2011 International Conference on Modeling, Simulation and Control
IPCSIT vol.10 (2011) © (2011) IACSIT Press, Singapore

SQL Injection Protector

Wiwat Sriphum, Thawatchai Chomsiri, Ponlawat Attanak, Panuwat Noitarong

Faculty of Informatics, Mahasarakham University

swiwat@yahoo.com, thawatchai@msu.ac.th, nunthabhureng@gmail.com, hottoa21@gmail.com

Abstract. Due to an increasing number of the internet user, services have recently been adjusted to be served on the websites; for example, money transference for banking affairs, trading and purchasing on the internet and email, as well as social networks e.g. Hi5 and Face Book. At the mean time, the attacks launched toward the websites or web attack also have been found more often. In particular, SQL Injection Attack is considered the most harmful web attack and has been found and ranked on top of the frequently detected web-attack. Typically, the website developed by the web programmer with well-comprehension of SQL Injection will be prevented from this attacking. At the same time, over a half of the web programmers unfortunately has not studied well about this kind of web attack; therefore, a number of websites are still encountering the risk. Regarding the security, there were the uses of the IPS installation and firewall application with protecting functions against SQL Injection Attack as well as the Top 100 web-attack protections; however, the main problem is a very expensive cost of the protection-tool installation, in which medium and small organizations are unable to afford the installation cost.

This research aimed to offer the solution for the protection system against SQL Injection Attack in which the web programmer needed not to waste the time in writing codes on the website. Besides, the system, designed by the researchers, cost a lower expense. The researchers proposed the protection system by using a computer and Linux as a blocker between the website and the user (placed on the server zone), and creating GUI Application that could define the rules in IPTABLES (a free firewall as the software equipped with Linux) in order to possibly protect against SQL Injection Attack. Moreover, the researchers' team assessed the performance of the protection system's security and functional speed, and found that the proposed protection could authentically protect the websites from SQL Injection Attack whereas the drop of the website's functional speed did not exceed 1%.

Keywords: Web Security, Internet Security, SQL Injection, Hack, IPTABLES

1. Introduction

Nowadays, information technology is broadly spread out and the communication via the internet becomes truly necessary. For that reason, the majority of organizations presently possess its own website so that the users can make use of the website with more convenient services. Unfortunately, the communicative channels of web application are an uncontrollable communication, so the users with negative intention including the hackers take chances to launch the attacks through web application without being detected by a firewall. In fact, there are various attacking forms launched to harm web application but the frequently used one is SQL Injection Attack [1] [2]. Technically, most of the websites commonly require the users to firstly login before using the fully-activated website. This login process becomes a channel for bad users or hackers to attack web application. According to the mentioned problem, the researchers, as web developers, propose the concept of developing the protection system against web attack, in which this system will be embodied on the server functioning for additional services (not on the web server). The protection system will function in fulfilling the weakness of web application, used in an organization, without a high expense of the tool installation.

2. Backgrounds

SQL Injection Attack was an attacking process that the hacker was able to add a Malicious Code or command, used for web application hacking, into the database that the server was connected to, e.g. MS Database, SQL Server, Oracle, and MySQL, via the use of SQL command (or a part of the command). Similarly, the hacker was able to call for an external program through a shell command of the operational system. Regularly, the hacker would launch the attack during the authentication or login process via web application; for example, many websites normally employed the administrator's account as a gateway to the webpage for the web administration. During this process, it was possible for the hacker to attack the web system as well as to get into the website with the administrator's authority.

The creation of SQL Injection Attack was that the hacker keyed in any username but the password must be keyed in as "Logic Statement" e.g. ' or '1' = '1 or ' or "='. If web application was not equipped with an "Input Investigation" for detecting and filtering for such foreign strings, the hacker would be able to get through the web authentication and login to the website with any unknown username and password. The protection against SQL Injection Attack [1] [2] could be done by the following procedure:

1. Investigate the keyed characters via an input form e.g. blocking the following characters:

   ' ; - " | ^ % & @ ( )

2. Limit the length of the input's field.

3. Remove the protection script on the client to the server.

The completion of SQL Injection Attack depended on the performance of web application written by the programmer. Mostly, the possibility that ASP would be successfully attacked was higher than PHP. However, PHP was still in risk whereas the packet filtering firewall was unable to perfectly prevent this form of web attack.

3. Structure of the Designed Security System

The researchers' team designed the protection system by using a computer as the web protector. This computer functioned under an operation of Linux and used IPTABLES as a firewall. Presently, new versions of IPTABLES could define the rule for data investigation at an application layer level. Examples of the rule [3] were as the followings: iptables [table] <command> --string <match> <target/jump>

Hence, in order to detect and protect the website from SQL Injection Attack, the researchers could define the rule as presented below:

iptables -I FORWARD -s 0.0.0.0/0 -m string --string "%27+or+%271%27%3d%271" --algo bm -j DROP

The "%27or%271%27%3D%271" was an encoding. When decoding the URL, it would result as a message ' or '1'='1. The abovementioned rule was able to block SQL Injection Attack by detecting the strings (packet) that matched the security requirement. When the packet that matched the defined string was found, that packet would be immediately dropout.

Positioning of web protector was designed to be placed on the web server zone as a blocker between the web server and the firewall as shown in Figure 1, in which this positioning was a blocking between the web server and the users (both internet users and intranet users).

There were various forms of SQL Injection Attack [4]; for example, 'or 1=1; -- or 'or ''='. These included new forms of the attack that were recently discovered. When additional rules were individually added, the administrator must remote it into the web protector and to edit the file that recorded IPTABLES rules via an editor program. After that, the administrator must save the change and re-run IPTBLES. This procedure was quite complicated, so the researchers created GUI application which was the medium between the administrator and the web protector. The interface that the administrator could view was GUI: Graphic User Interface for an easier implementation, and when the rule (a form of SQL Injection Attack) was added, the administrator could command it for saving and functioning by pressing the key. GUI application would generate the rule and append it to IPTABLES rule on the web protector. At last, GUI would command an activation of the rule as illustrated in Figure 2.

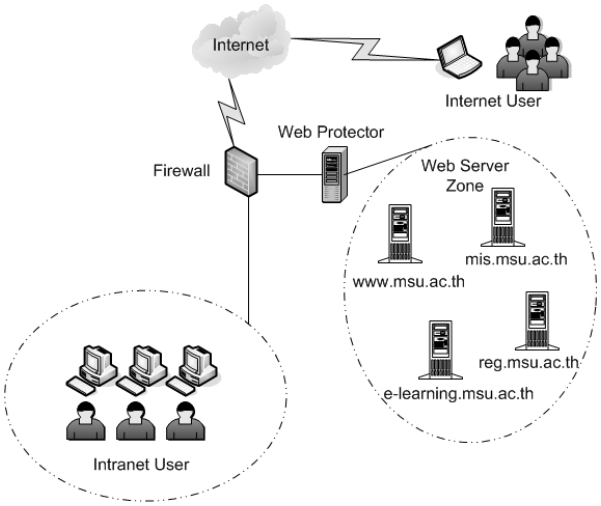

Fig. 1: Positioning of Web Protector.

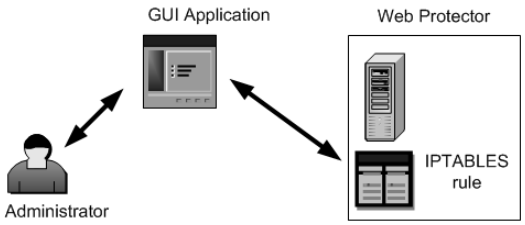

Fig. 2: the administrator could easily customize the rule by using GUI Application.

The benefit of the web protector was that beside the protector's performance in detecting and protecting against SLQ Injection Attack, the researchers additionally could extend the protector's features to protect against new forms of web attack e.g. XSS: Cross Site Scripting and RFI: Remote File Inclusion, as well as other web attacks listed on the Top 10 Web Attack (see more on the OWASP website [5]). An example of XSS: Cross Site Scripting [6] was shown as followings:

```
<script>
idata = new Image;
idata.src="http://www.evil.com/path/cookie.php?cook="+escape(document.cookie);
</script>
```

An example of RFI: Remote File Inclusion [7] was as below:

```
http://www.victim.com/main.php?page=news.php
```

An additional benefit of the web protector was that the user could implement IPTABLES and parameter – L for saving the attack's log which included the hacker's entry IP address, the destination website under the attack, moment of attacking, and forms of the attack. These were the useful information that assisted the administrator to monitor the hacker that was attacking the website.

4. Assessment of the System's Performance

To assess the performance of the web protector, the researchers would consider the security and performance as followings:

4.1. Security

The researchers studied forms of SQL Injection Attack and known that a website: http://www.governmentsecurity.org [8] had collected various forms of SQL Injection and it totally was 23

forms. Thus, the researchers decided to try out the web protector with every forms of the attack by designing the rule with smallest number of the line (for a quick processing). For example, the researchers could combine 3 forms of the attack consisting of:

' or 1=1

' or 1=1--

' or 1=1;--

to be a single form, which namely was ' or 1=1

In other cases, it could be done via the same procedure above. Thus, after combining the rules, number of the rules to be put into IPTABLES conclusively became 9 as presented below.

iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27+or+%271%27%3d%271” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27+or+1%3d1” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27+or+%271%27%3d%271” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+%27%27%3d%27” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+1%3d1” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+%271%27%3d%271” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+(%27%27%3d%27” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+(1%3d1” --algo bm -j DROP
iptables -I FORWARD -s 0.0.0.0/0 -m string --string “%27)+or+(%271%27%3d%271” --algo bm -j DROP

These mentioned rules were able to protect the websites from all of 23 SQL Injection Attacks listed on the website: http://www.governmentsecurity.org. In conclusion, when excluding new forms of web attack, it could be resolved that the protection system, designed by the researchers’ team, was able to effectively protect the website from SQL Injection Attack.

4.2. Performance

The researchers’ team undertook an assessment of the system’s performance by assessing the response time value of the webpage request (if the web protector was placed as a blocker, the user would take longer time before getting an access to the full webpage). In details, the researchers carried out the test and made a comparison between the response time before and after implementing the web protector. Based on the testing result, the researchers found that the increase of response time value did not exceed 1%, as presented on Table 1.

Table 1. The response time value of the webpage request

The tested websites	Response time	
	Before the implementation (seconds)	After the implementation (seconds)
www.google.com	23.62	24.38
www.facebook.com	16.18	16.59
www.youtube.com	7.53	7.81
www.yahoo.com	12.48	13.27
www.blogspot.com	10.92	11.45
www.baidu.com	15.74	17.12
www.live.com	9.82	10.32
www.wikipedia.org	13.19	14.08
www.twitter.com	18.37	19.45
www.qq.com	6.12	7.54

In addition, the researchers tested the protection system on the regular PC that functioned as the web protector with the different amounts of the users, in order to examine the CPU’s task load, and the result was as shown on Table 2. This result was the testing result undertaken with various amount of the users

(undergraduate students), in order to assess the CPU's task load by allowing the users to make a webpage request from the internet with a normal frequency of use. At this point, it was not necessary whether it was SQL Injection Attack or a regular webpage request, because the data of all HTTPS request needed to get through the web protector in all cases.

Table 2. The CPU's task load

Number of users	CPU load (%)
1	0.12
2	0.15
4	0.39
8	0.62
16	1.21
32	2.67
64	4.13

Note: Specification of the PC as the web protector required the following components:

CPU: 1.8 GHz, RAM: 2G, HDD: 500 GB, and OS: Cent OS 5.5.

5. Conclusion

In this research, the researchers suggested the protection system against SQL Injection Attack via the application of the web protector for the detection and protection. The web protector's positioning will be on the server zone as a blocking between the website and the user. All the packets that pass through the web server will be filtered by IPTABLES. In particular, the web protector can be implemented to detect and protect against SQL Injection Attack by defining the rule for data investigation at the application layer level; the rule will contain the signatures of SQL Injection, such as ' or ''='. Furthermore, the researchers simplify the web protector's functions by creating GUI application to be the medium between the web administrator and the web protector, as well as to allow the administrator to simply customize the rules via GUI. Functionally, GUI application can generate the input rule as IPTABLES rule in a text form. Moreover, the researchers assessed the performance of the web protector's security and found that the web protector can practically protect the website from SQL Injection Attack. In the same vein, after testing the web protector with a huge number of the users, it was firmly indicated that the protection system, proposed in this research, can be implemented on regular PC (as the web protector) as well as effectively used in the medium and small organizations.

6. References

[1] HACKING EXPOSED 6 Edition, Stuart McClure, Joel Scambray, and George Kurtz ISBN: 978-0-07-161374-3

[2] "SQL Injection - OWASP", <https://www.owasp.org/index.php/SQL_Injection>

[3] "netfilter/iptables project homepage - The netfilter.org project", <http://www.netfilter.org>

[4] C. Anley. Advanced SQL Injection in SQL Server Applications. An NGS Software Insight Security Research (NISR) publication, 2002. URL: http://www.nextgenss.com/papers/advanced sql injection.pdf.

[5] "Top 10 2007".OWASP Top 10 2007! <http://www.owasp.org/index.php/Top_10_2007 >

[6] G. A. D. Lucca, A. R. Fasolino, M. Mastroianni, and P. Tramontana, "Identifying Cross Site Scripting Vulnerabilities in Web Applications",in Proceedings Sixth IEEE International Workshop on Web Site Evolution (WSE 2004), Chicago, IL, U.S.A., 2004, pp. 71-80.

[7] Hugo F. Gonz?lez Robledo, "Types of Hosts on a Remote File Inclusion (RFI) Botnet," cerma, pp.105-109, 2008 Electronics, Robotics and Automotive Mechanics Conference, 2008.

[8] "SQL injection Basic Tutorial" Network Security Resources" <http://www.governmentsecurity.org/articles/sql-injection-basic-tutorial.html>