed individually. After all data being analyzed, the combination of results on each scenario will be presented towards the end of this section.

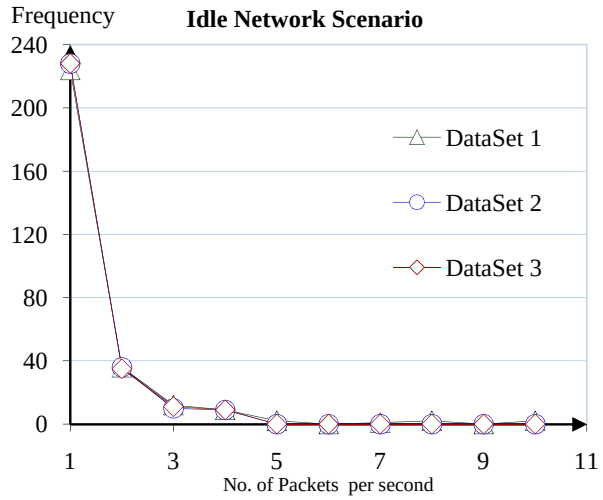

Fig. 4: Frequency number of packet sent in one second in idle network

A. *Idle Network Scenario*

This data are captured within 10 minutes duration after a network is considered stable. In this scenario, there is no specific activity but still some packets are still flowing out and being captured. Typically, Spanning-Tree Protocol (STP) and Dynamic Trunking Protocol (DTP) are being used for network convergence.

Fig. 4 shows the frequency of number of packets being released within one second during an idle network. This figure is extracted from 3 datasets. The line with triangle symbol represents data from DataSet 1. The line with circle symbol represents data from DataSet 2 while DataSet 3 is presented by the line with diamond symbol. Based on this data, the data captured are consistent among the three datasets. It can be assumed that the traffic pattern in idle network is following exponential distribution. The mode is only one packet released on the network within one second. The frequency goes down exponentially as the number of packet released per second increased.

B. *Nimda Infection Scenario*

In this second scenario, a Nimda worm will be released on the testbed. These data are captured from 10 minutes duration after a network is considered infected by the worm. The following figure shows data captured in three different scenarios.

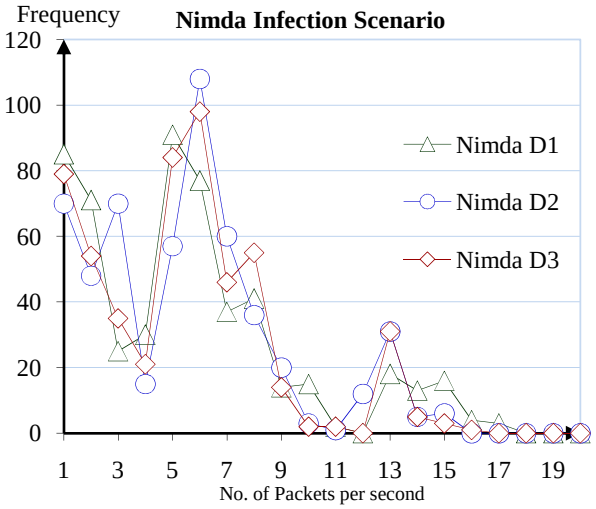

Fig. 5 Frequency number of packet released in one second in nimda infected network

Fig. 5 shows the frequency of the number of packets being released within one second after the network being infected by a nimda worm. This figure is extracted from three datasets. The line with triangle symbol represents data from dataset1 (Nimda D1). The line with circle symbol represents data from dataset2 (Nimda D2) while dataset3 (Nimda D3) is presented by the line with diamond symbol. The data represent the combination of both IPv4 and IPv6 network traffic on the dual-stack network testbed. Later, data from the three dataset will be combined. Then, the data will be classified based on the protocol used either IPv4 or IPv6 protocol for further analysis.

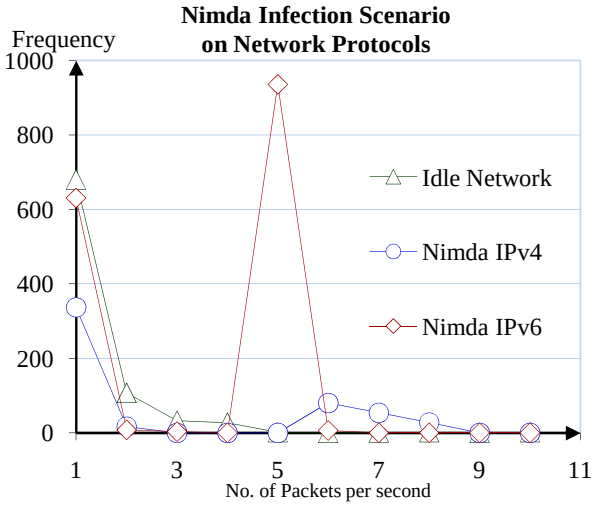

Fig. 6 Extracted data from the combined nimda infected datasets

Fig. 6 shows the extracted data from the combined data from 3 datasets as depicted in Figure 5 according to respective protocols.

From the combined data, IPv4 and IPv6 packets have being distinguished to analyze packets on different network protocol. The line with diamond symbol represents data for packets being released by using the IPv6 protocol. The line with circle symbol represents data from packet being released by using the IPv4 protocol while the idle network is represented by the line with triangle symbol. The idle network line is represented the combined idle network datasets.

From the Fig. 6, after a network being infected by a nimda worm it can be seen that the highest frequency of IPv6 packets being released is 5 packets per seconds. In terms of protocols, most of IPv6 packets were using UDP protocols such as Link-local Multicast Name Resolution (LLMNR) and DNS.

Meanwhile, the highest frequency of IPv4 packets being released is actually 1 packet per second. However, there is another peak which is 6 packets per seconds. This second peak shows there is a possibility of worm activity as this pattern was not occurred in the idle network dataset. In terms of protocols, most of IPv4 packets were using TCP protocols such as HTTP, and Server Message Block (SMB).

When comparing both IPv4 and IPv6 protocols, the detection of worm activity in IPv6 should be different from IPv4 network environment. The protocol used is different as well as the highest frequency of packet being released by each IP protocols are also different. Notice that the peak of Nimda_IPv6 is significantly high. This is probably because of the frequency packets are represented by the combination of packets being released for network convergence and being released by the worm itself.

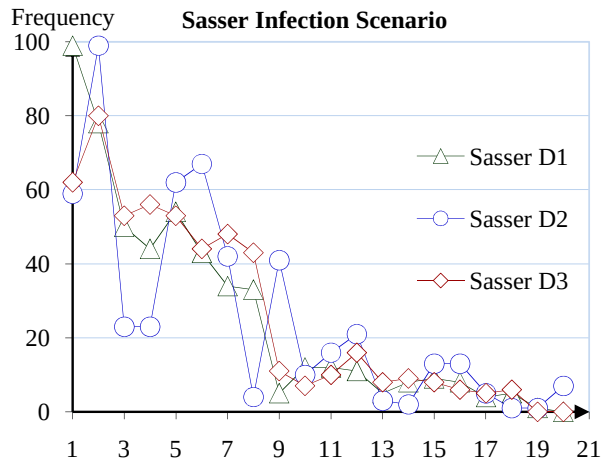

Fig. 7 Frequency number of packet released in one second in Sasser infected network.

C. *Sasser Infection Scenario*

In this third scenario, a Sasser worm will be released on the testbed. These data are captured from 10 minutes duration after a network is considered infected by the worm. The following Fig. 7 shows data captured in three different scenarios.

Fig. 7 shows the frequency of number of packet released in one second within after the network being infected by a sasser worm. The line with triangle symbol represents data from dataset1 (Sasser D1). The line with circle symbol represents data from dataset2 (Sasser D2) while dataset3 (Sasser D3) is presented by line with diamond symbol. In these data, both IPv4 and IPv6 packets are being captured in the dual-stack testbed. For further analysis, these three dataset were combined and both IPv4 and IPv6 packets will be distinguished to analyze each protocol individually. Fig. 8 shows the extracted data from the combined sasser infected datasets.

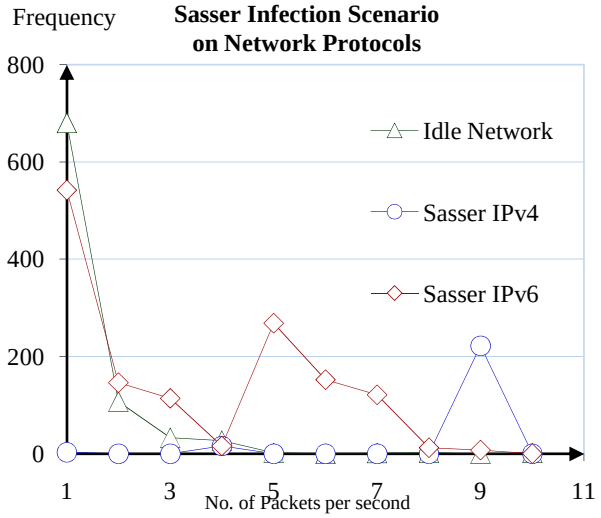

Fig. 8 Extracted data from the combined sasser infected datasets

Fig. 8 shows the extracted data from the combined sasser infected network datasets. In this figure, both IPv4 and IPv6 packets were isolated to analyze the protocol individually. The line with diamond symbol represents data for packet being released by using the IPv6 protocol. The line with circle symbol represents data for packet being released by using the IPv4 protocol while the idle network is represented by the thick line with triangle symbol.

Based on Fig. 8, after a network being infected by sasser it can be seen that the highest frequency of IPv6 packets being released is 1 packet per second which is consistent with the idle network scenario. At the same time, there is another peak point which is 5 packets per seconds. This peak point possibly shows there is an anomaly activities occurred in the network as the second peak does not occur in the idle network scenario. In terms of protocols, most of IPv6 packets were using ICMPv6 and UDP protocols such as Link-local Multicast Name Resolution (LLMNR) and DNS.

Meanwhile, the highest frequency of IPv4 packets is 1 packet per seconds which is consistent with the idle network scenario. Meanwhile, there is another peak point which is 9 packets per seconds. This peak point could represent an anomaly activities occurred in the network as the second peak does not occur in the idle network scenario.

In terms of protocols, most of IPv4 packets were using ICMP and TCP protocols such as HTTP and Server Message Block (SMB).

When comparing both IPv4 and IPv6 scenarios, the detection of worm activity in IPv6 should be different from IPv4 network environment. The protocols being used are different as well as the highest frequency of packet being released by each IP protocols are also different.

VI. Findings and Discussion

Based on result gathered from the experiment, it can be seen that malware is behaving differently on different IP protocols. Ideally, the number of packet being released on an idle network follows an exponential distribution. However, once a network is being infected by a worm, the frequency distribution of packets changes accordingly. Both scenarios show that the highest frequency of packets being released and the protocols used by each IP protocol are different. The following table shows the comparison between different scenarios:

TABLE II
Comparison between Nimda and Sasser Infection Scenario

Comparison Element	Nimda Infection Scenario		Sasser Infection Scenario	
	IPv4	**IPv6**	**IPv4**	**IPv6**
IP address	32 bits	128 bits	32 bits	128 bits
Most Protocol Used	TCP (HTTP & SMB)	UDP (LLMNR,DNS)	TCP (HTTP & SMB)	UDP (LLMNR,DNS)
Highest frequency packet released	6 pkts/sec	5 pkts/sec	9 pkts/sec	5 pkts/sec

Table II compares and summarizes the experimental results from two different scenarios, namely, Nimda and Sasser infection scenarios. From the table, the IP address used on both protocols are different where the IPv4 protocol uses 32 bits while the IPv6 protocol uses 128bits for IP addresses. What is more, it can be seen that in both scenarios most of IPv4 packets were using TCP protocols whereas IPv6 packets were using UDP protocols. In terms of packet being released, the highest frequency peak released in IPv4 is much higher compares to IPv6 network environment. Based on these three observations, we would like to recommend that the malware detection in IPv4 network environment need to be improvised in order to detect malware activities in either dual-stack or IPv6 network environment more efficiently. The current malware detection for IPv4 network will not do well in IPv6 network environment.

VII. Conclusion

A broad use of IPv6 is eventually coming. Several transition mechanisms are used to promote migration from IPv4 to IPv6 network environment. The implementation of transition mechanisms need to be carefully designed as this study shows that malware could use the transition mechanism as a vulnerable point to propagate its attack. It can be concluded that malware behave differently in different network environment. Based on the experimental results, it has been observed that three main features are used by some intrusion detection techniques to construct their models are different between IPv4 and IPv6 protocol.

Hence, the current detection techniques need to be improvised in order to detect malware activities more efficiently in either dual-stack or IPv6 environment. For future work, the testbed network design will be extended and the use of IPv6 packet generator tools will be more realistic to imitate IPv6 network environment.

Acknowledgment

The research presented in this paper is financially supported by Ministry of Higher Education of Malaysia and the study was conducted at Faculty of Information and Communication Technology (FTMK), University of Technical Malaysia Malacca (UTeM).

References

[1] Cheng, M. *Research on network security based on IPv6 architecture.* in *Electronics and Optoelectronics (ICEOE), 2011 International Conference on*. 2011.

[2] Waddington, D.G. and F. Chang, *Realizing the transition to IPv6.* IEEE Communications Magazine, 2002. 40(6): p. 138-147.

[3] Badamchizadeh, M.A. and A.A. Chianeh. *Security in IPv6. Proceedings of the 5th WSEAS International Conference on Signal Processing*. 2006. Istanbul, Turkey.

[4] Zheng, Q., T. Liu, X. Guan, Y. Qu, and N. Wang, *A new worm exploiting IPv4-IPv6 dual-stack networks*, in *Proceedings of the 2007 ACM workshop on Recurring malcode*. 2007, ACM: Alexandria, Virginia, USA.

[5] Kamra, A., H. Feng, V. Misra, and A.D. Keromytis. *The effect of DNS delays on worm propagation in an IPv6 Internet*. in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. 2005.

[6] Bellovin, S.M., B. Cheswick, and A.D. Keromytis, *Worm propagation strategies in an IPv6 Internet.* LOGIN: The USENIX Magazine, 2006. 31(1): p. 70-76.

[7] Computer Economics, *Annual Worldwide Economic Damages from Malware Exceed $13 Billion*. 2007.

[8] Bellovin, S.M., *Perceptions and Reality.* Security & Privacy, IEEE, 2010. 8(5): p. 88-88.

[9] Stewart, J., *Behavioural malware analysis using sandnets.* Computer Fraud & Security, 2006. 2006(12): p. 4-6.

[10] Jiann-Liang, C., C. Yao-Chung, and L. Chien-Hsiu. *Performance investigation of IPv4/IPv6 transition mechanisms.* in *Advanced Communication Technology, 2004. The 6th International Conference on*. 2004.

[11] Karresand, M., *A proposed taxonomy of software weapons.* No. FOI, 2002.

[12] Robiah, Y., S.S. Rahayu, M.M. Zaki, S. Shahrin, M.A. Faizal, and R. Marliza, *A New Generic Taxonomy on Hybrid Malware Detection Technique.* Arxiv preprint arXiv:0909.4860, 2009.

[13] Zulkiflee, M., M.A. Faizal, I.O. Mohd Fairuz, A. Nur Azman, and S. Shahrin, *Behavioral Analysis on IPv4 Malware in both IPv4 and IPv6 Network Environment.* International Journal of Computer Science and Information Security (IJCSIS), 2011. 9(2).

[14] Chen, Z. and C. Ji, *An information-theoretic view of network-aware malware attacks.* 2008.

[15] Cliff, C.Z., T. Don, G. Weibo, and C. Songlin, *Advanced Routing Worm and Its Security Challenges.* Simulation, 2006. 82(1): p. 75-85.

[16] Zesheng, C. and J. Chuanyi, *Optimal worm scanning method using vulnerable host distributions.* International Journal Security Network, 2007. 2(1/2): p. 71-80.

[17] McHugh, J., *Intrusion and intrusion detection.* International Journal of Information Security, 2001. 1(1): p. 14-35.

[18] Sangkatsanee, P., N. Wattanapongsakorn, and C. Charnsripinyo, *Practical real-time intrusion detection using machine learning approaches.* Computer Communications, 2011. 34(18): p. 2227-2235.

[19] Antonis, P., P. Michalis, and P.M. Evangelos, *Improving the accuracy of network intrusion detection systems under load using selective packet discarding*, in *Proceedings of the Third European Workshop on System Security*. 2010, ACM: Paris, France.

[20] Li, Z., Y. Gao, and Y. Chen, *HiFIND: A high-speed flow-level intrusion detection approach with DoS resiliency*. Computer Networks, 2009. 54(8): p. 1282-1299.
[21] Mohd Faizal, A., *Enhanced Fast Attack Detection Technique For Network Intrusion Detection System.* 2009, Phd Thesis at Universiti Teknikal Malaysia Melaka (UTeM).
[22] Labib, K. and R. Vemuri, *NSOM: A Real-Time Network-Based Intrusion Detection System Using Self-Organizing Maps.* Networks and Security, 2002.
[23] Su, F., Z.-w. Lin, and Y. Ma, *Modeling and analysis of Internet worm propagation.* The Journal of China Universities of Posts and Telecommunications, 2010. 17(4): p. 63-68.
[24] Okamura, H., H. Kobayashi, and T. Dohi. *Markovian modeling and analysis of Internet worm propagation.* in *Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on*. 2005.
[25] Ting, L., G. Xiaohong, Z. Qinghua, and Q. Yu, *A new worm exploiting IPv6 and IPv4-IPv6 dual-stack networks: experiment, modeling, simulation, and defense.* Network, IEEE, 2009. 23(5): p. 22-29.
[26] Zagar, D., K.i. Grgic, and S. Rimac-Drlje, *Security aspects in IPv6 networks implementation and testing.* Computers & Electrical Engineering, 2007. 33(5-6): p. 425-437.
[27] Bruce J, N., *An introduction to investigating IPv6 networks.* Digital Investigation, 2007. 4(2): p. 59-67.
[28] Qiao, P. and P. Changxing, *Distributed sampling measurement method of network traffic in high-speed IPv6 networks.* Journal of Systems Engineering and Electronics, 2007. 18(4): p. 835-840.
[29] Caida. *Anonymized 2011 IPv6 Day Internet Traces.* 2011 [cited; Available from: https://data.caida.org/datasets/passive-2011-ipv6day/.
[30] Zolkipli, M.F. and A. Jantan. *Malware Behavior Analysis: Learning and Understanding Current Malware Threats.* in *Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on*. 2010.
[31] Rahayu, S.S., Y. Robiah, S. Shahrin, M.M. Zaki, M.A. Faizal, and Z.A. Zaheera, *Advanced Trace Pattern For Computer Intrusion Discovery.* Arxiv preprint arXiv:1006.4569, 2010.
[32] Robiah, Y., S.S. Rahayu, S. Sahib, M.M. Zaki, M.A. Faizal, and R. Marliza. *An improved traditional worm attack pattern.* in *Information Technology (ITSim), 2010 International Symposium in*. 2010.
[33] Liu, T., X. Guan, Q. Zheng, and Y. Qu, *A New Worm Exploiting IPv6 and IPv4-IPv6 Dual-Stack Networks: Experiment, Modeling, Simulation and Defense*. 2009, IEEE Network.

Zulkiflee Muslim, a Senior Lecturer at University of Technical Malaysia Melaka (UTeM). He earned MSc. in Data Communication and Software from University of Birmingham City, UK and BSc. in Computer Science from University of Technology Malaysia. He has professional certifications: CCNA, CCAI, CFOT and IPv6 Network Engineer Certified.