Network Security Lab
Experiment#8
SSL/TLS Heartbleed Attack

Dr. Ahmed Awad & Eng. Ibrahim Amreyeh

April 11, 2021

1 Objectives

The purpose of this experiment is to understand the serious vulnerability of Heartbleed attack for SSL/TLS sessions.

2 Introduction

Heartbleed is a severe flaw resultant from the implementation of Heartbeat protocol of Secure Socket Layer (SSL) or Transport Layer Security (TLS) to keep the connection alive. In this attack, an attacker might be able to steal critical data from the memory of the victim machine such as: TLS session keys, passwords, etc.
Any HTTPS website that uses SSL/TLS can be attacked. However, to understand the Heartbleed attack, there should be a clear understanding for the heartbeat protocol that is implemented in SSL/TLS to keep the connection alive.
The heartbeat protocol consists of two types of packets: HeartbeatRequest packet and HeartbeatResponse packet. When a client sends HeartbeatRequest packet to the server, the server sends HeartbeatResponse packet in which a copy of the received HeartbeatRequest is included. Fig.1 illustrates the idea of heartbeat protocol. When the client sends a request to the server, the server will copy the data received from the client in its response. For example, consider the HeartbeatRequest shown in 1 which contains 3 bytes of data (ABC) with a length field of 3 in the packet. The server places this data in the memory, and then copies the first 3 bytes from the beginning of the data to its response packet. However, in the attacking case, the length field of the HeartbeatRequest packet is modified. For instance, assume that the data in the HeartbeatRequest is (ABC), with a length field of 1003 bytes. then when the server prepares the HeartbeatResponse packet, it will copy the first 1003 bytes of its memory, which contains the value "ABC" and other private data related to that server, such as

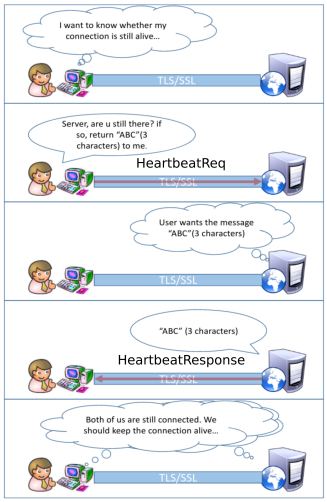

Figure 1: Heartbeat Protocol.

secret keys.
Fig.2 illustrates the general format of the request/response packets in Heartbeat protocol. The data from the sender is contained in the payload field according to the payload length field. In the server side, the data will be copied to the memory. When the server prepares the response packet, it will check the payload length field to retrieve the indicated number of bytes from its own memory and include it in the response packet. As mentioned before, when the Heart-

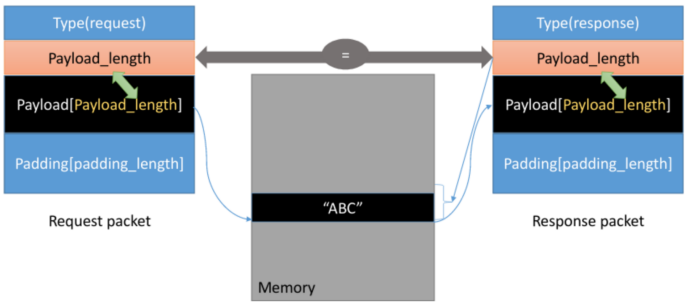

Figure 2: Heartbeat Request and Response Packets.

Bleed attack occurs, the same payload is kept. However, the payload length field is modified to be larger than the original payload (see the example shown in Fig.3 with a value of 1003 bytes). The server then blindly takes the length value when building the response packet. At this point, the server points to the

original data and copies 1003 bytes from its own memory to build the response packet. Thus, besides the original data, extra 1000 bytes are copied which might contain some secret information.

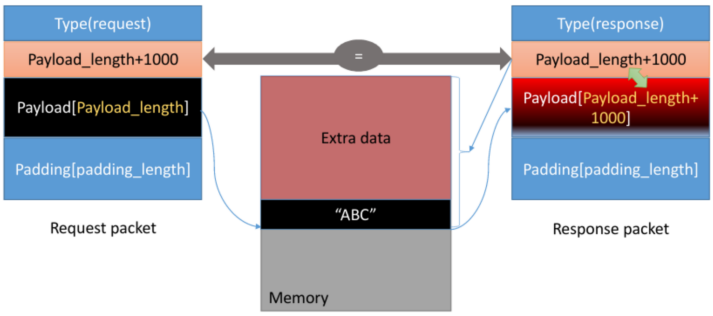

Figure 3: HeartBleed Attack

3 Experimental Setup

In this experiment, you need two systems running each on a VM, the attacker system and the victim system (Web server). You will need an Apache server with SSL support on the victim machine. Make sure to install OpenSSL version 1.0.1 on the server machine.

4 Procedure

4.1 Preparing the Environment

a. Confirm that both machines are fully updated using the commands **apt-get update** and **apt-get upgrade**.

b. Install OpenSSL version 1.0.1 on the server machine.

c. Install Apache web server on the server machine using the command **apt install apache2**.

d. Enable the SSL support for the apache server using the command **a2enmod ssl**

e. Restart the apache server.

f. Create a directory named **ssl** under **/etc/apache2**. This directory will be used to store the web-server key and its certificates.

g. Create a self-signed certificate using openssl library using the following command under the directory **/etc/apache2/ssl**
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/webserver.key -out /etc/apache2/ssl/webserver.crt.
Name the domain as **nis.najah.ps**. **Explain each part in the command**.

h. Configure the Apache server to use the certificate you have created. To do so, edit the file **/etc/apache2/sites-available/default-ssl.conf**. In your edit you should do the following:

- Add the following line to the file (after the line ServerAdmin webmaster .....):
  IPX:PortNum
  where IPX represents the IP address of your network interface and the port number equals to 443. **Why do you use this port?**
- Set SSLEngine to be on (SSLEngine on).
- Add the absolute path of webserver.crt file for SSLCertificateFile field.
- Add the absolute path of webserver.key file for SSLCertificateKeyFile field.

i. Activate you have done in the SSL configuration file using the command **2ensite default-ssl.conf**

j. Restart the apache server.

k. Use your web-browser to access the web-server with HTTPS. **Can you do that? Justify your answer**.

l. Use **nmap tool** to check the status of the https session.

m. Make sure that the web server is accessible by the other machine in your system.

n. Use Wireshark to capture Heartbeat request and response packets on the other machine (attacker machine).

4.2 Performing the HeartBleed Attack

To perform the attack, we will use Metasploit tool.

a. Follow all steps in the following website to install the Metasploit tool on the attacker machine. **https://websiteforstudents.com/how-to-install-metasploit-framework-on-ubuntu-18-04-16-04/**

b. Make sure that you have successfully installed the Metasploit framework.

c. Make sure that you have successfully setup the Metasploit database.

d. Open the metasploit console using the command **msfconsole**.

e. Set the auxiliary scanner to be openssl-heartbleed using the following command
use auxiliary/scanner/ssl/openssl-heartbleed

f. Check the options that you have for the attack using the command **show options**

g. Set the RHOSTS to be the IP address of the victim machine (web server) using the command **set RHOSTS**.

h. Make sure that the RHOSTS attribute has been successfully modified.

i. Set the verbose mode to true.

j. Set the TLS version to 1.1.

k. Run the attack using the command **run**.

l. Find the leaked information from the server machine and explain the results.

m. Use Wireshark to capture any leaked info from the server.

n. Upgrate OpenSSL version on the server and re-perform the attack. State your conclusions.

References

https://seedsecuritylabs.org/
https://alexandreborgesbrazil.files.wordpress.com/2014/04