**TELKOMNIKA**, Vol. 11, No. 6, June 2013, pp. 3178 ~ 3186
e-ISSN: 2087-278X



# Design of a Parallel and Distributed Network Security Simulation Platform

**Songchang Jin*[1], Songhe Jin[2], Shuqiang Yang[3], Xiang Zhu[4]**
[1,3,4]School of Computer, National University of Defense Technology, Changsha, Hunan Province, China 410073.
[2]School of Computer and Communication Engineering, Zhengzhou University of Light Industry Zhengzhou, Henan Province, China 450002.
*Corresponding author, e-mail: jsc04@126.com*, zhuxiang19881117@gmail.com, sqyang9999@126.com, jinsonghe76@126.com

***Abstract***

*With the rapid development of computer science and technology, network attack and defense has become an important research topic in the field of information security. Teaching on network attack and defense technology in universities and research institutions has a very strong experimental characteristic, but due to the special nature and the destructive characteristic of network attack and defense technology, it is difficult to carry out experiments. In this paper, we designed and implemented a parallel and distributed network security experiments platform on which experiments can be carried out through a web interface and network simulation scripts. We believe that it is also an excellent platform for teaching courses in operating systems and networking. The results of experiments in this paper show that the platform could provide the laboratory personnels with independent experimental environment, perfect experimental control functions, excellent data collection and analysis services, etc.*

***Keywords****: network security, simulation, experiment, teaching*



## 1. Introduction

With the rapid development of computer science and technology, network security is becoming increasingly important in today's internet-worked systems. With the development of internet, its use on the public networks, the number and the severity of security threats has increased significantly [1], such as worms, distributed denial of service, Trojan, etc. Processing resources of routers, servers and firewalls are limited, network attacks will result in a huge loss of network service providers, ranging from impaired quality of service to damaged physical equipments or even network paralysis [2]. And network attack and defense has become an important research topic.

Many universities and institutions are very focused on information security training of technical personnels, and building their own network security teaching experiment platforms, which focusing on the understanding of network attack and defense, verifying the type of experiments and the passive defense. However, with the progress of network technology, network attack and defense trading off and taking turns, the original passive defense platforms couldn't meet the need of the information security technology teaching [3]. On the other hand, due to network laboratory equipment is expensive, many university network security laboratory unwilling or unable to replace the new security laboratory equipments, resulting in the trainees cannot be hands-on experimental experience for a new type of intrusion, which is not conducive to the understanding and mastering of the latest technology in network security [4].

In this paper, we designed and implemented a parallel and distributed network security experiments platform on which several experiments can be carried out at the same time through a web interface and network simulation scripts. We believe that it is also an excellent platform for teaching courses in operating systems and networking.

*Received January 23, 2013; Revised April 2, 2013; Accepted April 18, 2013*

## 2. System Design

### 2.1. System Architecture

Taking into account the special and destructive characteristics of network attack experiments, networks in the simulation platform designed in this paper are divided into two parts: Control Net (also as Ctrl Net) and Experiment Net (also as Exp Net), which are shown in Figure 1.

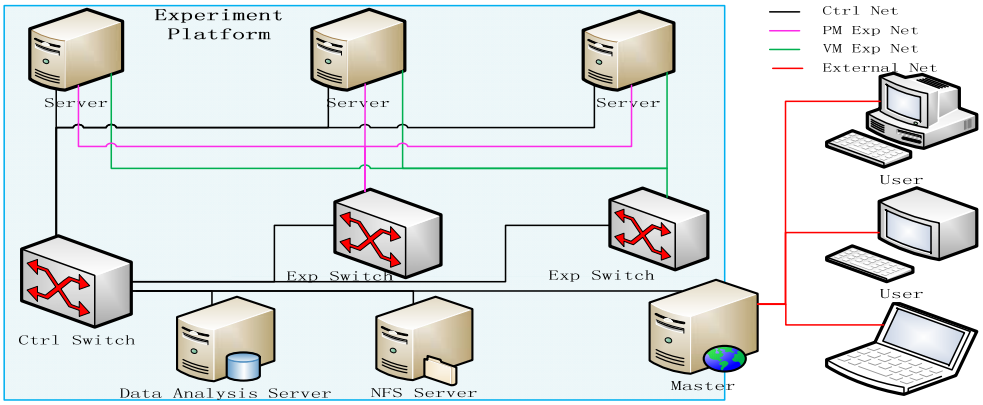


Figure 1. System Architecture

The design method of separating the control and experiment network makes the control flow and experimental data flow be independent of each other and do not interfere with each other. It gives the experiment personnels enough operating space, and at the same time, ensures the smooth transferring of control instruction from the Master to all the other nodes. In case of emergency, administrators can get the control of the experiment forcedly to prevent the platform from more serious damage, and the experiments on the platform would be with high reliability and controllability.

The platform, giving researchers and experiment personnels a wide range of environments on which to develop, debug, and evaluate their systems, consists of a set of experiment nodes (Servers), a set of switches (Ctrl Switch and Exp Switch) that interconnect the nodes, a control node (Master), which is the most important part in the platform, a NFS Server and a data analysis server. Switches are used to interconnect the experiment nodes. Experiment nodes may be servers, personal computers or other devices, such as Intrusion Detection Systems, etc. Each experiment node has three network interfaces, one of which is connected to the control network and the other interfaces are connected to the experiment network. Control network consists of Master, Servers, Switches (Ctrl Switch and Exp Switch), NFS Server and Data Analysis Server. Experiment network consists of Servers and Exp Switch. The experiment network is further divided into two types according to the machine type: VM experiment network (VM Exp Net in Figure 1) and physical machine network (PM Exp Net in Figure 1). So, the platform can provide a combined simulation environment of the virtualness and reality for laboratory personnels.

Master controls each network node apparatus, creates VLANs on the Exp Switches, and provides users with a Web-based experimental environment. Users could access the platform through the Internet, and complete the design of the experimental environment through the Web-based interface. The system will complete the environment structures according to the users' settings. The experimental environment can be recorded and repeated. NFS server records all the experiment-related resources, such as experiment scripts, experiment procedures and experiment results, etc [5].

An experiment is specified using a network simulation file or a web interface. The control software on the Master server enables multiple, separate experiments to be simultaneously run on the platform. The software isolates experiments by assigning each experiment to one or more unique Virtual Local Area Networks (VLANs) that connect together the experimental interfaces on each experiment node either using simulated bandwidth-limited

and lossy links or using LANs. By using separate VLANs, an experiment's experimental traffic is isolated from other experiments. To prevent one experiment's network traffic from interfering with that of other experiments because of insufficient internal switch or inter-switch bandwidth, the assign program is responsible for mapping an experiment's link bandwidth requirements onto the available switch resources in a manner that ensures that the experiment's bandwidth demands match available inter- and intra-switch bandwidths.

The process of swapping in an experiment consists of several steps: mapping the users' desired network topologies onto available nodes and switch resources, conFigureuring VLANs on the switches to connect the experiment nodes into the users' desired network topology, installing an initial mini filesystem kernel and root filesystem onto the experiment nodes, and then loading and running the desired operating system and software.

### 2.2. System Function

In order to carry out large scale network attack and defense experiments, analyze and evaluate effects of the attacks, we have implemented the platform with multiple-layer fidelity. Figure 2 shows the function architecture of the system. There are 3 layers in the architecture: Resource layer, Emulation layer and user layer. The management module runs through these three layers.

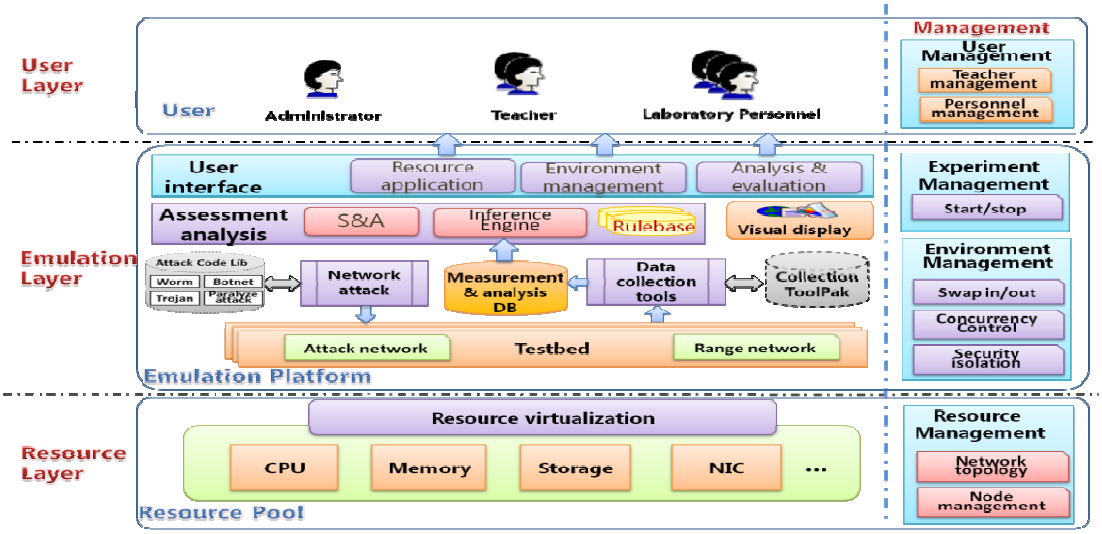


Figure 2. Function Architecture

#### 2.2.1. Resource Layer

The resource layer is responsible for building resources pool, to achieve the underlying resource virtualization, such as CPUs, memories and so on. Taking into account the demand of combining the virtual with the reality, we introduced the open source Eucalyptus [6] platform to create virtual machines automatically. The flexibility of the virtual machines enables us to create various types of virtual machines according to the actual demand.

Figure 3 shows the physical deployment of the platform. All the servers are distributed in three data centers. We have a Tianhe-1A supercomputer in the NUDT Data Center, and a total of 312 PowerLeader PT6510N servers in Antvision Data Center and Hunan Software Park Data Center. The three data centers are connected with dedicated lines.

Taking into account the configuration of the nodes in Tianhe-1A supercomputer (2 Intel Xeon X5670 6 core 2.93GHz CPU, NVIDIA Tesla M2050 GPU, 32GB Memory, 40Gbps I/O) [7,8], we built the Eucalyptus cluster on the Tianhe-1A supercomputer to provide VMs, and the left servers in the other data center provide physical experimental machines.

#### 2.2.2. Emulation Layer

The emulation layer is mainly composed of three parts: Testbed, Assessment analysis and User interface. It is the core component of the system.

## 1. Testbed

A testbed is a platform for experiments of large development projects. Testbeds allow for rigorous, transparent, and replicable testing of scientific theories, computational tools, and new technologies [9]. A testbed in this paper is a complete experimental environment which is used to test and analyze variety of network attacks. The main function of the testbed is to provide the users a variety of network topologies and experimental resources. User can create many testbeds on the platform and testbeds are physical isolated to each other by VLAN.

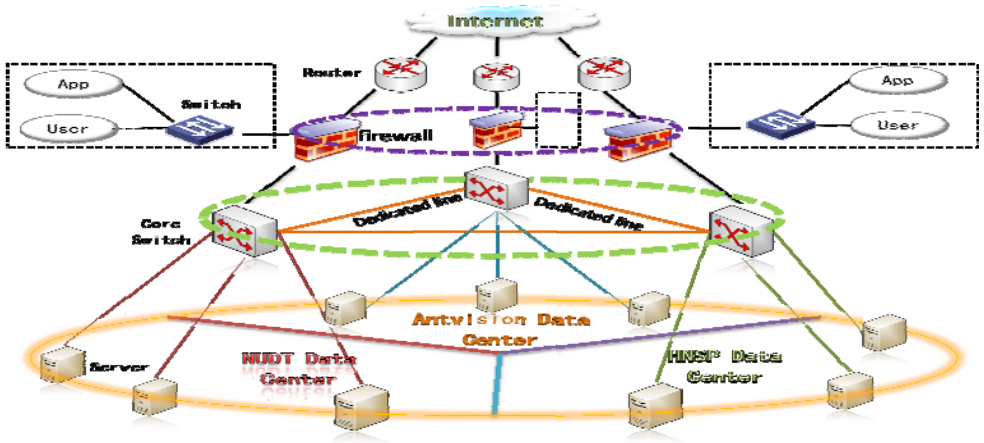


Figure 3. Deployment of the Platform

Network topologies are divided into two categories: attack network and range network (defense network). According to the network simulation script file provided by the user, the system generates a corresponding physical network topology and environment, and maintains an attack code library. Users can upload their own attack code to the library. After a testbed is established, user can deploy attack code on attack network, select the appropriate measurement tools and data collection tools provided in the ToolPak and deployed them on defense network. All after that, the experiment could start.

## 2. Assessment Analysis

The assessment analysis module on the data analysis server carries out cluster analysis on the data received from the defense network, offers a variety of correlation analysis, filters false alarms, offers a visual way to display network attack events, and provides users with a detailed view of all aspects of the experimental network security (attack, flow, assets, vulnerability, etc.). Its main function is data integration and correlation analysis, integrating network attack data from various heterogeneous network and making them work in a common environment. So we developed data collection tools, correlation engine, rulebase and data management and display tools. The architecture of the assessment analysis module in Figure 4.

The assessment analysis module consists of several sub-modules.

(1)Various types of front-end probes are deployed on defense network to monitor host, service and network, to scan vulnerabilities and to collect data.

(2)Data integration includes two types: Event data integration is used to integrate the security data get from front-end probes, such as Snort, Ntop, etc. Scan/inventory data integration is used to integrate data from Nmap, nessus, etc.

(3)Association analysis is the core of the assessment analysis. After preprocessing by security event analysis and network traffic analysis, the data is transmitted to association analysis. It analyzes the events and assets (Logical association), the event and the vulnerability of the events associated with other events (Cross-correlation), then, makes risk assessment based on association rules, and eventually generates the correct alarm and warning [10]. It converts these complex and abstract, large-scale network attacks into understandable warnings in an intelligent way for users.

(4)DB, all the data from data integration and analysis are stored in the DB.

(5)Security event display module presents the results of the evaluation of the network attack in such as chart, graphs and other.

The accuracy of the evaluation of the network attacks, to a large extent depends on the accuracy of the association rules.

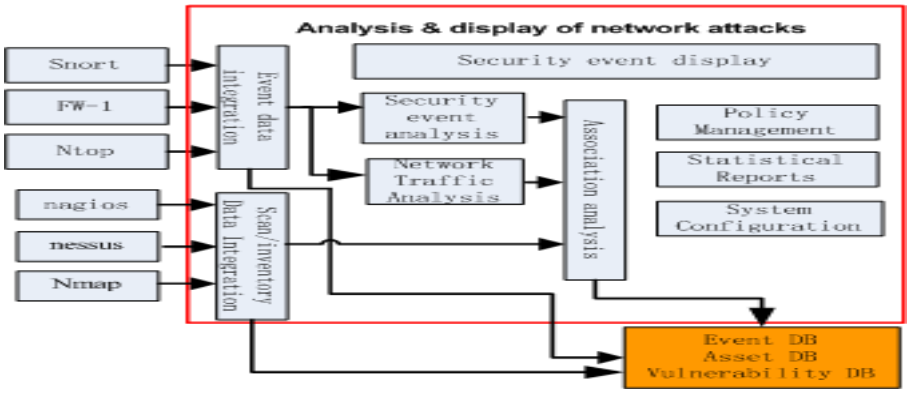


Figure 4. Architecture of the Assessment Analysis Module

### 2.2.3. User Interface

User interface provides an interactive interface for the users to access and control the testbeds. Resource application is used for users to apply resource from the system to set up a testbed and the administrator respond to the users' requests. Testbed management is used for all users including administrator and teachers to manage the testbeds of their own. Analysis and evaluation is used to present the evaluation results of the network attacks.

### 2.2.4. Management

Each layer of the system contains the management module. Resource management consists of network topology management and node management. Node management is responsible for maintaining the node type information, adding new nodes, deleting the existing nodes and so on. Testbed management is responsible for swapping testbeds in/out, multi-testbeds' concurrency control and security isolation. Experiment management controls experiments to start or stop on the testbeds. User management manages the users and projects in the system, including user registration and authorization, project application, authorization and cancellation, etc.

## 3. Results and Analysis

The system designed in this paper consists of 82 Tianhe-1A servers and 312 PowerLeader PT6510N servers (see section 2.2.1), 18 Cisco WS-C3750G-48TS-S gigabit switches including 9 Ctrl switches, 2 VM Exp switches and 7 PM Exp switches.

With the platform under normal operating conditions, we setup and swap in 2 testbeds. The network topologies created by netlab-client of the 2 testbeds are shown in Figure. 5. Figure. 6 shows the active testbeds running in the system right now. We can see that the two testbeds just created appears in the head position of Figure. 6. So this means that the system supports several testbeds running at the same time.

In the system, we create a lot of operating system images, including multi-version of 32bit and 64bit Windows Xp, Ubuntu, CentOS, Fedora and FreeBSD operating systems. We can quickly establish a user-specified network topology and physical network operating system on experiment nodes. Table 1 lists the average time of creating and swapping in a multi-node test bed. Data in Table 1 shows that the speed of establishing a large-scale network topology in the system could not be compared by live network.

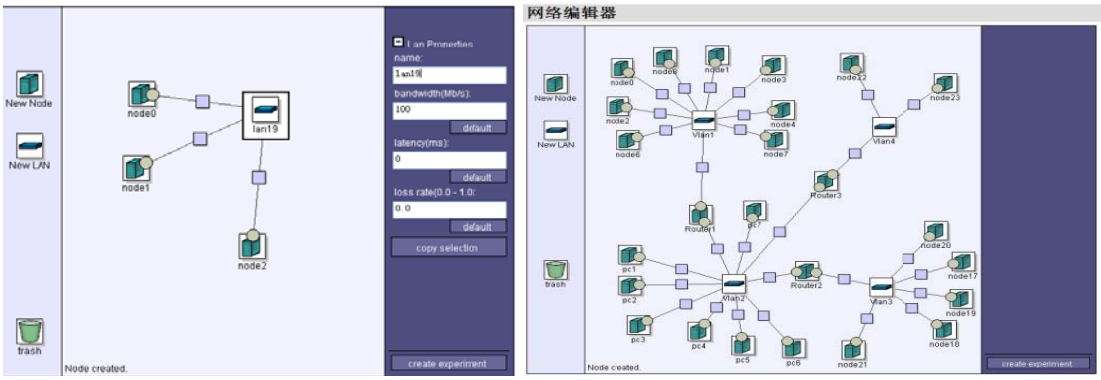


Figure 5. Network Topologies of Two Experiments

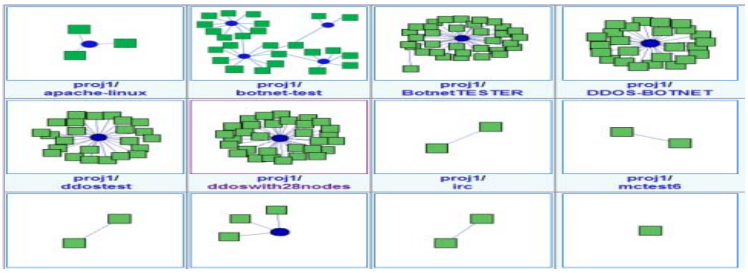


Figure 6. Part of Active Testbeds Network Topologies on the Platform

Table 1. Average Time Consuming (min) of Completing a OS

| OS \ nodes | 1 | 10 | 30 | 50 | 70 | 90 |
|---|---|---|---|---|---|---|
| Windows XP SP2 32bit | 11.4 | 12.1 | 12.4 | 12.9 | 13.2 | 13.8 |
| Ubuntu 12.04 32bit | 4.2 | 4.5 | 4.7 | 5.1 | 5.5 | 6.1 |
| CentOS 5.5 64bit | 4.8 | 5.0 | 5.4 | 5.9 | 6.2 | 6.9 |
| FreeBSD 7.3 32bit | 3.5 | 3.8 | 4.2 | 4.8 | 5.2 | 5.5 |

Figure 7 shows the index page of the assessment analysis system of the experiment environment. System provides a variety of views, including network attacks situation analysis and display, network attacks event analysis and display, network traffic analysis and display, statistical reports, security strategies, association rules. Figure 8~Figure 15 shows the data analysis results obtained from a number of network attacks during system operating normally.

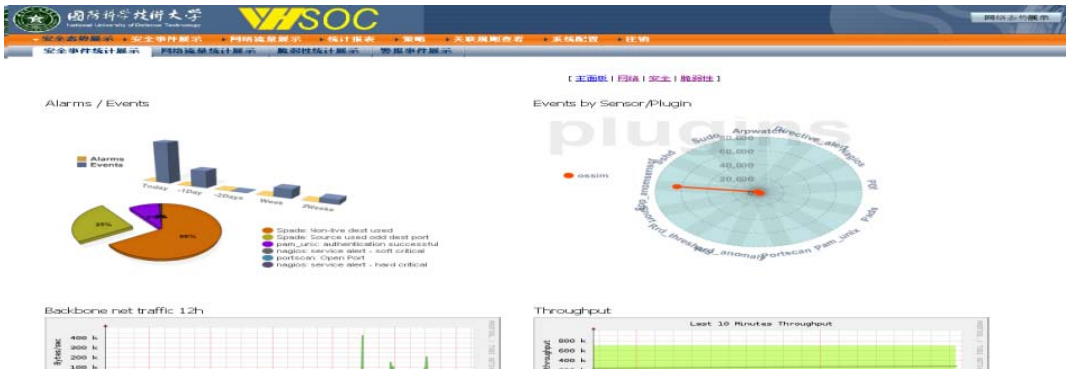


Figure 7. Index Page of the Assessment Analysis System

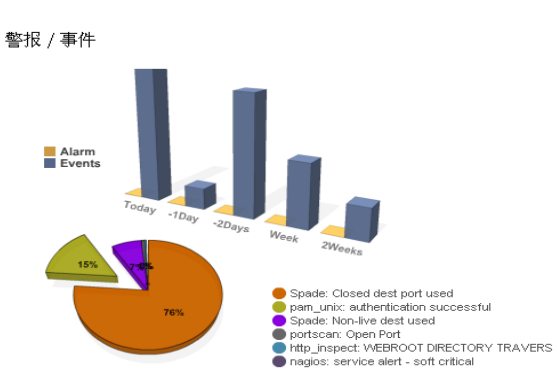


Figure 8. Alarm/event

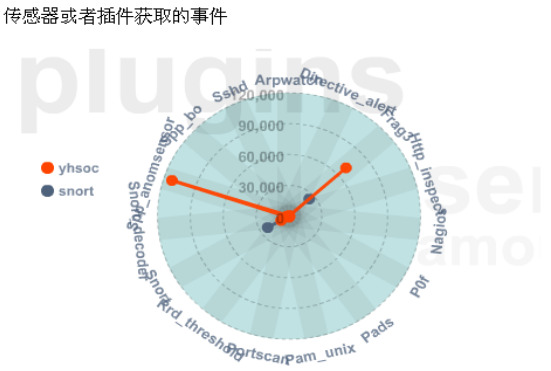


Figure 9. Sensors Event

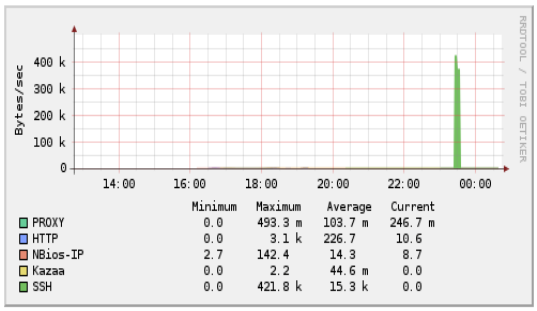


Figure 10. Network Traffic within 12 Hours

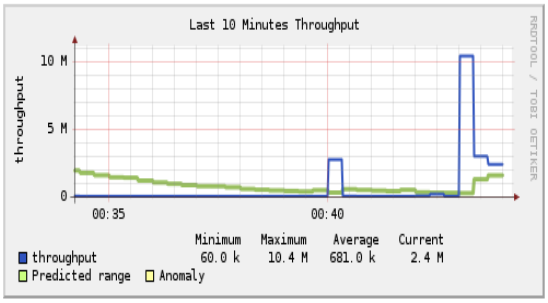


Figure 11. Network Traffic within 10 Minutes

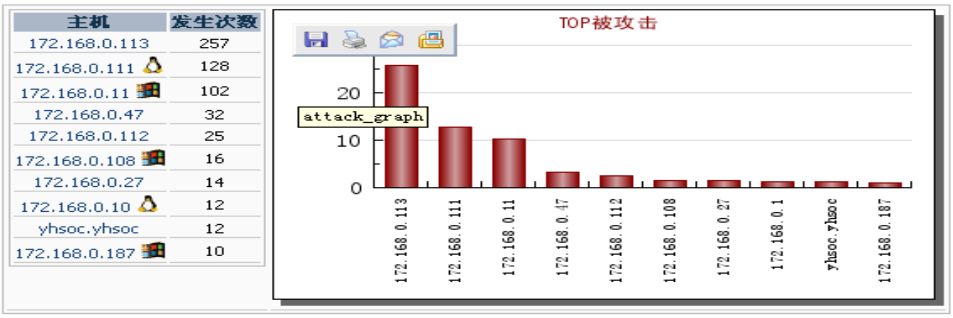


Figure 12. Top 10 of the Attacked Nodes

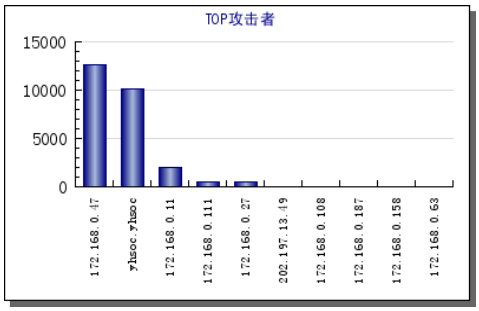


Figure 13. Top 10 of Attacking Nodes

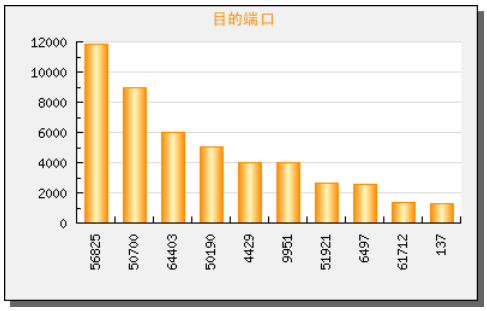


Figure 14. Top 10 Ports in using on Attacked Nodes

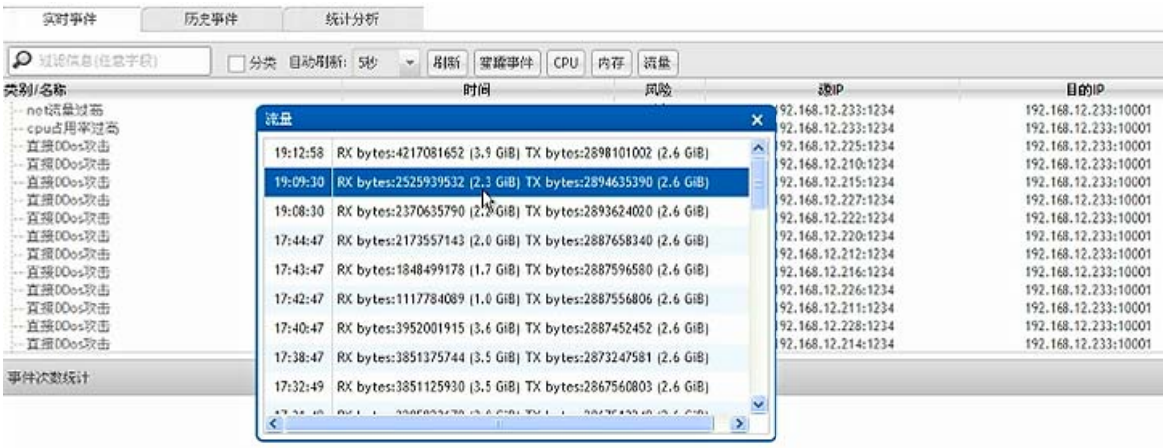


Figure 15. Real-time Attacks Event and Details from Association Analysis

Table 2 enumerates characteristics of traditional approaches. Each offers unique benefits, thus guaranteeing their continued importance. For example, simulation presents a controlled, repeatable environment. However, its high level of abstraction may be inappropriate, for example, when studying the effects of interrupt-induced receiver live lock on a heavily-loaded system.

Table 2. Characteristics of Experimental Platforms

| Metric | simulation. | emulation | live net | our platform |
|---|---|---|---|---|
| Ease of Use | √ | ModelNet ? | | √ |
| Performance | | √ | √ | √ |
| Repeatability | √ | √ | | √ |
| Packet-Level Control | √ | | | √ |
| Coarse-Grain Control | √ | √ | | √ |
| Scalability | varies | w/ ModelNet | varies | √ |
| Param. Space Explor. | √ | ModelNet ? | | √ |
| Reuse of Models | √ | ModelNet ? | | √ |
| Real Links | | | √ | √ |
| Real Routers | | | √ | √ |
| Real Hosts | | √ | √ | √ |
| Real Applications | | √ | √ | √ |
| Real Users | | | √ | |

## 4. Conclusion

This paper introduces a design and implementation of an experiment environment for large scale network attacks and defense, presents the method to measure the impacts of network attack and defense. In the future, we will improve the functions of the platform and make good use of the environment to study the efficiency of different network attacks and defense mechanisms.

## Acknowledgments

This research was supported by the Key Technologies R&D Program of China (No.2012BAH38B-04, 2012BAH38B06), National High-Tech R&D Program of China

(No.2010AA012505, 2011AA010702, 2012AA01A401, 2012AA01A402), National Natural Science Foundation of China (No. 60933005), Henan province science and technology projects ( No. 122102210124), National Information Security 242 Program of China (No. 2011A010).

## References

[1] VK Pachghare, VK Khatavkar, Dr Parag Kulkarni. Pattern Based Network Security Using Semi-Supervised Learning. *International Journal of Information & Network Security*. 2012; 1(3): 228-234.
[2] Muhammad Aamir. Review on Attack and Defense in Tor. *International Journal of Information & Network Security*. 2012; 1(2): 105-109.
[3] Hui Dong, Jian Ma. Network attack and defense experimental platform based on virtual honeynet. *Journal of Qiqihar University*. 2012; 28(2): 67-71.
[4] Yangxia Xiang, Xiaoxuan Xiang. Application of method based on hypothesized target in network attack and defense experimental teaching. *Computer Engineering and Design*. 2009; 30(4): 855-857.
[5] Haiqing Chu. The Research for the Network Emulation Experiment Testbed Mapping Problem. Master Thesis. Nanjing: Nanjing University of Posts and Telecommunications; 2010.
[6] Nurmi D, Wolski R, Grzegorczyk C, Obertelli G. *The Eucalyptus Open-Source Cloud-Computing System.* Proceedings of Cluster Computing and the Grid. Shanghai. 2009: 124-131.
[7] Sheng Ma, Jerger, NE, Zhiying Wang. *Supporting efficient collective communication in NoCs*. 2012 IEEE 18th International Symposium on High Performance Computer Architecture (HPCA). New Orleans. 2012: 1-12.
[8] XJ Yang, XK Liao, K Lu, QF Hu, JQ Song, JS Su. *The TianHe-1A Supercomputer: Its Hardware and Software*. Journal of Computer Science and Technology. 2011; 26: 344-351.
[9] Chun B, Culler D, Roscoe T, Bavier A, Peterson L, Wawrzoniak M, Bowman M. *Planetlab: an overlay testbed for broad-coverage services*. SIGCOMM Computer Communication Review. 2003; 33(3): 3-12.
[10] Jingang Li. Research and design of network security event correlation engine. Bachelor Thesis. Beijing: Beijing University of Posts and Telecommunications. 2010.