A Study of Intrusion Signature Based on Honeypot

Jun-feng TIAN[☆] Jian-ling WANG[☆] Ren-ling LI[★] Xiao-hui YANG[☆]

[☆]*College of Mathematic & Computer of Hebei University, Baoding, China, 071002*
E-mail: wang_jl@126.com

[★]*The Library of Hebei medical university, Shijiazhuang, China, 050091*
E-mail: li_rl@126.com

Abstract

To meet network security requirement, the model of SISH (Study of Intrusion Signature based on Honeypot) takes advantage of honeypot technology to collect the activities of attackers on the honeypot. According to information replied from the honeypot, and data captured on the network, one can reproduce the attack with attack tree. Having classified the attack information of the attacked objects, one can create the signature of the intrusion data in the same class by LCS(Longest Common Sub-string) algorithm. new signatures are used to enrich the signature database of the IDS to make up for the deficiency of the IDSs and perfect defense system.

1. Introduction

With network popularization, the affairs of network intrusion are increasingly high frequently occurring. At present, network security techniques, such as firewall and IDS, generally based on rules or matching, protect from the known attack. However, security techniques existing are short of identifying new-type attacks, which impair the nation or individual in period of time[1].

According to judging the behavior by signature base, firewall and IDS greatly depend on the system. They suffer from the limit of known knowledge, and lose capability to new-type attacks [2]. Honeypot can collect more information about attackers by interacting with them [2]. At present, honeypot technique is mainly applied to network trapping, and makes network protection turn passively tracking to actively responding [3]. The honeypot system is a kind of network resources, which main purpose is attacked or probed; and which data may be forged, like a real host providing important service and strongly lure hackers.

Honeypot has no licit user, so that each connection with the honeypot is suspicious [4]. And honeypot can imitate kinds of operate system with known leaks and services (e.g. FTP). Once it interacts with hacker, and logs the interactive date with background software [5]. Subsequently, we use tool to analyze that data, in order to discover the method and motive of aggressor. If this kind of attack isn't detected by IDS, the record and analysis are more meaningful.

Currently, the most network intrusions to same object have homology or similar traffic, and the attack can be reused [6]. This paper puts forward a model of SISH, using Honeypot to collect network intrusion packets with capture tools in multiplayer and activities in infected host. We analyze packets, and reconstruct attack process with attack tree method; we classify the information according to the attack object. We create signature with the LCS (Longest Common Sub-string) in the same class to strengthen IDS and firewall.

2. Architecture of SISH model

2.1. Construction of SISH

To collect the intrusion information, create signature and perfect network defense system, we design the model of SISH, showed in figure 1. A router is placed between Internet and deception network, in order to make honeynet more really. Moreover, the router as secondary data control device, enhance firewall control ability, to ensure that trapping host can't be used to attack other hosts.

In front of honeynet, firewall and network data capture tool intercept all connections in and from honeypots, and perform data control to prevent from

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

making honeypot a launch base to attack other hosts. Honeywall [7] mediates the network traffic going in and out of the honeypot, captures and analyzes them. Transparent bridging being activated on the honeywall, relays network frames between two NICs. It littlely modifies the content of the frames. This transparency bridge makes honeywall invisible to intruders. The third NIC, Meanwhile, connects with production network to manage honeywall and store data collected. Moreover, the other two NICs have no IP address to eliminate the risk of attacking production network, after an intruder has compromised the honeywall.

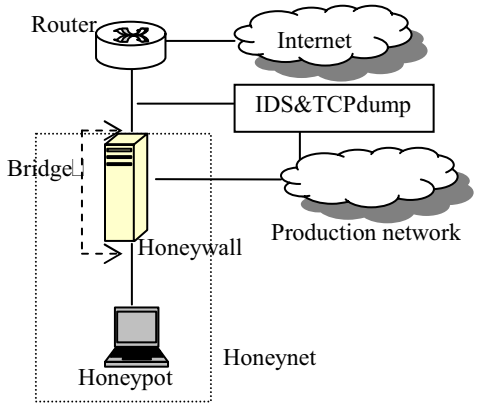


Figure 1. The architecture of SISH model

The model interacts with the intruders and intercepts attack data by honeypot. The model adopts high-interaction honeypot, to collect much more, higher value information than the low-interaction one [8]. The honeypot expose OS leaks and trap intruders. The intrusion data captured is sent to remote security host by secure tunnel.

The functions of SISH include data capture, data disposal and creating signature (illustrated in Figure 2).

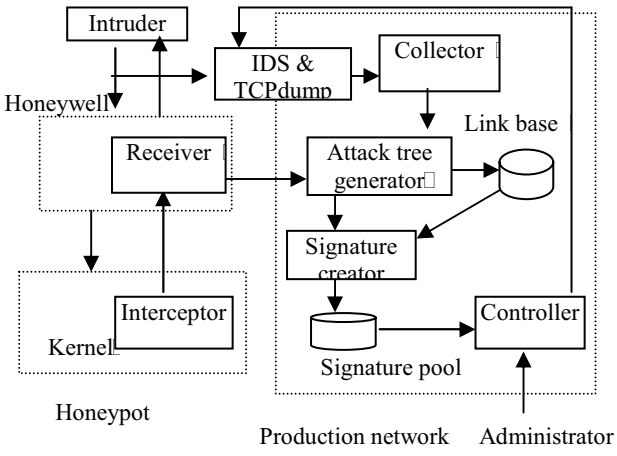


Figure 2. The function structure of SISH

Data capture intercepts network intrusion traffic in multi-layer, in order to ensure to collect full attack information. One is honeynet that intercepts the intrusion actions in object host. Others capture the traffic sent by intruders on network. The data captured must be converted to remote security host from being destroyed. The processor in production network receives them and creates the intrusion signatures.

2.2. Capturing data

The model of SISH adopts multi-layer data capture on network and in honeypot, to ensure collect full needed information. TCPdump and IDSs sniff packets in and out of honeypots on network to prevent of missing some information. We can recover toolkits from the log. Even with encrypted protocols, we can use fingerprint analysis of the packets to determine the type and possible location of attacking system.

The firewall can't only capture data, but filter packets to restrict outbound connections, in order to avoid honeypot being launch base to attack others. Firewall can redirect links from honeypot, which may attack to other hosts, to special honeypot. And it alarm as early as possible.

To know the object and status of files in file system, the model take advantage of kernel module, inserted the target system. The kernel module intercept the activities that attacker performed in honeypot, such as keystrokes. The data intercepted by kernel module is collected to remote host through security system. Interceptor modules in the kernel of honeypot intercept dynamically system calls, inner information file system processing, and send them to honeywall. The receiver module in honeywall is directly connecting with NIC diver, and intruders can't detect them. Receiver clutches the data intercepted in honeypot, and logs that to be used for data analysis, information recovering, and restructuring the intrusion [9].

Interceptor module, a Loadable Kernel Module (LKM), is loaded into kernel space. The interceptor can manipulate kernel modules, inclusive of system call table. It makes one access to kernel space of honeypot and discovers attack actions in honeypot. Once LKM loaded, we must disguise it to keep from being detected by attackers. It requests that the module can't be listed by *lsmod* instruction. Thus, we use another LKM, which deletes the nodes relating with interceptor in module connection table.

The interceptor module is an extension of Sebek module, Sebek, showed in fig.3, based on kernel rootkit, monitors the *sys_read* call. A new pointer is added in system call table, pointing to a new read function, instead of normal system call *read*(). When the standard *read*() function is called by processes, the

pointer of *read* points to its own implementation. The execution switches into the kernel context and begins to execut the new Sebek read call function. It can obtain intrusion data, even if which using encrypted facilities such as SSH. The interceptor module enhances the capability of kernel monitor. It intercepts not only one, but all the system calls involved directly or indirectly in the modification of the filesystem, for example, sys_open(), sys_creat(); sys_write(),etc.

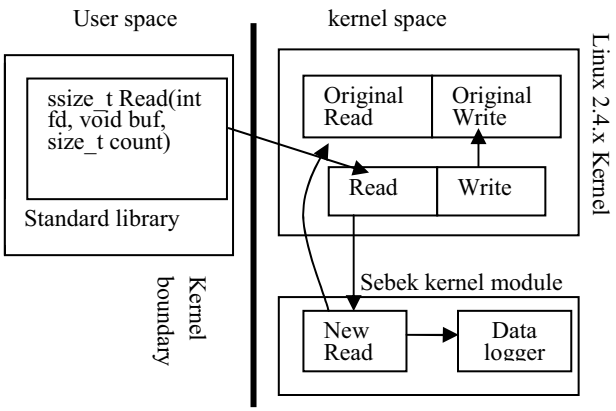


Figure 3. the module of Sebek system call

The interceptor module changes the pointers pointing to system call function, and implements the code of the interceptor module replacing the original code. When intruders call relative system calls, the interceptor module dynamically substitutes for original function. The interceptor module collects all intrusion activities in the kernel, and sends the information to the receiver module. To avoid being detected by intruders, interceptor module creates a data block, inclusive of the parameters given to the call, and system context information. Then, all intercepted calls are associated to an integer that uniquely identifies them in a certain block. This identifier is used to determine the type of system call to which the block corresponds by the receiver module, so it can interpret the block correctly.

Subsequently, they are encapsulated in Ethernet frames and sent to the network with IP and UDP protocol. The frame generated by the *gen_pkt* function, is initially allocates a socket buffer, and copied data to load field. Each frame has a unique identifier, to be identified when receiver module capture them.

Receiver module locates firewall, and passively captures the network frames passing, followed by the verification of the UDP destination port, to identify the ones sent by the interceptor module. The blocks are extracted from the identified frames and stored sequentially in a block log. Data processor performs sequence analysis, reconstruction the received blocks, and its context and system call represented are reproduced locally, and so on, based on every block identifier, to reconstruct the course of an attack.

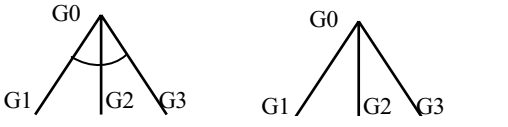
2.3 Handling data

Having performed sequence analysis, data processor constructs an evidence tree, including all attack information, i.e. the sequence of the change of file status attacked by intruders. The root of the tree is specified in the receiver module's configuration. Therefore, this root directory contains an evidence tree, which nodes are all the evidence created by the intruder on the honeypot and reconstructed by the processor.

Data processor in production network, receives packets captured by TCPdump and IDS on network. Firstly, protocol analysis is performed on network and transport layers, analyzing IP, TCP and UDP protocol header and recording their characteristic, e.g. invalid IP fragment offset or unusual TCP flag combinations. And we perform IP fragment recombination, reassemble TCP packet, since hackers may insert intrusion signature in many IP fragmentations and TCP packets, to keep away from IDS based on simple packet analysis.

Subsequently, we build an attack tree including network packets and information responded by honeypot, in term of data captured in kernel space. Attack tree is modeling tool that has expression in tree, and fits to explain the essence of attack. And it may be expressed neatly, and reused [10].

The model of attack tree expresses an intrusion with a form of "AND-OR" tree. The root is the final goal of network intrusion, which sub node is precondition of achieving it. Every way from root to leaf node denotes the integral course of an attack. The node of attack tree is classified "AND" and "OR" node. A set of attack sub-goals, all of which must be achieved for the attack to succeed, that are represented as an AND-decomposition; A set of attack sub-goals, any one of which must be achieved for the attack to succeed, that are represented as an OR-decomposition. Attack trees can be represented graphically in figure 4.



The graph of "AND" node The graph of "OR" node

Figure 4. Diagram of "AND" and "OR" node

The attack tree is formed through backward inferring. The module, firstly, ascertains an intrusion action and takes the final object as the root, namely, the root of evidence tree. Then, we remount to the precondition of the root, and make it sub node of the root. And we express their relation with "AND "or

"OR", the link between two nodes is network link that causes the changes of file system.

In succession, every node is refined with the same method, until that is leaf node. In practically, to meet different application, we may select the most interesting node to analyze more. In this course, we recover the intrusion event. Finally, we create a table titled with nodes produced. We connect the payloads of packets in "AND" node, and specify an identifier. Otherwise, this connection is appended to the connection table of that node, and compared with every connection in this list, to create the common sub-string.

Thus, the signatures that created based on data captured by IDS, and so on, can be utilized more efficaciously by IDS. It must improve the efficiency and correctness of IDS.

2.4 Creating signature

There maintains a count of the number of TCP or IP connections, in order to compare new traffic captured with them to create the signature of attack packets. Therefore, the existent connections are very necessary. We keep these connections until we ensure them no any more value.

We reassemble the flow of the "AND" node in course of protocol analysis. If the connection is new, we create empty signature record, check packets and store the fact detected. The signature record is then augmented continuously throughout the detection process. After connection table updated, the new connection compares with others that have the same node by LCS algorithm to create the longest common sub-string in their payloads.

Every sub-string of new connection X is inspected whether it is the common sub-string of the connection Y that links the same node in the list. If the common sub-string is discovered, we select the longest one. If X=<X1, X2, … Xm >, Y=<Y1, Y2, …Yn>, the longest sub-string is Z=<Z1, Z2, …Zk>, then the solution of the longest sub-string is as follows:

(1) if Xm=Yn, then Zk=Xm=Yn, and Z(k-1) is the longest common sub-string of Xm-1 and Yn-1;

(2) if Xm≠ Yn, and Zk≠ Xm, then Z is the longest common sub-string of X(m-1) and Y;

(3) if Xm≠ Yn, and Zk≠ Yn, then Z is the longest common sub-string of X and Y(n-1);

where Xm-1=<X1, X2, …,Xm-1> , Yn-1=<Y1, Y2, …, Yn-1>, Zk-1=<Z1, Z2, …, Zk-1>.

The longest common sub-string contains the prefix longest common sub-string of two strings, so that the solution of the longest common sub-string has the property of the optimization sub-structure, and fits to recursion theory. c[i,j], start denote the length and the section start the longest common sub-string in X respectively. The relation is as follows:

$$c[i,j]=\begin{cases}0 & \text{when } i=0 \text{ or } j=0\\ c[i-1,j-1]+1 & \text{when } i,j>0 \text{ and } x_i=y_j\\ \max(c[i,j-1], c[i-1,j]) & \text{when } i,j>0, \text{and } x_i \neq y_j\end{cases}$$

If c[i,j] is bigger than the number that the minimal sub-string need, then <Xstart-c[i,j]+1, …, Xstart> is a signature string of X, and append it in signature table. If there is a new signature created, then the signature pool is amended with that. The signature pool is a set, which stores new signatures detected for every node. The course is as follows:

Sig*i* is some signature in signature pool, $1 \leq i \leq n$, and n is the size of the list; sig is the new signature.

Sig = sig*i*: when sig and sig*i* match in all attributes except signature identity, the signature pool keeps old state. The new signature is dropped.

sig $\subset$ sig*i*: The new signature is a sub-string of some signature in the table, and the pool is not updated. The new one is dropped.

sig $\supset$ sig*i*: The new one is the superset of some signature in the pool, then we improve the old one.

In others cases, it is appended in the pool.

The records in the signature pool have to be periodically reported to administrator in readable format. The administrator audits the report, selects real signature to augment the signature base of IDS. At the same time, he deletes false signatures, to avid the model departing from the requirement.

3 Experimental Results

To obtain intrusion data, and create primary signature of the intrusion, we perform the experiment of Worm Blaster. We configure a honeynet with Honeyd, simulate file server running Windows 2000. Port 135 is open, and RPC DCOM is running. There is interceptor module is loaded in Linux 2.4 kernel, and the receiver is on the honeywall. The detail is described above. The simple configure of Honeyd is below.

```
create windows
create default set default personality " Windows
      NT 4.0 Server SP5-SP6"
set windows default tcp action reset
add default tcp port 135 open
add default tcp port 4444 "/bin/sh
      scripts/WormCatcher.sh $ipsrc $ipdst"
set default tcp action block
set default udp action block
```

Port 135 open to receive attack by Worm Blaster, and add port 4444, in order to make Worm Blaster gain

remote SHELL, inclusive of TFTP downloading code. Having been three-handshake, it begins to send attack data. The receiver captured data showed below.

 # /bin/sh
mkdir /tmp/10.187.84.1
cd /tmp/10.187.84.1
tftp 10.187.84.1
get msblast.exe
copy msblast.exe system32/msblast.exe

The evidence tree generated in experiment is showed in figure 5. We construct an attack tree with the corresponding packets and information responded by honeypot, as figure 6.

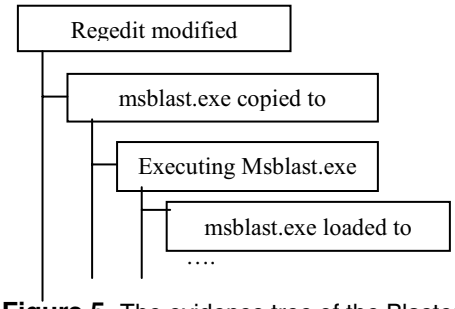


Figure 5. The evidence tree of the Blaster

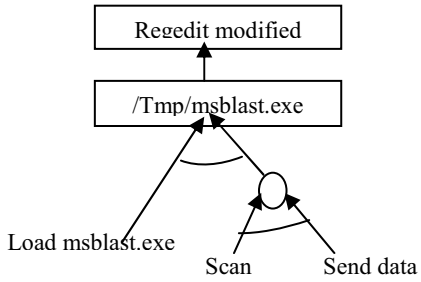


Figure 6. The attack tree of the Blaster

Subsequently, we classified the information in term of the node, reassembled the flow, created connection list, and appended data captured. We obtain the longest common sub-string by comparing to others in the same list. It is as follows:

73 74 61 72 74 20 25 73 0A 00 74 66 74 70 20 2D 69 20 25 73 20 47 45 54 20 25 73

and

74 6F 20 73 61 79 20 4C 4F 56 45 20 59 4F 55 20 53 41 4E 21 21 00 62 69 6C 6C 79 20 67 61 74 65 73 20 77 68 79 20 64 6F 20 79 6F 7520 6D 61 6B 65 20 74 68 69 73 20 70 6F 73 73 69 62 6C 65 20 3F 20 53 74 6F 70 20 6D 61 6B 69 6E 67 20 6D 6F 6E 65 79 20 61 6E 64 20 66 69 78 20

The model of SISH resolves the problem that traditional network security technologies can't identify new kind of intrusion to certain extent. It takes advantage of honeypot to capture intrusion information, classifies the data based on the object attacked, and creates signatures, and to perfect the firewall and IDS.

4 Conclusions

The model of SISH traps intruders, and capture data in multiplayer, to ensure to receive complete intrusion information. It analyzes the captured data, and creates primary signature from them. It aims to study new kind of intrusion and make network protection system active. In addition, the system can quickly react according to security policy, to protect the network.

References

[1] John McHugh, Alan Christie, and Julia Allen,"Defending Yourself: The Role of Intrusion Detection Systems" S e p t e m b e r / O c t o b e r 2 0 0 0, *IEEE SOFTWARE* pp:42-51

[2] Niels Provos," A Virtual Honeypot Framework" [R] *the 13th USENIX security symposium*, San Diego, CA,2004.

[3] Lance Spitzner 《*Honeypots: Tracking Hackers*》[M] Addison Wesley Professional 2002.9.

[4] Christian Kreibich, Jon Crowcroft "Honeycomb - Creating Intrusion Detection Signatures Using Honeypots" [J] *Computer Communication Review* (ACM SIGCOMM), 2004, 34(1) pp: 51-56.

[5] Anton Chuvakin, Ph.D. GCIA. GCIH "Honeynets: High Value Security Data"[J],*Network Security* , 2003(8) pp:11-15

[6] Xinzhou Qin, David Dagon, Guofei Gu " Worm Detection Using Local Networks" [R] Technical Report GIT-CC-04-04, College of Computing, Georgia Tech, 2004.

[7] George Chamales, " know your enemy: Honeywall cdrom" [J],*IEEE Computer Society*, 2004,2(2) pp: 77-79

[8] Lance Spitzner " trapping the hackers" [J] , *IEEE Security and Privacy* , 2003,1(2) pp: 15-23.

[9] Martim d'Orey Posser de Andrade Carbone and Paulo Lício de Geus. " A Mechanism for Automatic Digital Evidence Collection on High-Interaction Honeypots" [R]. IAW'04, *5th Annual IEEE Information Assurance Workshop*, West Point, NY, USA, 10–11, Jun 2004. 1–8.

[10] Bruce Schneier. " Attack trees:Modeling security threats" [J] , *Dr. Dobb's Journal*, 1999,12(24) pp:21-29.