Security Against Network Attacks on Web Application System

Yashu Liu[1,2(✉)], Zhihai Wang[1], and Shu Tian[2]

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
ly_s8020@163.com
[2] School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

Abstract. With the development of Internet, web applications are more and more. Network attacks have become increasingly serious problem. How to make network security administrators quickly discover vulnerabilities and protect networks against attacks has become an important part of network security protection. In this paper, it introduces the principle of web vulnerabilities and implements a forum system built in some web vulnerabilities. Then it simulates the process of web attacks according to various types of vulnerabilities and gives the defensive means separately. This system can be used to conduct security training, test security tools, and practice common penetration testing techniques for network administrators and web developers.

Keywords: Network attack · Defensive measure · SQL injection · XSS

1 Introduction

It was reported that "Coinhive" mining software intruded into Youtube video platform in 2018 [1]. The malicious software tried to hijack the users' CPU and excavated the encrypted currency, which is called "cryptojacking". It is a famous network attack. The 2018 global risk report released by the Davos world economic forum, showed that global leaders were worried that the threat of large-scale network attacks ware more than terrorism in January 23, 2018 [2]. The large-scale network attacks have been the third of the most likely major risks.

It is very important to be prepared for future attacks. In order to improve the network security administrators' defense skills, we implement a forum built in some common web vulnerabilities. On the forum, network attacks are simulated, and given defense means. Thus, it can provide a real attack and defense environment for users. It is very useful to users teach or learn web application security.

The rest of this paper is organized as follows. In Sect. 2, we discuss the related work in network attack. In Sect. 3 we describe common web vulnerabilities principle. Implementation of forum system is detailed in Sect. 4. Simulation of attack and defense process is shown in Sect. 5. We give the conclusion in Sect. 6.

X. Yun et al. (Eds.): CNCERT 2018, CCIS 970, pp. 145–152, 2019.
https://doi.org/10.1007/978-981-13-6621-5_12

2 Related Work

In foreign countries, the mainstream of vulnerability simulation environment is released by open source agencies. For example, Webgoat, dvwa, Metasploitable and so on. The most important one is Webgoat [3], which is a teaching environment released by OWASP based on OWASP TOP 10. Webgoat has common web vulnerabilities, such as SQL injection and cross site scripting. Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable [4]. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment. Metasploitable is an Linux virtual machine [5]. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

The main contributions of this paper are as follows:

- We develop a forum which has common functions, for example, registering, logging in, starting new topics, replying topics and so on.
- The forum has some vulnerabilities which are appear or disappear by hand. Users can understand the harm of vulnerabilities.
- It can simulate the process of network attack. Users can launch the attack according to the types of vulnerabilities by themselves. During the process, they can understand network attack and web vulnerabilities more deeply.
- It can provide defensive measures according to network attack. Users can learn how to defense various network attack and avoid web vulnerabilities.

3 Common Web Vulnerabilities Principle

In our forum application, it introduces some common web vulnerabilities, for example, brute force vulnerability, SQL injection vulnerability, XSS vulnerability, file upload vulnerability [6–8].

Brue force uses exhaustive methods to decipher passwords, verification codes, etc. It can calculate the password one by one until it finds the real one. Generally, it doesn't know the scope and specification of the password.

SQL injection means that it can insert some SQL codes into query strings of domain name or page request [9]. Thus, the server is deceived to execute malicious SQL codes. In another words, SQL injection makes the database server doesn't execute the correct SQL codes, but to execute the malicious codes if it has some vulnerabilities.

XSS is known as a cross station script attack, which is a type of vulnerability on web application [10–12]. It makes attackers inject JavaScript codes into some web pages which have some vulnerabilities. Then the users open the URL whose pages have malicious codes on the web browser, and the malicious script is executed.

Most websites and applications have file upload function. File upload function doesn't restrict the uploaded file suffix and file type on some websites strictly, which

can be attacked by uploading various malicious files into Server, for example, PHP files. When PHP files are interpreted, Trojan horse, virus, malicious script, or WebShell will be executed on the server.

File upload vulnerability is a huge harmful one, WebShell has expanded the impact of harm. Most file upload vulnerabilities can be attacked, attackers will leave WebShell in order to access to the system later.

4 Implementation of Forum System

In order to provide a secure and legitimate website for web attack tests, we implement a forum system. Except general forum functions, it is also built in various vulnerabilities.

4.1 The Functions of Forum System

The forum has some common functions and been built in some web vulnerabilities. The forum system is designed by thinkPHP and MVC. It has three main function, "community management", "users management" and "post management". Each function is subdivided into sub functions. The function module diagram of the forum system is shown in Fig. 1.

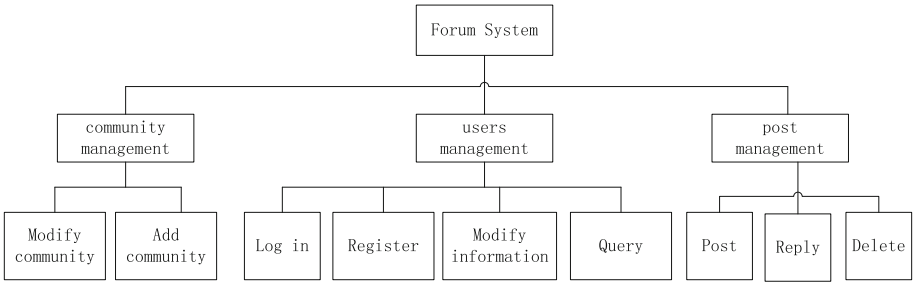

Fig. 1. The function module diagram of the forum system.

In addition to, we have created a database with four tables. They are "bbs_category", "bbs_user", "bbs_closeip" and "bbs_details". The information of topic category, users, IP and topics is stored in them separately. For example, the details of "bbs_details" are shown in Table 1.

4.2 Reserved Web Vulnerabilities

In the forum, it is reserved some web vulnerabilities, in order to simulate the attack. XSS vulnerability is built in registration module. In the module, it should check the users' data posted by form, but it ignores the process. The key codes are shown in Table 2.

Table 1. The table of bbs_details.

Field	Datatype
Id	Int(11)
Userid	Int(11)
Parentid	Int(11)
Title	Varchar(255)
Content	Text
Addtime	Datetime
Addip	Int(10)
Hitcount	Int(11)
Replycount	Int(11)
Isdelete	Int(11)

Table 2. The process of building in XSS vulnerability.

```
if(!isset($_SERVER['HTTP_REFERER'])){
     $this->notice('<font      color="red">Please      login      in
</font>','index.php?m=index&a=reg');
        }
        //get the user's data
        $uname=trim($_POST['username']); //get user's name
        $pass=trim($_POST['password']); //get the password
        $repass = trim($_POST['repassword']);//get the password again
        if($pass!=$repass){
         $this->notice('<font color="red">Please input again, because
the   values   of   the   pass   and   repass      are   not   equal
</font>','index.php?m=index&a=reg');
    }//here doesn't check and filter user's input, built in XSS
   vulnerability
       ……
        $data = [
             'username' =>$uname,
             'password' =>md5(md5($pass.$this->salt).$this->salt),
             'regtime' =>date('Y-m-d H:i:s',time()),
             'regip' =>$ip,
             'lasttime' =>date('Y-m-d H:i:s',time()),
             'money' =>200,
        ];
        if($userObj->add($data)){ //write the data into database
            $this->notice('<font color="gray">Successfully!
</font>','index.php?m=index&a=index');
        }
    $this->notice('<font color="red">Failure!
</font>','index.php?m=index&a=reg');
```

In login module, the forum doesn't check and filter user's data. It is built in SQL injection vulnerability. At the same time, the system doesn't limit the times of login requests by users, leaving a potential threat of brute force.

For file upload vulnerability, the forum is built in the module of modifying user's avatar. In the module, user can upload an avatar file, but it doesn't be limited the size and type. Thus, user upload the files which may be malicious ones. The codes are shown in Table 3.

Table 3. The process of building in file upload vulnerability.

```
.....
$file_true_name=$_FILES['myfile']['name'];
$move_to_file=$user_path."/touxiang.jpeg";
  if(move_uploaded_file($uploaded_file,iconv("utf-8","gb2312",
$move_to_file))) {
       echo $_FILES['myfile']['name']."Upload successfully!";
       $this->notice('<font  color="green">  Upload  successfully!
</font>','/index.php');
    } else {
          $this->notice('<font   color="red">   Upload   Failed!
</font>','/index.php');
        }
.....
```

In addition to, some vulnerabilities are built in the module of forum community modification, user permission modification and posting new topic, replying topic, deleting topic and so on.

5 Attack and Defense

On the forum, we can simulate web attack and defense which help web developers better understand the processes and means of securing web applications.

5.1 Simulate Attacks

In login module, the forum system doesn't check the data of form and filter the key SQL codes. It is built in vulnerabilities. To know if there is a SQL vulnerability, we write three SQL codes, as follows:

(1) **select * from bbs_user where username='username' and password='password'**
(2) **select * from bbs_user where username='username' and password='password' and '1'='1'**
(3) **select * from bbs_user where username='username' and password='password' and '1'='2'**

If (1) and (3) are executed abnormally, and (2) is executed normally, the system has SQL vulnerability. Thus, we can try to log in as an administrator using the following SQL code:

select * from bbs_user where username='admin' and password='XXX' or '1'

The SQL string implements to log in the forum as an administrator without password verification. Then, the information of the forum will be revealed.

In forum system, it doesn't filter "JavaScript" codes in controller layer and put the data from web form into database directly. Thus, we simulate XSS attack. On register page, we input the following string as username:

"<script>alert(\'h Attack，Just a kidding!\');</script>"

Once it is registered successfully, the malicious JavaScript code will be executed. Of course, it will cause more harm. When the followed codes are written into database, the users' information will be transferred to hacker website who open the pages built in malicious codes. The codes are as follows:

"<script>window.location.href=\'http://www.hacked.com/index.php/?c=\' + document.cookie + \'&url=\' + window.location.href </script></p>"

To simulate file upload attack, we use hacker software to test the vulnerability. In the module of avatar modification, we upload malicious php file from the website. Then the malicious file can modify the data or execute shell script to make more serious attack.

5.2 Defense Means

To secure web applications, there will be relative defense means against network attack. In this paper, it will give some means.

In order to prevent brute force attacks, we can take restrictions on the times of requests from users. It can also increase the difficulty and cost of brute force by encrypting passwords again. The example of encrypting is shown in Table 4.

Table 4. The example of encrypting.

```
// when registering
……
$data = [
        'username' =>$uname,
        'password'
=>md5(md5($pass.$this->salt).$this->salt),//encrypting
……
// when logging in
if($user[0]['password']  !=md5(md5($password.$this->salt).$this->salt))
```

It is strictly forbidden to connect directly to the database in the controller layer. The database can be accessed through the model layer, which greatly reduces the risk of SQL injection attack. Filtering user input data on form is also a common means to prevent it.

Checking user input data is the most effective and practical means to defense XSS attack. The example is shown in Table 5. In addition to, the field is limited by length and type in database which can defense many malicious attacks.

Table 5. The example of XSS attack

```
……
$obj = new Details();
        $deltails = $obj->getDetailsById($id);

$deltails[0]["content"]=strtr($deltails[0]["content"],"script","js"
);
        $this->assign('details',$deltails[0]);
```

Defensing file upload attack can be implemented by prohibiting all suspicious files uploaded to the server strictly. In forum system, it adds some codes to check the size and type of files. The details are shown in Table 6.

Table 6. Filtering JavaScript code.

```
……
if($file_size>2*1024*1024) { echo "The size of file is not over 2M. ";
        exit();   }
   $file_type=$_FILES['myfile']['type'];
    echo $file_type;
    if($file_type!="image/jpeg" && $file_type!='image/pjpeg') {
        echo "only jpg!";
        exit();
    }
```

6 Conclusion

In this paper, it implements a forum built in some web vulnerabilities which can aid users to learn web application security. It can be provided to network administrators to learn how to defense web attacks. At the same time, it can give some advice to web application developers who can avoid web vulnerabilities. It presents defensive measure of brute force vulnerability, SQL injection vulnerability, XSS vulnerability and file upload vulnerability, but that's not enough. We will pay more attention to other network vulnerabilities later.

Acknowledgments. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803604, 2015BAK21B01, in part by the National Science Foundation of China under Grant 61672086, 61672086.

References

1. Cryptojacking news. http://www.sohu.com/a/219580006_609556. Accessed 29 Jan 2018
2. 2018 global risk report. http://www.sohu.com/a/218458499_781333. Accessed 23 Jan 2018
3. Webgoat. https://sourceforge.net/projects/metasploitable. Accessed 22 Jan 2018
4. Damn Vulnerable Web App (DVWA). http://www.dvwa.co.uk. Accessed 25 Jan 2018
5. Metasploitable. https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project. Accessed 1 Aug 2018
6. Kang, C., Zhu, Z.: Network attack and defense experimental platform based on cloud computing technology. J. Xi'an Univ. Posts Telecommun. **3**, 87–88 (2013)
7. Zhang, L., Zhou, H.: Design of network security attack and defense experimental platform based on cloud computing technology. In: Software Guide, vol. 9, pp. 188–191 (2015)
8. Huang, X.: Development and implementation of network attack and defense experimental platform. Exp. Technol. Manag. **5**, 73–76 (2017)
9. Liu, Y.: Brief discussion of the deployment and design of field exercises on Internet security. Cyberspace Secur. **8**, 88–90 (2017)
10. Huang, J., Zhang, H., Pei, J.: Design of virtual simulation experimental teaching system for network security. Res. Explor. Lab. **10**, 170–174 (2016)
11. Ma, L.: Design and implementation of network security attack and defense training platform. Wirel. Internet Technol. **22**, 75–77 (2017)
12. Ye, J., Zhang, P., Gao, Y.: Design and construction of a network attack and defense combat training platform based on Openstack. Exp. Technol. Manag. **3**, 86–89 (2016)