A Secure and Efficient Model for Network Defensive Systems

C.-H. Lin, F.-C. Jiang, Y.-L. Huang, C.-H. Huang
Department of Computer Science and Information Engineering,
Tunghai University, Taichung, Taiwan, ROC
Computer Center Hsiuping Institute of Technology, Taichung, Taiwan, ROC
{chlin,admor,g942803}@thu.edu.tw, chhuang@mail.hit.edu.tw

ABSTRACT.

Using conventional technology, a network administrator has to do multifarious work to adapt the security system to block the intruders. In order to make a network defensive system more sensitive, automatic and reactive timely and correctly upon intrusions, we propose a secure and efficient model for network security in this paper. We make use of a dispatcher and both of the characteristics of firewall and intrusion detection to against attacks. Incoming packets are distributed averagely to two or more firewalls, mitigate the load of firewalls, and forward packets in different route successfully. An experimental model has been set up for verifying the novel approach we proposed. By conducting a series of experiments, we can see that the proposed model is advantageous for both of security and efficiency. The dispatcher is responsible for the even partition of all incoming packets between firewalls. The intrusion detection system can detect the bulk of intrusion or attack behavior. Based on IDS alert logs, firewalls can update their rules automatically to deny the attacker's packets.

Keywords: Network defensive system; Firewall; Network dispatcher; Intrusion detection system; Dispatcher; Self-adapting rule system

1: INTRODUCTIONS

Nowadays, information systems become more and more convenient but more dangerous for the progress of intruding methods. It is an important research subject how to construct a network security system correctly. Usually, we use two technologies to build a network security system: firewalls and intrusion detection systems (IDS) [1, 3]. The former is located at the key point of a network, which is the front line to filter the incoming or outgoing packets; and the latter is to analyze data and launch alarm of intrusion.

However, these sub-systems have their own constraints inherently. Firstly, a firewall has to inspect all of the packets, and then it may result in an intolerable delay. To improve the filtering efficiency, hardware-based firewall approach has been adopted to meet the high-throughput needs in some applications. However, it is more expensive. Besides, when the packet flow increases abnormally, e.g. suffering from DDoS attacks, the firewall still becomes the bottleneck. For these reasons, we try to use multiple software firewalls. This will be cost-effective and risk-distributed. Secondly, under normal cases, most of the IDSs only provide an alarm of attack and the network administrator has to adapt the firewall rule manually. But it is a multifarious work. Therefore, it is worth to propose a defensive system that integrates intrusion detection with firewall and that has good efficiency and security.

In this paper, we propose a defensive system to fulfill the above idea. We make use of a dispatcher and both of the characteristics of firewall and intrusion detection to against attacks. Incoming packets are distributed averagely to two or more firewalls, mitigate the load of firewalls, and forward packets in different route successfully. By conducting a series of experiments, we can see that the proposed system is a feasible solution for both of security and efficiency. The proposed defensive system is composed of the following parts:

1.1: Firewall dispatcher

As the network activities growing up, firewall often becomes a bottleneck of network communication. Some attacking ticks, like DDoS, will make the problem more serious. To remedy the problem, we use the concept of firewall clusters. A cluster of firewalls would be organized to provide better filtering performance, instead of traditional single-point firewall framework. By using this idea, it becomes very flexible to add extra firewalls for performance consideration.

1.2: Self-adaptive rule system

Although an intrusion detection system can provide the attack alarm, the administrator still has to tune the firewall rule by himself. It would be minute and complicated and the chance of just-in-time defense would be lost. If the intrusion detection system can dynamically and automatically changes the firewall rules in order to block the attack as quickly as possible

when alerts occur. It would make the defensive system more intelligent responding to attacks.

The rest of the paper is organized as follows. In Section 2, the basic idea of firewall is briefly described. In Section 3, briefly review the intrusion detection system. In Section 4, we give our proposed system architecture and the experiment design. In Section 5, the experimental results are present. Finally, we have a conclusion.

2: FIREWALL

Firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized users from accessing private networks connected to the Internet, especially intranets. All messages incoming or outgoing the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. There are several types of firewall techniques:

2.1: Packet filter:

Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.

2.2: Application gateway:

Applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but can impose performance degradation.

2.3: Circuit-level gateway:

Applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

2.4: Proxy server:

Intercepts all messages incoming or outgoing the network. The proxy server effectively hides the true network addresses.

In practice, many firewalls use two or more of these techniques in concert. A firewall is considered as a front line of defense in protecting private information.

3: INTRUSION DETECTION SYSTEM

An intrusion detection system (IDS) inspects all inbound and outbound network activities and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. There are several ways to categorize an IDS:

3.1: Misuse detection vs. anomaly detection:

In misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against. In anomaly detection, the system administrator defines the baseline, or normal, state of the network traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.

3.2: Network-based vs. host-based:

In a network-based system, or NIDS, the every individual packet flowing through a network is analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host.

3.3: Passive vs. active:

In a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In an active system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source.

Though both are related to network security, an IDS differs from a firewall in that a firewall looks out for intrusions in order to stop them from happening. The firewall limits the access between networks in order to prevent intrusion and does not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system.

4: SYSTEM ARCHITECTURE AND EXPERIMENT DESIGN

In this section, we propose the system architecture and our experiment design. Since internal host uses the public IP address but not private IP address, we design our system architecture as Figure 1.

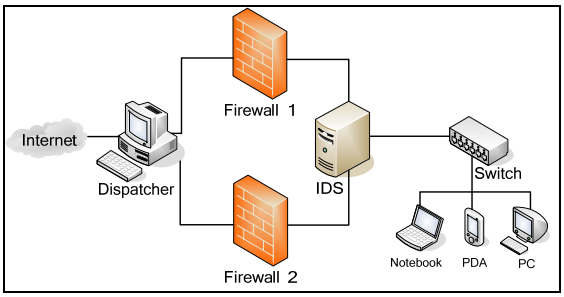

Figure 1. System architecture

The IP addresses assigned for our experiment are listed as follows:

- 140.128.98.80/255.255.255.240
- 140.128.98.96/255.255.255.240

Before conducting the experiments, we try two modes: the bridge mode and the static routing IP mode, and we found that the later can fit our requirement.

4.1: Bridge mode

In this case, we use bridge mode for each firewall, IDS and Dispatcher, respectively. We found that, using this mode, it has to start "promiscuous mode" and it will make the packets in an endless cycle between the front switch and the firewalls, shown as Figure 2. Therefore, we do not use the bridge mode.

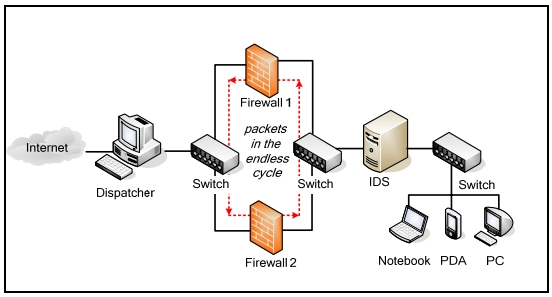

Figure 2. Using bridge mode

4.2: Static Routing mode

In this mode, we use static routing on Dispatcher, IDS and firewalls, and use private IP for Dispatcher and IDS. The static routing profile is depicted in Figure 3. This mode can meet our requirements to conduct the following experiments.

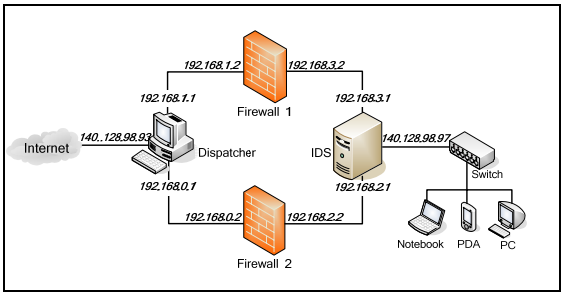

Figure 3. Using static routing mode

5: EXPERIMENTAL RESULTS

To show that the proposed system is feasible, we conduct a series of experiments using the static routing mode. We have some observations from the experimental results.

Observation 1: IDS should be separated from and set behind the firewalls.

This observation includes two experiments, Experiment 1 and Experiment 2, and the system architectures are shown in Figure 4 and Figure 5, respectively. External attackers launch attacks through the Internet, using the famous "nmap" [4] as the attacking tool, to internal hosts.(IP Address: 140.128.98.98) We record the attacking information (e.g. system usage rate and packet flow, etc) and start to scan the port the IP opened as "%nmap -sT 140.128.98.98" at the 5th second. After every attack we take 0.5 seconds of rest and continue to conduct another scanning attack. Finally, we terminate the attacks at the 175th seconds, and stop recording data at the 180th seconds.

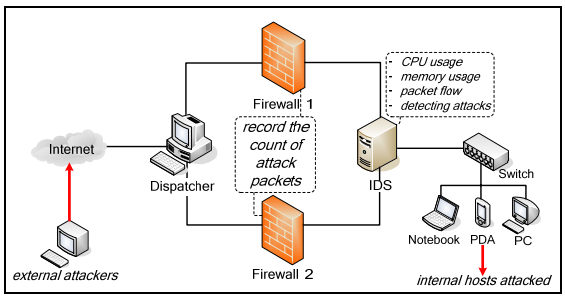

Figure 4. System architecture for Experiment 1

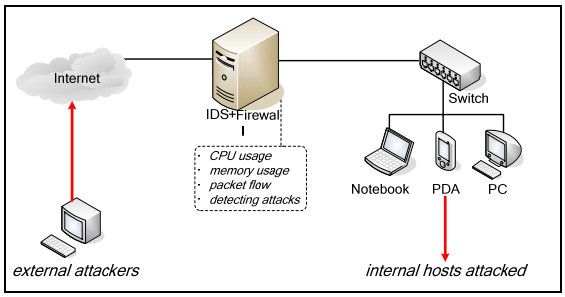

Figure 5. System architecture for Experiment 2

Experiment 1:

As shown in Figure 4, we separate the IDS from and set it behind the firewalls and record the information. We record the system usage rate, the attacking packets, the intercepted attacks and packet flow between IDS and two firewalls. Then the external attacker starts

"nmap" scanning attacks to each internal host every 2 seconds.

Experiment 2:

As shown in Figure 5, we integrate the IDS and firewall into the same host. The conditions are the same as those of Experiment 1. The external attacker launches attacks to the internal hosts every 0.5 second, starting from the 5th second to the 175th second.

After the experiments, we get the statistics of IDS packets flow, IDS system usage rate, packets detected by the IDS and blocked packets by the firewall.

In Experiment 1, the IDS sends notice to the firewalls, on detecting the attacking packets. The firewall could update the rules accordingly, and thus the next incoming attacking packets could be blocked by the firewall immediately.

Therefore, the detecting workload of the IDS can be decreased dramatically. And hence, the efficiency of IDS operation obtains a large-scale improvement due to the absence of evil packets blocked by the firewalls in advance.

In Experiment 2, the IDS and firewall are put on the same host. On getting attacking packets, the libpcap [5] would intercept the packets and sends them to the IDS (snort) for analysis. And then, the IDS notices the firewall to block. It wastes the resource of the host because the IDS just handles the unnecessary work. For the same reason, it can protect the insiders when the firewall is set behind the IDS, while the IDS becomes an encumbrance.

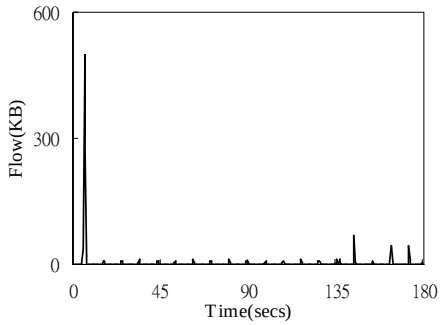

Figure 6. Packet flow from the IDS for Experiment 1

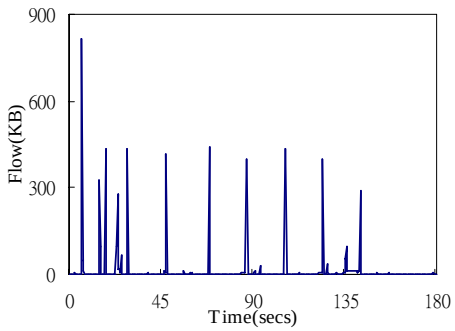

Figure 7. Packets flow from the IDS for Experiment 2

Comparing the results of Experiment 1(Figure 6) and Experiment 2(Figure 7), we can see that the packets enter the IDS in Experiment 1 decreases significantly. Because the physical locations of firewalls and IDS are interchanged, they have different result (Figure 8(a)(b)). For the most attacking software, when the attacked host does not have any response, it may give up the subsequent attacks. On the other hand, to depress the attacker's intention can also be viewed as a successful approach of improving system security. From Figure 9, the experimental evidence has shown that the number of attacking packets is reduced by using the system architecture in Experiment 1.

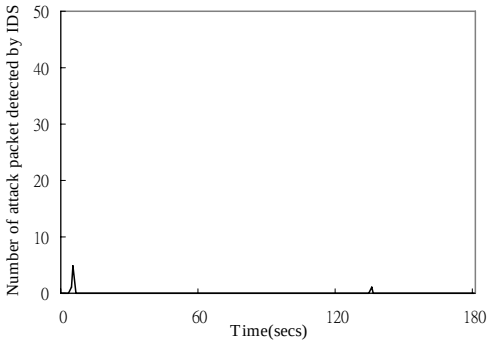

Figure 8. (a) Packets detected by the IDS for Experiment 1

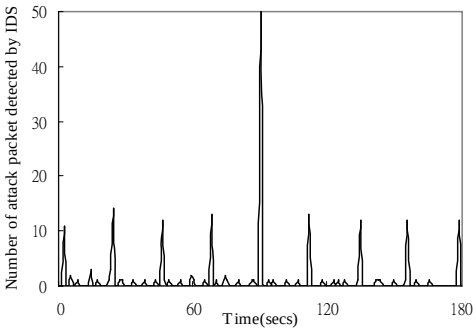

Figure 8. (b)Packets detected by the IDS for Experiment 1

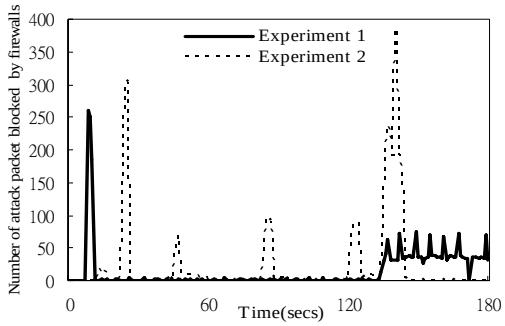

Figure 9. Packets blocked by the firewall

Now, we would try that whether it has better performance for legitimate packet flow using a dispatcher and multiple firewalls. We design the Experiment 3(Figure 10) and Experiment 4(Figure 11). To distribute the incoming packets among the multiple firewalls, we apply the round-robin algorithm and "iproute2", of which the most important are ip and traffic control.

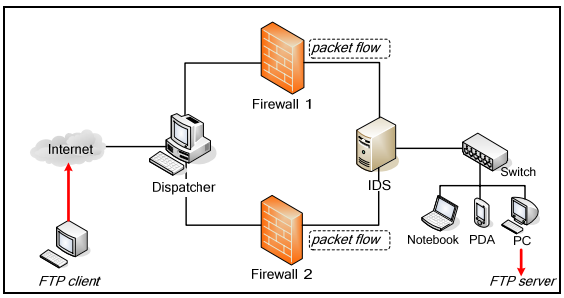

Figure 10. System architecture for Experiment 3

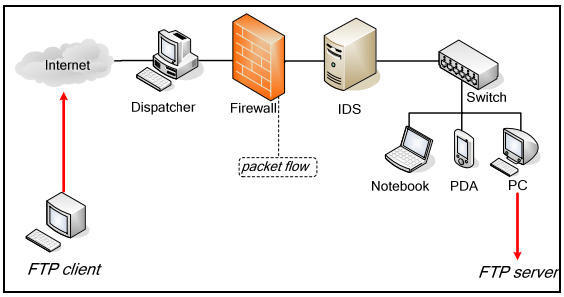

Figure 11. System architecture for Experiment 4

Observation 2: Using a dispatcher has better performance.

We let one of the hosts Serv-U 2.5 [6] to be an FTP server, and allow the external users to connect and to download files. There are 50 files in the FTP server for the experiments. Besides, we let an external user to use the CuteFTP Pro 3.0 [7], set the connection numbers up to 20 and download files by 4 parts.

Experiment 3:

In Experiment 3, we use three external FTP clients, and let them download the 50 files from the FTP server at the same time. It costs 126 seconds to complete. The analysis of packet flow is shown in Figure 12.

Experiment 4:

The conditions are same as in Experiment 3, but we use only one firewall. When we let the three external FTP clients download the 50 files at the same time, it costs 188 seconds to complete. The analysis of packet flow is shown in Figure 13.

Judged from the comparisons between Experiment 3 and Experiment 4, our proposed defensive mechanism has also shown that the excellent improvement figure can be up to 33% for legitimate packet flow in transfer-time performance.

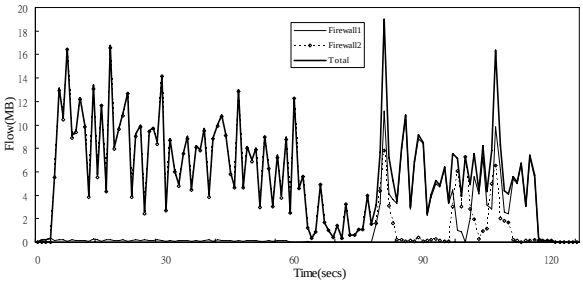

Figure 12. Packet flow on firewall on Experiment 3

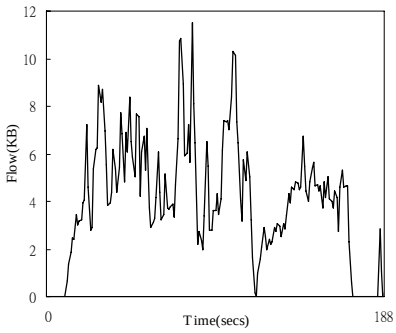

Figure 13. Packet flow on firewall on Experiment 4

In additional to the above experiments and observations, we also conduct several attacks to our system. It can be seen that our system can depress the types of attacks as shown in Table 1.

6: CONCLUSION

In this article, we propose a network defensive system that incorporates intrusion detection with multi-firewalls and good efficiency and security has been obtained. We make use of a dispatcher and both of the characteristics of firewall and intrusion detection to

depress attacks. With the location interchange of firewalls and IDS, the experimental data has verified that our proposed approach can alleviate the bottleneck at IDS, and speedup the packet-flow rate for the whole system. Adopting multi-firewalls model, the vast of incoming packets can be distributed uniformly to two or more firewalls, mitigate the load of firewalls, and forward packets in different route successfully. By conducting a series of experiments, we can see that the proposed system is a feasible solution for both of security and efficiency.

TABLE I. ATTACK TO THE PROPOSED SYSTEM

Attack	Types of attacking applications	Blocked ?
Scan	Nmap	✓
	Hackbot	✓
Eavesdrop packets	Ethereal	✗
System hole	Unicode file system fault	✓
	Double decode	✓
	WEB-IIS cmd.exe access	✓
	IIS buffer overflow	✓
Backdoor	NetBus	✓
DoS	UDP protocol flooding attack	✓
	ICMP signal flooding attack	✓
	Smurf	✓

ACKNOWLEDGEMENT

This research is partially supported by the National Science Council, Taiwan, by contract no. NSC 94-2213-E-029-001.

REFERENCES

[1] U. Lindquist and E. Jonsson, “How to Systematically Classify Computer Security Intrusions,” Proceedings IEEE Symposium Research in Security and Privacy, Oakland, CA, 1997.

[2] K. Jackson, M. C. Neumann, D. Simmonds, C. Stallings, J. Thompson and G. Christoph, “An Automated Computer Misuse Detection System for UNICOS,” Proceedings of the Cray Users Group Conference, Tours, France, 1994.

[3] Enterasys Networks, “Intrusion Detection System: Hackers Are Getting Smarter,” Enterasys Networks, 2001.

[4] nmap, “Nmap - Free Security Scanner for Network Exploration & Security Audits,” http://www.insecure.org/nmap/.

[5] libpcap, http://sourceforge.net/projects/libpacp/.

[6] Serv-U, http:// www.serv-u.com/.

[7] CuteFTP, http://www.cuteftp.com/.

[8] Andre Zuequete, “Improving the Functionality of SYN Cookies,” Proceedings of 6th IFIP Communications and Multimedia Security Conference, pp. 57-77, Sep. 2002.

[9] John Ran, Anonymous, “Maximum Linux Security, 2nd Edition,” published by Sams ,ISBN: 067-2321-34-3, June 2001.

[10] Rob Flickenger, “Linux Server Hacks,” published by O’REILLY, ISBN: 986-7794-19-2, Jan. 2003.

[11] Michael D. Baure, “Building Secure Servers with Linux,” published by O’REELLY, Oct. 2002.

[12] Brian hatch, James Lee and George Kurtz, “Hacking Linux Exposed: Linux Security Secrets & Solutions,” published by McGraw-Hill, ISBN: 0-07-213140-3, 2001.

[13] Joel Scambray and Stuart McClure, “Hacking Exposed Windows 2000: Network Security Secrets & Solutions,” published by McGraw-Hill, ISBN: 0-07-219262-3, 2001.

[14] Raven Alder, Jacob Babbin, Adam Doxtater, James C. Foster, Toby Kohlenberg and Michael Rash, “Snort 2.1 Second Edition,” Published by Syngress, ISBN 1-931836-04-3, 2004.