http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (21): 6449-6454, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6449.6454
© 2013 Asian Network for Scientific Information

A Mechanism of Intrusion Detection System Cooperating with Firewall

Zijian Cao and Xiaofeng Rong
Xi'an Technological University, Xi'an, 710021, China

Abstract: In order to solve the disadvantages of Intrusion Detection System (IDS) mainly based on audit tracing and the lack of active and real-time access control ability, this article puts forward a new interactive collaboration response model which combined both detection function of IDS and access control function of firewall. It researched linkage system model using of system call and rule pattern match, designed a IDS based on system call and rule pattern match, introduced the key techniques of linkage about IDS and firewall and proposed the overall implementation process of the linkage. This study has certain instructional significance to the further application and development of the intrusion detection technology.

Key words: Intrusion detection system, firewall, audit trail, access control, interactive response, system call

INTRODUCTION

Along with the rapid development of network technology, more and more systems were threatened by the invasion attack, Intrusion Detection System (IDS) has been applied widely in network security system and it becomes an important part of the information security system (Denning, 1987; Debar *et al*., 1999). When network attack been detected by IDS, the traditional response mode notifies the network administrator by displaying message, recording log, giving the alarm and so forth, then prevents intrusions by the network administrator take corresponding measures manually. It no doubt increases the degree of difficulties in safety management and it is lack of real-time performance for blocking network attack.

There are many IDSs developed with different techniques, mainly include system call (Hofmeyr *et al*., 1998; Xu *et al*., 2004; Jia *et al*., 2007; Tao *et al*., 2010), data mining (Dai and Li, 2009) and traditional scanning technique (Lee *et al*., 2001; Okazaki *et al*., 2002; Wang and Song, 2009).The method based on system call,uses sequences of system calls to describe the application of the normal behavior, has simple and direct advantages. But it just considers the sequence relationship between timing, so some attack through the system call arguments to inject code way will bypass the detection, thus it affects the measurement accuracy and efficiency. The method based on data mining, has the advantages of data extracted from the characteristics and rules. If we get connection number of records in very few cases, U2R and R2L attack detection effect is not very ideal. The method based on traditional scanning technique can detect a class of specific intrusion, but is lack of new or unknown attack detection ability.

In addition, Li *et al*. (2009) gives a new training-free model to gain more determinism and resolves indirect call/JMP through static-dynamic hybrid approach. Feng *et al*. (2003, 2004) presents formalizing program models that facilitate understanding and comparison and exposing additional program state that improves monitoring speed and model accuracy. Tian *et al*. (2008) gives a method of entropy weight coefficient that is applied to calculate the weight of factors and decrease subjective judgment on the effect of the weight coefficient for intrusion detection.

All of the above method mainly studied the function of IDS, further, Yang *et al*. (2005) proposes the use of manager-agent to achieve both linkage of IDS and firewall. Besides, Wang *et al*. (2004) designed an internet IDS cooperating with firewall, referred to as Guarder. The above two methods only use a method to detect intrusion has some limitations. On the other hand the above two methods only did not give a specific implementation process.

This study makes two principal contributions. First, it presents a linkage system model of IDS and firewall from the system-call level to the application-rule level. In the system-call level, we monitor every active process to detect attacks. On the other hand, we use rule pattern match algorithm to find intrusions in the application-rule level. Second, it gives an overall implementation process of the linkage and a set of experimental steps and result.
The rest of this study is organized as follows. Section 2 presents a linkage system model of IDS and firewall based on system call and rule pattern match, Section 3 gives the key techniques for linkage, includes system call, pattern matching algorithm, realization of IDS based on libnids, an application programming interface library and linkage of IDS and firewall. The detailed experiment test steps are

Corresponding Author: Cao Zi-Jian, Xi'an Technological University, Xi'an, 710021, China

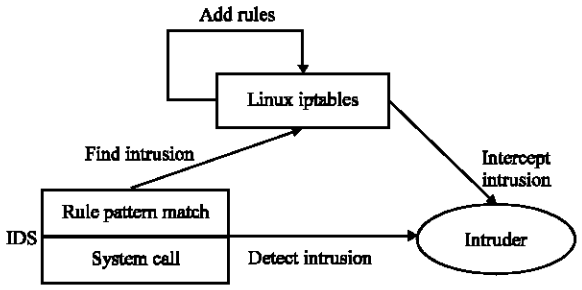

Fig. 1: Linkage system model

introduced in Section 4. Section 5 gives the analysis of experimental results. Finally, Section 6 concludes the study.

LINKAGE SYSTEM MODEL

In this study, the experimental environment is based on the Linux operating system. Iptables are deployed in the model which is integrated into Linux kernel as a firewall tool. The linkage system model is shown in Fig. 1.

In linkage system model, the IDS detects intrusion through two ways including system call and rule pattern match. System call is the underlying intrusion detection method based on Linux kernel and rule pattern match is the application level method using of string matching algorithm.

When network anomaly attack complied with the rules in application-level has been detected by IDS, the IDS first determines whether the attack packet corresponding to the source IP address has already contained in the block list. If it's not in the block list, adding a corresponding blocking rule to the firewall and then firewall will intercept it automatically. Otherwise, it's in the block list. It indicates that the corresponding rules have been added into the firewall rules already.

At the same time, when network anomaly attack complied with fixed-length sequences of other underlying privileged programs has been detected by IDS, the IDS works similar to the rules in application level.

The model detects network intrusion using the IDS and then adds filter rules to the firewall, ultimately achieves both linkage of IDS and firewall.

KEY TECHNIQUES FOR LINKAGE

The new linkage model includes lots of key techniques which contain system call trace, rule pattern match, the development of IDS based on rules and the method of adding automatically rules to the firewall.

System call trace: System call is an event which happens at the user-kernel interface. Linux system call count is 190 in kernel 2.2 and more then 300 in kernel 2.6 (Gary, 2004). Some alternative approaches are considered to trace system call of individual processes including audit packages, system-call tracing utilities (such as strace) and instrumented libraries.

In this experiment, STIDE (Julie, 1999) is used as a system call IDS tool. STIDE is an abbreviation for Sequence Time-Delay Embedding. Its function is to accept as input a time series or a set of time series, divide it into a set of fixed-length sequences, compare that set of sequences with an existing database of fixed length sequences and report on the consistency of the time series with the existing database. STIDE counts abnormal short sequence number about detected process system call series and determines whether an intrusion alarm by the local frame (LPC). STIDE also is an open source software in compliance with the GNU agreement.

Pattern match: An IDS takes the monitored user network data packet for pattern matching with the characteristic database contents. When the monitored user network data packet contains the content that matches with the characteristic database, IDS takes this behavior as an invasion. This experiment uses string matching algorithm for feature matching. The researchers commonly use string matching algorithms such as Boyer-Moore, KMP algorithm, etc. The experiment uses Boyer-Moore algorithm.

Boyer-Moore algorithm is an exact string matching algorithm published by Boyer and Moore (1977). In logic, it has certain advantages in contrast to the general matching algorithm. This algorithm mainly implements reverse character comparison on Character string to search. When the contents do not match any strings, it doesn't need to search for the whole Character string. Usage would include tasks like recursively searching files for virus patterns, searching databases for keys or data, text and word processing and any other task that requires handling large amounts of data at very high speed.

Development of IDS based on rules: The function of intrusion detection is implemented using open source library libnids. Tcp_stream which is an important data structure in libnids, is shown in Fig. 2.

The development flow of IDS based on rules is shown in Fig. 3. In addition, the linkage system also uses other open source libnet and libpcap interface to capture network data packet.

In Fig. 3, the function nids_init is used for initialization. If the system is successful initialized, it

```
struct tcp_stream{
struct tuple4 addr;
char nids_state;
struct lurker_node *listeners;
struct half_stream client;
struct half_stream server;
struct tcp_stream *next_node;
struct tcp_stream *prev_node;
int hash_index;
struct tcp_stream *next_time;
```

Fig. 2: Tcp_stream data structure in libnids

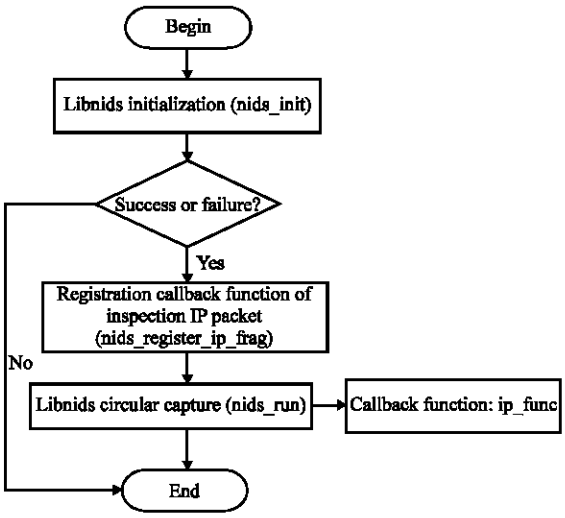

Fig. 3: Development flow of IDS based on libnids

needs to registry for the callback function for detecting and analysis the IP data packet. Ultimately, call function nids_init into the cycle captures data packet.

Realization of linkage: During the period of IDS monitoring the network communication, if IDS finds network data packet matched the pattern string, IDS will add a new rule to the rule list of the firewall with the source IP address of this network data communication data as a parameter. This rule will block the source IP address and any subsequent communication of the computer. If the suspected source IP address is 192.168.1.6, the firewall will be added a new rule. The command of adding rule in firewall like:

iptables –A INPUT –s 192.168.1.100 -j DROP

If no other rules conflict with the new rule, iptables will block network data packet from the source IP address. In C program, programmer can use the system function to invoke the shell command. Using system function, the method of adding automatically rules to the firewall is realized.

KEY TECHNIQUES FOR LINKAGE

Experiment environment: In the experimental environment, host A and host B are a group in a subnet. Host A's IP is 192.168.1.100 and host B's IP is 192.168.1.6. Host A's operating system is Linux and B's operating system is Windows 2003 Server, respectively. The linkage program is deployed in host A and Host B is the test invasion host.

System call experiment: If sequence length is 4 and normal trace is like belows: (open, read, mmap, mmap, open, getrlimit, mmap, close, read, mmap, close). According to this testers now can create a database of fixed-length sequences from scratch or to add to a pre-existing database like Table 1.

We divide measured trace into the same length of short sequence and use window to scan measured trace and one by one to compare whether is abnormal. If the measured trace is like (open, read, mmap, mmap, open, getrlimit, close), there is abnormal sequences (mmap, open, getrlimit, close) in the fourth line in Table 2. LPC value is 1 and maximum Hamming distance is also 1.

Linkage experiment:

- **Step 1:** Run IDS linkage program

In host A, we run the IDS linkage program. The command is: /ids_firewall Fedora 0. The first parameter (here "Fedora") represents pattern string (feature value) and the second parameter (in this case 0) expresses from upper IP packet loading data for pattern matching as starting address.

- Step 2: Firewall rules view before attack

The command of viewing firewall rules is: iptables-L. Viewing results are shown in Table 3 before attack.

Results in Table 3 show that firewall rules are empty in three chains of INPUT, FORWARD and OUTPUT before attack.

- **Step 3:** Host B begins to attack

Host B begins to attack host A. In this experiment, host B accesses the Web Server of host A and accessing

Table 1: Normal database of system call

ID	syscall1	syscall2	syscall3	syscall4
1	Open	Read	Mmap	Mmap
2	Read	Mmap	Mmap	Open
3	Mmap	Mmap	Open	Getrlimit
4	Mmap	Open	Getrlimit	mmap
5	Open	Getrlimit	Mmap	Close
6	Getrlimit	Mmap	Close	Read
7	Mmap	Close	Read	Mmap
8	Close	Read	Mmap	close

Table 2: Measured trace of system call

ID	syscall1	syscall2	syscall3	syscall4
1	Open	Read	Mmap	Mmap
2	Read	Mmap	Mmap	Open
3	Mmap	Mmap	Open	Getrlimit
err	mmap	open	getrlimit	close

Table 3: Firework rules view before attack

Chain INPUT (policy ACCEPT)				
Target	Prot	Opt	Source	Destination
Chain FORWARD (policy ACCEPT)				
Target	Prot	Opt	Source	Destination
Chain OUTPUT (policy ACCEPT)				
Target	Prot	Opt	Source	Destination

Table 4: Firework rules view after attack

Chain INPUT (policy ACCEPT)				
Target	Prot	Opt	Source	Destination
DROP	all	--	192.168.1.6	192.168.1.100
Chain FORWARD (policy ACCEPT)				
Target	Prot	Opt	Source	Destination
Chain OUTPUT (policy ACCEPT)				
Target	Prot	Opt	Source	Destination

```
!!!Alert:Find invasion From 192.168.1.6 to 192.168.1.100
Firewall start to block 192.168.1.6 's access.....
!!!Alert:Find invasion From 192.168.1.6 to 192.168.1.100
Firewall blocks network to 192.168.1.6
!!!Alert:Find invasion From 192.168.1.6 to 192.168.1.100
Firewall blocks network to 192.168.1.6
```

Fig. 4: Results of intrusion detection in console

path is http://192.168.1.100. We assume that it is an invasion if the users submit content like "Fedora" character in the page.

- **Step 4:** Intrusion Detection

When IDS program detects with a "Fedora" feature characters, it outputs alarm information in the console, shown in Fig. 4.

At the same time, firewall iptables will be added a rule to block the invasion. If viewing firewall rules after attack, the result is shown in Table 4.

Table 4 shows that firewall rules have a rule in INPUT Chain after attack. The rule is DROP from host B (192.168.1.6) to host A (192.168.1.100). When host B accesses the Web Server of host A again, the service will be refused.

THE ANALYSIS OF EXPERIMENT RESULTS

Experiment innovation

Experiment environment is linux: Because the linkage system uses Linux as experimental OS, software is attacked very few by hacker in Linux than in Window. Even though IDS can detect intrusion, we must first ensure that the IDS itself is not to be attacked. Based the facts above, it is security to develop the linkage platform of IDS and firewall in Linux than in Windows.

Using open source software interface: The linkage system uses open source software interface, including libnids, libnet, libpcap and STIDE. It will be no copyright issues. Libnids, libnet, libpcap and STIDE are open source software, consistent with the GNU agreement.

Linkage design of IDS and firewall: The linkage design of IDS and firewall will increase the active response ability of IDS. Using the linkage design, once IDS detects the characteristic values of the attack, IDS can add corresponding firewall rules, then firewall intercept immediately network attack and reduce the unnecessary duplication of work to network administrator.

Firewall is built-in iptables service in linux: The linkage system uses built-in iptables service in Linux as a firewall. It has high response efficiency. Iptables is an IP packet filtering system in the latest version of Linux 2.4.x kernel. It can better control IP packet filters and firewall configuration in Linux system. Thus it is high efficiency in response if we use iptables as a firewall in linkage system.

Performance analysis

Execution time overhead results: The linkage design of IDS and firewall uses Boyer-Moore fast matching algorithm as pattern matching method. Intrusion Detection time result is shown in Table 5.

Table 5 shows that Boyer-Moore algorithm has faster detection speed in case sensitive than case insensitive.

Performance of intrusion detection: There are two performance indicators in IDS: false positives and false negatives.

A false positive is not accurate to intrusion detection for attack and a false negative occurs when an intrusion generated by an intruder is classified as normal. False positive rate is calculated: Rate = false positives event number (X) / total event (N) *100%.

General, random manner, i.e. randomly is used to select part events (typically 30 to 50 events). The researchers use the real attack tool to trigger these events,

Table 5: Intrusion Detection time (us)

Length of pattern string	Boyer-Moore (Case insensitive)	Boyer-Moore (Case sensitive)
3	1.246	0.707
5	1.985	0.921
8	2.431	1.342
12	3.241	2.112

or use the playback of prior captured event data with packet capture tool, to analysis IDS alarm results, thereby get IDS false positives.

The test tools have attracting tools, scanning tools and packet capturing tools. Attracting tools have many types, for example, Blade, Fragroute, SYN flood, UDP flood, etc. Scanning tools have many types, for example, X-Scan, PortScan, Nmap, etc. Packet capturing tools have many types, for example, Sniffer, IRIS, etc.

This experiment uses scanning tool, Nmap, to test port scan detection,our linkage system can detect all scans. The results display false positive almost is zero under the condition of a small quantity attack.

In addition, this experiment uses automatically a program to send requests that include pattern string to Web server in bulk; the linkage system can detect all requests, but firewall has some jams while adding rules if there are a great number of requests.

Limitations: Although the model uses two detection methods, system call and rule pattern match, there are still limitations. It is well documented that attackers can exploit weaknesses and limitations of intrusion detection models to avoid detection. An attacker can exploit incomplete information in the model to evade the IDS. So the system shall update intrusion detection method constantly.

CONCLUSIONS

This study presents a linkage system model of IDS and firewall. The development of IDS uses two detection methods, system call and rule pattern match, it will greatly increase the performance of detection and improve detection rate. The linkage system combines intrusion detection function of IDS and access control function of firewall. It can improve real-time performance for blocking network attack.

It remains future work to investigate the security mechanisms and protocols of linkage system and detect accurately distributed, collaborative complex attack.

REFERENCES

Boyer, R.S. and J.S. Moore, 1977. A fast string searching algorithm. Commun. ACM., 20: 762-772.

Dai, H. and H. Li, 2009. Research on network intrusion detection system based on data mining. J. Intell., 28: 168-171.

Debar, H., M. Dacier and A. Wespi, 1999. Towards a taxonomy of intrusion-detection systems. Comput. Networks, 31: 805-822.

Denning, D.E., 1987. An intrusion-detection model. IEEE Trans. Software Eng., SE-13: 222-232.

Feng, H.H., J.T. Giffin, Y. Huang, S. Jha, W. Lee and B.P. Miller, 2004. Formalizing sensitivity in static analysis for intrusion detection. Proceedings of the IEEE Symposium on Security and Privacy, May 9-12, 2004, Berkeley, CA., USA., pp: 194-208.

Feng, H.H., O.M. Kolesnikov, P. Fogla, W. Lee and W. Gong, 2003. Anomaly detection using call stack information. Proceedings of the IEEE Symposium on Security and Privacy, May 11-14, 2003, Berkeley, CA., USA., pp: 62-75.

Gary, L.B., 2004. Linux system call table. http://docs.cs.up.ac.za/programming/asm/derick_tut/syscalls.html.

Hofmeyr, S.A., S. Forrest and A. Somayaji, 1998. Intrusion detection using sequences of system calls. J. Comput. Secur., 6: 151-180.

Jia, C.F., A.M. Zhong, X. Zhou, R. Tian and X.T. Duan, 2007. Research on syscall-based intrusion detection technology for linux system. Application Res. Comput., 24: 147-150.

Julie, R., 1999. DRAFT: User documentation for the STIDE software package. http://www.cs.unm.edu/~immsec/software/stide_user_doc.ps.

Lee, S.C. and D.V. Heinbuch, 2001. Training a neural-network based intrusion detector to recognize novel attacks. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum., 31: 294-299.

Li, W., Y.X. Dai, Y.F. Lian and P.H. Feng, 2009. Context sensitive host-based IDS using hybrid automaton. J. Software, 20: 138-151.

Okazaki, Y., I. Sato and S. Goto, 2002. A new intrusion detection method based on process profiling. Proceedings of the Symposium on Applications and the Internet, January 28-February 1, 2002, Nara, Japan, pp: 82-90.

Tao, F., Z.Y. Yin and J.M. Fu, 2010. Software behavior model based on system calls. Comput. Sci., 37: 151-157.

Tian, J.F., T. Liu and X.X. Chen, 2008. Survey in evaluation of intrusion detection system. Comput. Eng. Appl., 44: 113-117.

Wang, L.H., T. Li and X.P. Zhang, 2004. An internet intrusion detection system cooperating with firewall. Appl. Res. Comput., 23: 95-97.

Wang, Y.P. and G.J. Song, 2009. The study for host defend system based on port detection. Microcomput. Inform., 25: 80-82.

Xu, M., C. Chen and J. Ying, 2004. Anomaly detection based on system call classification. J. Software, 15: 391-403.

Yang, Q., J.H. Yang, X.P. Wang and B. Ma, 2005. System design based on the combination of firewall and intrusion detection technology. J. Wuhan Univ. Technol., 27: 112-115.