Development of A Packet Simulator for Performance Test of Information Security System

Wankyung Kim[1], Wooyoung Soh[1] and Jason Sigfred[2]
[1]Department of Computer Engineering, Hannam University
e-mail : {wankk12, wsoh}@hannam.ac.kr
[2]Surigao Sur Polytechnic State College, Cantilan Campus, Philippines
e-mail : jasonseril@hotmail.com

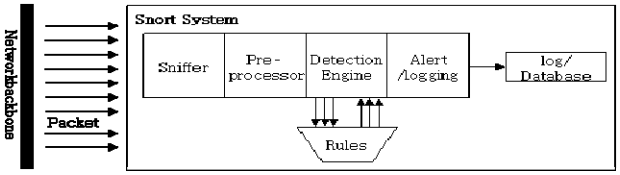
Abstract Development of information security system is brought by problem by the development of network environment, and the need of equipment for performance test, but performance test equipments are expensive and difficult to use. Therefore, we need an environment which can develop a performance test for an information security system. In this paper, the design and implementation of an APS(Attack Packet Simulator) extracts the attack information from Snort rule and creates an attack information in the Database using the extracted information. Stored information in the database creates and transmits the packets which are analyzed for comparing the results to other systems.

Keyword: Security, Packet Simulator

1. Introduction

Various Security Systems including a firewall and an intrusion detection system (IDS) have been developed to counter malicious incidents such as egress of information and illegal access on the information systems connected through computer networks.

Security systems generally are designed to operate on a network, and connected with many other information systems. Therefore, such security systems, when they are developed, need to be tested on the network environment being used for their security test and performance evaluation. It is desired that the test being done on the real network environment. However, it is usually tested in a virtual test environment (a closed network environment), due to the possible damage occurred during the test, possibly propagated through the network. It is specially



(Fig.1) Architecture of Snort

the case when the real network environment is too sensitive or important to allow any corruption of network or system during the test, or when building a real-like test environment is too expensive.

Though information security system needs a test equipment for developing a performance test, its high prices still occurs[1][2][3].

This paper presents an attack packet simulator which performs test of information security system to solve that problem. This paper also presents an environment that can verify a target system.

The remainder of the paper is structured as follow. Section 2 introduces snort, libnet and other systems. Section 3 describes the design of APS(Attack Packet Simulator) and a implementation. Section 4 analyze result of comparison with other systems. Section 5 draws conclusions and outlines future work.

Rule header	Rule option
Alert tcp any any -> 192.168.0.0/24 111	(contents:"|00 10 86 a5|";msg:"mounted access";)

(Fig.2) Snort rule

2. Related work

2.1 Snort

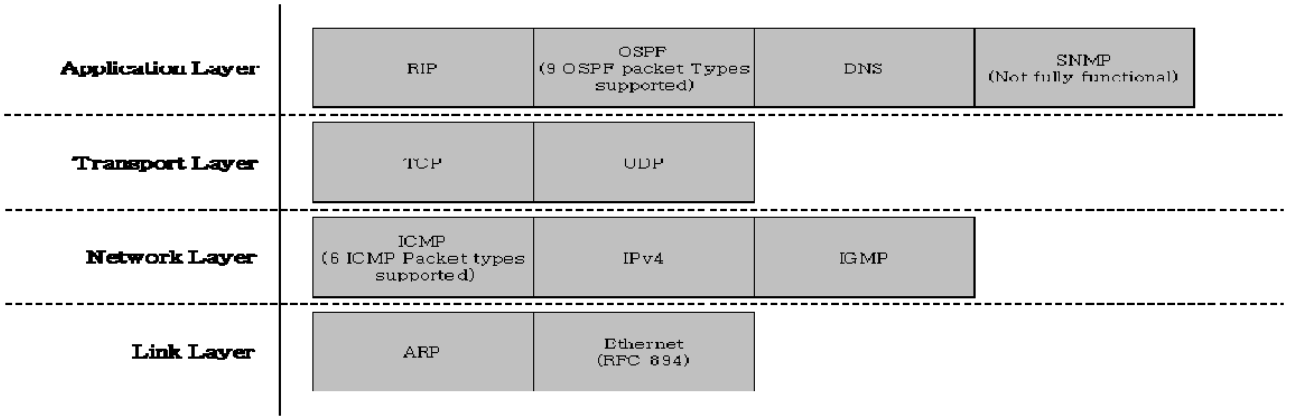
2.1.1 Architecture of Snort

Snort is a packet sniffer, packet logger and a network IDS. Snort begun to develop by a packet sniffer, act in various OS, print out packet contents by hexadecimal and print out each other network packet by equal method. A Snort is a rule-based intrusion detection system function begun to add January 1999 and 1.5 versions in December 1999 was announced. Present 2.2 versions did newly architecture and is made by code of about 75,000 lines[4].

Early Snort was no pre-treatment function and plug-in function. Snort developed including function such as improved network follow, database plug-in and pre-treatment plug-in in the course of time.

2.1.2 Snort Rule

A snort rule consists of a one line, and use back slush at end of the line when used in two lines. A snort rule consists of a rule header part and a rule option part. Rule header consists of Action, Protocol, source IP address, destination IP address, netmask, source port and destination port. Rule option consists of used segment to detect along with warning message.



(Fig. 3) Supported protocols of libnet

2.2 libnet

2.2.1 Architecture of libnet

Libnet is designed small, easy to use and efficient. Libnet's biggest purpose is to create and write packet that have portability. Libnet is consisted of code of about 8000 line in 43 file in version 1.0.2. Users can use 63 functions, and 26 packet generating routines. Libnet supports 10 protocols and (fig. 3) shows the kinds of supported protocols [5].

2.2.2 create packet using libnet

To create and transmit packet to any network, there is standard order to follow. To transmit packet to network, users must follow subsequent 5 steps:

- Network initialization
- Memory initialization
- Packet creation
- Packet Checksum
- Packet transmission

2.3 Other Systems

2.3.1 Hardware equipment

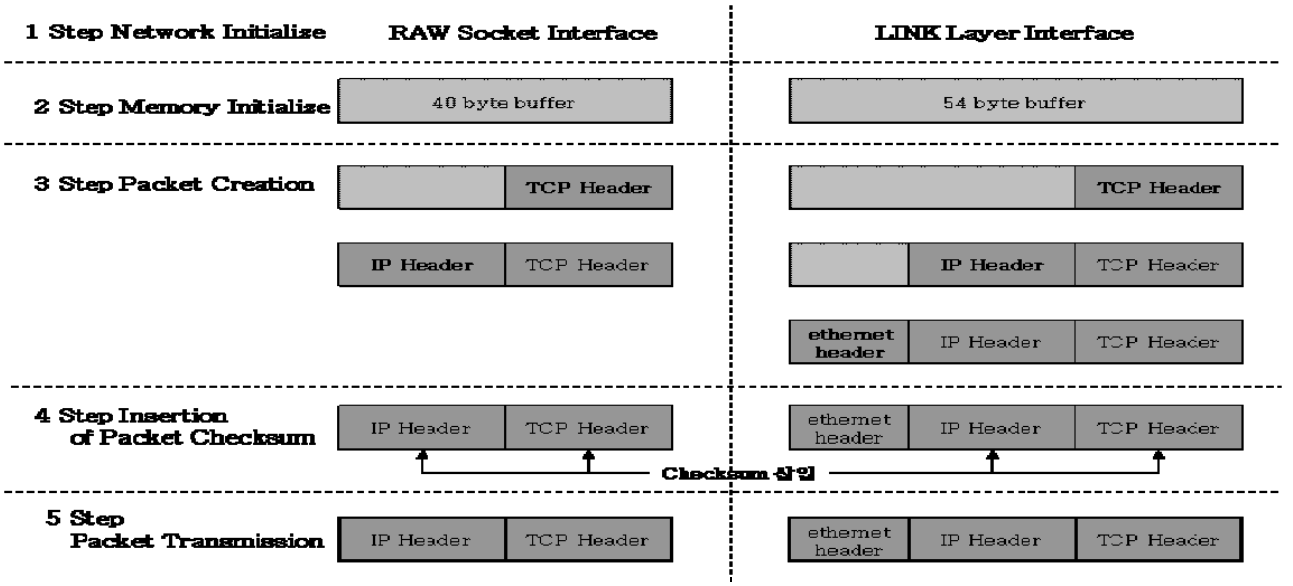
1) Test equipment of L2~L3

L3 test equipment has functions that generate traffic L2 class(Transport Layer) and L3 class(Network Layer) of Etherent, SONET, etc.. L2 and L3 are used mainly to test switching equipment or router equipment, also to test the routing performances of BGP, RIP, OSPF, etc.

Spirent company's smartbits product, IXIA company's IXIA, Agilent company's Router Tester, etc. is frequently used. Recently, it supplied equipment which generates 10G ethernet traffic. The test equipment is for testing the packet lost rate, delayed action, processing performance, etc., and it also offers partially to session connected function.

2) Test equipment of L4~L7

L4~L7 are the network security equipments, content switching equipments and test equipments that is offering support application protocols, especially for performance test of application servers. L4~L7 test the performance and process functions of services such as web, ftp, DHCP, e-mail server, gateway and security equipment. These can measure protocol examination of application level(firewall, IDS, IPS, VPN, web server, load balancer, SSL accelerator, etc.), TCP performance after the Flow Control of contents switch and encrypted packet processing performance. Most companies and individuals cannot afford the L4~L7 hardwares because it is expensive, even though it has an excellent performance.



(Fig. 4) 5 steps for packet creation and transmission

2.3.2 Software Systems

There are two kinds of system, the transmit attack information to a target system based on snort rules and the transmit attack information to a target system using TCP Dump data.

1) Snot

Snot is a random packet generator, and uses snort rule file to source of packet, and extracts information of rule file and generates traffic. Also, snot creates an attack packet after an input of network class, generates traffic, and makes hard to control the network. It uses Libnet for transmission of packet. In addition, Winpcap packet driver must be installed in Windows, on the other hand, it is available in LINUX and FreeBSD. Snot can use at parsing declaring extension variable(ex. Var HTTP_SERVER, var SERVER_ADDR) in Snort rule file[6].

2) Mucus

Mucus transmits to target system by option after doing data structure and uses these structures to create attack traffic for analyzing option on each rule as simulator that creates attack traffic to use Snort rule file. This action is done in Snot[7].

3) Packet Excalibur

Packet Excalibur is a simulator to create and transmit using TCP Dump data. It is very useful to create an attack packet as well as back-ground traffic. TCP Dump data that have an attack information is difficult to get [8].

Most of packet simulators are freeware but there is some problem that must parse whenever a packet is created. Also, it has a low detection rate of packets and

rule parsing success rate due to unapplied the latest snort version.

3. Design and implementation of packet simulator

3.1 Structure of packet simulator

The design of the attack packet simulator consists of 4 modules.

First, an attack information database creates module which collects necessary information after changing snort rules. Second, an attack information database management module updates and deletes a created attack information database. Third, an attack packet creation and transmission module creates the attack packets and transmits to target system. Lastly, a result output module shows transmitted results. This system works on Linux and uses MySql as its database.

3.2 Design and implementation of modules

1) An attack information database creation module

An attack information database creation module is for saving module through collected information by each section which separates a header and an option. It provides convenience for an attack information management as a one time parsing.

2) An attack information database management module

An attack information database management module is used when it updates and deletes a content of an attack information database.

3) An attack packet creation and transmission module

An attack packet creation and transmission module is created with a scenario file, then transmits packet to the target system. A scenario file is produced automatically when an attack packet is selected and created. This module can transmit same attack information to several systems.

4) A result output module

A result output module displays the results(such as attack name, transmission rates and time) of packet creation and transmission.

4. Result of implementation of APS(Attack packet Simulator)

In section 2 of this paper, executed comparative test with Snot and Mucus verifies the performance of the system. APS is a system that is implemented in this paper for convenience.

4.1 Result of changing of Attack Information database

A snort rule extracts the information which needs to construct a packet, utilize that a line is constructed with a header and an option. Extracted information is saved in an attack information database. It must parse to extract any information from snort rule. Because existent programs do parsing according to rule file standard of Snort 1.X version progressing parsing, parsing of rule options that is added on present 2.X version was impossible. [Table 1] shows parsing success by Snort version.

[Table 1] parsing success by Snort version

	Snort 1.8.6 (1,212)	Snort 2.2.0 (1,838)
Snot	528(43.6%)	582(31.7%)
Mucus	725(59.8%)	831(45.2%)
APS	1,100(90.8%)	1,272(69.2%)

4.2 Comparison with other Systems

It shows no differences about the packet transmission rate. In the case of Mucus using Libpcap, thus a Snot using Libnet. Implemented system is used to make function in analyzing Libpcap and Libnet. When creating packet, consummation of the packet depends on accuracy of rule file.

The existing programs have low completion rate because these program use rule file of snort 1.x version. There is a striking contrast when using snort 2.x version. [Table 2] shows the packet detection rate on each snort version.

[Table 2] packet detection rate on each snort version

	Snort 1.8.6	Snort 2.2.0
Snot	396(75.0%)	421(72.3%)
Mucus	684(94.3%)	769(92.5%)
APS	967(87.9%)	1,097(86.2%)

5. Conclusions and Future Work

By the arrival of info-age, Information-Communication developed brilliantly. Accordingly, dysfunction of

Information-Communication has become an important social problem. For the effort to minimize dysfunction it is concentrated in information security system and for the trust guarantee of information security system, performance test is an important part.

In this paper, the design and implementation of the attack packet simulator by method will do this performance test more easily and comfortably. Because the APS that implements using changing Snort rule in this paper is possible when test performance of information security system is rule based.

In the future, this system must be upgraded with a function of 3 hand shake then it is possible to apply for rule based analysis as well as Heuristic analysis. And through GUI implementation, study about that offer more convenient environment should be proceeded.

References

[1] Korea Information Security Agency, “Tracking report of Information disfunction”.

[2] Dorothy. E. Dening, “An Intrusion Detection Model,” IEEE Transactions on Software Engineering, Vol.13, No.2, pp.222-232, February 1997.

[3] P. Porras, "The Common Intrusion Detection Framework Architecture," CIDF Working Group Document, September 1999.

[4] Jay Beale, "Snort 2.0", 2003.

[5] Mike Schiffman, libnet, http://www.packetfactory.net/projects/libnet, 2002.

[6] Sniph, Snot, http://www.sec33.com/sniph/, 2001.

[7] Darren Mutz, Giovanni Vigna, Richard Kemmerer, "An Experience Developing an IDS Simulator for the Black-Box Testing of Network Intrusion Detection Systems", 2003.

[8] Jitsu, http://www.securitybugware.org/excalibur/

Authors

Wankyung Kim

Received a B.S. degree in Computer Engineering from Hannam University 2003, and M.S. degree in Computer Engineering from Hannam University 2005.

Wooyoung Soh

Received a B.S. degree in Computer Science from Jung-Ang University 1979, and M.S. degree in Computer Science from Seoul National University 1981 and Ph D. degree in Computer Science from Maryland University 1991.

In 1991 he joined the faculty of Hannam University where he is currently a professor in Department of Computer & Multimedia Engineering. His research interests include Nueral Networks, Information Security, and Computer Networking. He is a Member of KIPS, KIISC, KMMS, KIAS, and DCS.

Jason Sigfred

Received M.S. degree in Computer Engineering from Hannam University 2005.
Surigao Sur Polytechnic State College – Cantilan Campus, Philippines