The Design of SQL Injection Analysis System based on Honeynet

Zelong Yin, Zhen Niu and Feifan Tong

Abstract—**Along with the development of Browser/Server Application Development, SQL injection attacks threaten the security of Web Services more greatly, which bring enormous potential risks to data of users and service providers. Focusing on SQL injection, we designed an analysis system based on Honeynet to improve the efficiency of identification and ability of sample-taking.**

Index Terms–SQL injection, Honeynet, Security, Analysis system

I. INTRODUCTION

A. *The concept of Honeynet*

Honeynet includes one or more honeypots with multilayer data control and data capture mechanism, which is a kind of network structure. Honeynet technology is a kind of research-based, high interaction honeypot technology which helps researchers analyze attack data deeply. The core technology of honeynet includes three aspects: Data control mechanism, Data capture mechanism and Data analysis mechanism. Data control mechanism prevents honeynet from being used to attack the third party by attackers or malwares; Data capture mechanism obtains behavioral data of hacker attacks or malwares; Data analysis mechanism analyses the data captured.

B. *The concept of SQL injection*

SQL injection is one of the most popular methods used by attackers to attack databases. With the development of Browser/Server Application, more and more programmers write applications with such pattern. Programmers' experience and levels varies with the individual, so a large number of them don't validate input data of users when they write code. This results in existence of application security risk. Users could submit a database query code and obtain some data he wanted from the result that the programs returns, which is called SQL injection[1].

C. *Applying Honeynet to SQL injection*

We designed and implemented an analysis system of SQL injection which combines Honeynet technology with SQL injection principle[2]. The system attracts attackers to carry out SQL injection with Honeynet technology. Then, the system filters and analyses attacks to judge whether it is a SQL injection by SQL injection principle. Finally, the system collects the statistical results and displays them to users.

II. THE ARCHITECTURE OF THE SYSTEM

The system named analysis system of SQL injection based on Honeynet is based on honeyd. We deploy the honeynet system on sensitive nodes in accordance with honeyd and configure it to be NAT mode. We make the honeyd host establish subnet relationship with the virtual machine which has different databases (like mysql, Oracle, SQL server 2000) installed on purpose to form a high-interaction part; We create many virtual IPs making use of honeyd functional property to form a low-interaction part together with the honeyd host; Databases are used by the bait website to provide web service; The virtual machine recognizes SQL injection with natural language processing and virtual machines are implemented on the virtualization platform with virtualization technology; The virtual IP helps analyze SQL injection with some unusual network behaviors after successful SQL injection. Figure 1 shows the network structure[3].

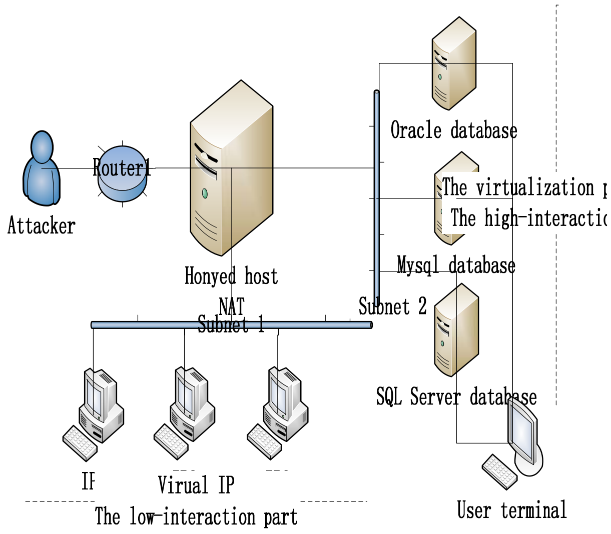

Fig 1.System architecture figure

Manuscript received September 19,2012; revised September 25, 2012.

Zelong Yin is with Harbin Institute of Technology, Weihai, Shantung 264209 CHN (e-mail:yinzelong.leon@gmail.com).

Zhen Niu is with Harbin Institue of Technology, Weihai, Shantung 264209 CHN (e-mail:zhenniuhit@gmail.com).

Feifan Tong is with Harbin Institue of Technology, Weihai, Shantung 264209 CHN (e-mail:107740160@qq.com).

The system is divided into 5 parts: Attackers part, Honeyd host part, Low-interaction part, High-interaction part, The display part.

A. Attackers part

By visiting the bait website on the Internet, attackers use SQL injection to attack the website on purpose to steal data from the website database.

B. Honeyed host part

The honeyd system, which is the core part of the system, is installed on the host. We firstly should configure the honeyd system to be NAT pattern mode so that an outer network IP is connected with the internet part and its inner network could configure two network interface cards. One of the cards is linked to the virtualization platform on purpose to form the high-interaction part[4]; the other one is connected with virtual IPs in order to form the low-interaction part. What's more, we should configure the host to realize so that users could visit different IPs of the inner network through an outer network IP. Figure 2 shows the core architecture.

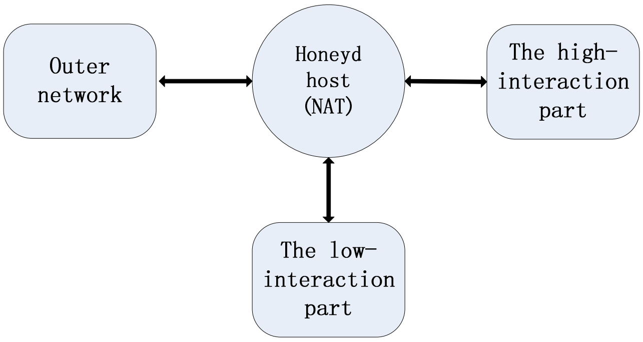

Fig 2.Core architecture figure

C. Low-interaction part

The honeyd host forms the low-interaction part together with virtual IPs created by honeyd. The virtual IPs are on the same virtual subnet and they help analyze SQL injection[5]. When the database is attacked by SQL injection, the worm from the host attacked may infect other hosts so that some unusual network behaviors will appear. The virtual IP could imitate network behaviors well, so we could collect network behaviors to aid in judging and analyzing SQL injection by virtual IPs.

D. High-interaction part

The honeyd host forms the high-interaction part together with the virtual machine which is installed on the virtualization platform and has mysql, SQL server 2000, Oracle[6]. The databases are used by web service and bait websites. At the same time, the virtual machine needs to collect analytical data into users' database.

E. The display part

The display part presents the analyzed data stored in the user's database to users.

III. THE IMPLEMENTATION PLAN

A. The implementation plan of Honeyed host

The whole framework of the host's design is that the front-end is the web proxy server and the back-end includes the low-interaction part and the high-interaction part[7]. Figure 3 below shows the framework of the host's design.

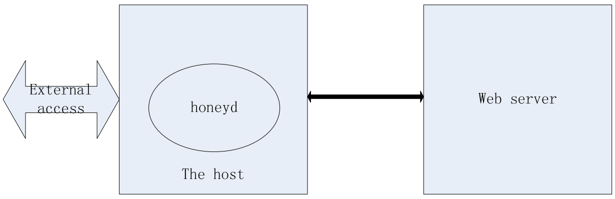

Fig 3.the framework of the host's design

We choose Nginx as the web server. Ngnix("engine x") is a high-performance HTTP and reverse proxy server. On the other hand, it is an IMAP/POP3/SMTP proxy server[8]. Multiple websites are deployed simultaneously on the server and visits of websites is monitored with Nginx HTTP proxy functions. We simulate multiple clients and servers (mail servers, ftp servers and so on) with honeyd[9]. What's more, we record attacks to help analyze and discover new means of attack in order to improve security of other parts.

B. The implementation plan of bait website

The designs of bait websites are similar to majority commerce websites and it includes five modules: product show module, user logging module, information center module, shopping cart module, website backstage management module[10]. We deploy bait websites on the web server. When we configure websites, obvious errors should be avoided in order to prevent attackers from suspecting[11]. In the meantime, the web database should contain valuable information to attract attacks. Detailed requirements are shown as follows.

Limitation of user permissions: users' permission is limited to access relevant data on purpose to prevent attackers from modifying data with registered member accounts.

Database quick recovery: the data base management system backs up and restores databases[12]. When attackers break through databases successfully, we could restore databases quickly in order to prevent data loss.

The validity of data: we copy data from real commerce websites and the validity of data couldn't be suspected.

Website recovery: once website information is updated, the file information of sites is backed up. After the website is destroyed, the backup data is imported.

The record of user behavior information: we firstly number input controls of the user logging module and record the input data with two threads and process to process communication[13]. The data recorded contains users' IP addresses, time, and input.

The bait website is a commerce website based on JAVA technology and it uses JTM (JDK + Tomcat + MySQL) IDE.

C. The implementation plan of virtual IP auxiliary analysis

When the web database is attacked successfully by SQL injection, the database host has the possibility of infecting other hosts. In the infection process, some network behaviors must be displayed firstly. ICMP and telnet packets are probably sent out. We filter the database host IP and virtual IPs with libpcap and libnet, detect packets from the database host on purpose to judge ICMP packets and telnet packets. Figure 4 shows virtual IP analysis schematic.

D. The implementation plan of the detection of SQL injection

We could divide SQL injection attacks into two kinds: identifiable attacks and unidentifiable attacks. Identifiable attacks are identifiable at present. We could deploy related policies on bait websites to shield the attacks. Unidentifiable attacks are unknown attacks which are very important in the internet. Figure 4 shows the detection structure diagram of SQL injection.

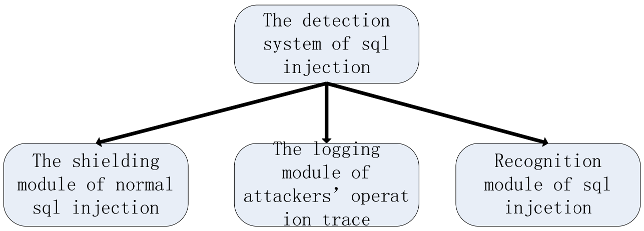
The detection system of sql injection
The shielding module of normal sql injection
The logging module of attackers' operat ion trace
Recognition module of sql injcetion

Fig 4. The detection structure diagram of SQL injection

The shielding module of common SQL injection attacks: we embed the module to web applications on purpose to shield the traditional attacks. After the application obtains the data from the browser, the database comments are filtered firstly and the database keywords which are noticed easily are filtered secondly.

The record module of attacks' operation traces: the attacks' operation traces are necessary for us to analyze new attacks. We could record them by three steps. We could deploy operations in the web application to obtain the visitors' source ips, the input form information and access time. Second, we transfer the operation information of the database to another file. What's more, we will also set up the behavior log of the operating system for the analysis of the future.

The identification module of SQL injection attacks: because characteristics of SQL injection sentences are changing, we couldn't extract their features fully. We take the exclusive method by two steps. First, we clear up database operations which are executed by the bait website in normal condition to extract main features. Sentences which satisfy main features are regarded as normal sentences and others come into the next judgment. Second, we strictly control the database's tables. At last, we record and track suspicious IPs which make a large number of trial operations[14].

IV. THE ANALYSIS OF EXPERIMENTAL RESULT

We compare the three databases' receiving SQL sentences and recognized SQL injection attacks in the same conditions. Then, we illustrate the databases' anti-attack ability by means of recognition rate calculated.

Table I. Log sheet 1

	Oracle	SQL server 2000	Mysql
Receiving SQL sentences	102	99	105
Recongnized SQL injection attacks	33	11	24
Recognition rate	32.4%	11.1%	22.9%

From the above analysis, we can see that the Oracle's recognition rate is highest and the SQL server 2000's recognition rate is lowest. We could think that the Oracle's anti-attack ability is best. We analyze the databases' infection degree by comparing the three databases' receiving ICMP packets and telnet packets.

Table II. Log sheet 2

	Oracle	SQL server 2000	Mysql
IMCP packets	22	53	35
telnet packets	23	55	40
Infection degree	small	middle	large

I can be seen from the above analysis. The Oracle's infection degree is lowest and the Mysql's infection degree is highest, which declares that the Mysql' anti-attack ability is worst.

V. RELATED WORK

When our team designs the system, we divide the system into five parts which are separately attackers part, honeyd host part, the high-interaction part, low-interaction part and the display part. We finish attracting SQL injection attacks and analyzing them through the combination of five parts. The core of the system is honeyd host and we use the modified open honeyd to deploy the host. According to the practical needs, we configure the host to be NAT mode on purpose to make the host own 3 network cards. One of the network cards is connected with the outer network and the others are linked to the inner network. The inner network is divided into two subnets. One subnet which has virtual IPs forms the low-interaction part together with the honeyd host and the other one which has web database hosts forms the high-interaction part together with the honeyd host. Because virtual IPs are created in the honeyd host, we could monitor the communication between virtual IPs and web database hosts with libpcap, libnet technology on purpose to help analyze SQL injection attacks. We could run simultaneously multiple operating systems on the host which has web databases and improve the authenticity of web databases from the value of the data, user permissions and so on. We also recognize SQL injection attacks on the database host by two steps in order to precisely recognize and record the known and unknown SQL injection attacks.

VI. CONCLUSION

We design the SQL injection analysis system depending on honeynet and honeypot technology with some new ideas. First, we use the virtual IP analysis method to help analyze SQL injection attacks on network behaviors, which improves the security and the accuracy of the analysis. Second, we could run simultaneously multiple operating systems on one computer with virtualization technology and every operating system runs different databases, which is convenient to compare SQL injection attacks impacts on different databases. The analysis of SQL injection attacks on the database host is the most important analysis. We analyze SQL injection attacks not only on security levels but also on semantic analysis and syntax analysis in order to improve the system's accuracy and efficiency of recognizing SQL injection attacks. We believe that the system is helpful for the security of web databases.

REFERENCES

[1] Thomas M. Chen, John Buford, Design Considerations for a Honeypot for SQL Injection Attacks, October 2009.

[2] GUAN Ling-qing, LOU Jia-peng, LIU Li, Extended Design and Implementation of Honeyd, Dec 2006.

[3] Xiren Xie, Computer Network, 2008.

[4] Xinyuan Zhang, Lianqing Zheng, Delude Remote Operating System(OS) Scan by Honeyd, 2009.

[5] Singh, A.N., Joshi, R.X., A honeypot system for efficient capture and analysis of network attack traffic, 2011.

[6] Visoottiviseth, V., Jaralrungroj, U., Phoomrungraungsuk, E., Kultanon, P., Distributed Honeypot log management and visualization of attacker geographical distribution., 2011.

[7] Wang, Haifeng, Chen, Qingkui, Design of cooperative deployment in distributed Honeynet system, 2010.

[8] Ronggui Hu, Jaolong Zeng, Haijun Huang, Jianjun Xia, Study and Design of Dynamic Load Balancing in Hybrid Honeynet, 2011.

[9] Chao-Hsi Yeh, Chung-Huang Yang, Design and implementation of honeypot systems based on open-source software, 2008.

[10] Wei, K., Muthuprasanna, M., Suraj Kothari, Preventing SQL injection attacks in stored procedures.

[11] kai-Xiang Zhang, Chia-Jun Lin, Shih-Jen Chen, Yanling Hwang, Hao-Lun Huang, Fu-Hau Hsu, TransSQL: A Translation and Validation-Based Solution for SQL-injection Attacks, 2011.

[12] Ju Fan, Guoliang Li, Lizhu Zhou, Interactive SQL query suggestion: Making databases user-friendly, 2011.

[13] Belknap, P., Dageville, B., Dias, K., Yagoub, K., Self-Tuning for SQL Performace in Oracle Database 11g, 2009.

[14] Rishe, N., Naboulsi, K., Wolfson, O., Ehlmann, B., An efficient Web-based semantic SQL query generator, 1999.