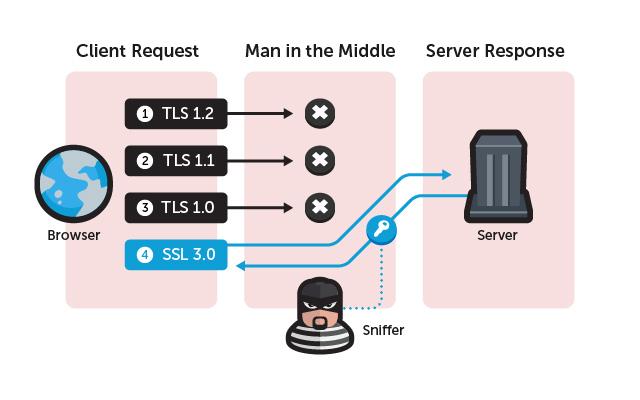
**Hands-On: POODLE ATTACK** 

Estimated time: 60-75 minutes

This workshop should be completed individually.

A **POODLE attack** is an exploit that an attacker can utilize to take advantage of the way browsers handle encryption. A **P**adding **O**racle **O**n **D**owngraded **L**egacy **E**ncryption can directly target browser-based communication that relies on Secure Socket Layer(SSL) 3.0 protocol for encryption and authentication. An MitM (man in the middle) attacker may use this vulnerability by exploiting POODLE and inserting themselves into a communication session and forcing the browser to use SSL 3.0.



**Step 1 – Use Nmap to target exploitable server**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Nmap** is a tool used as a network scanner; Nmap can discover hosts and services on a computer network by sending packets and analyzing the responses. In this case, we will utilize Nmap to check the TLS/SSL version of the server.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. We are now going to check if the address is vulnerable to the attack. Enter this command into the terminal in the attacker machine:

**nmap -sV --version-light --script ssl-poodle -p 443 <enter victim server ip>**

Based on your observations, in your own words, do your best to briefly describe **why an attack would use Nmap for a POODLE attack:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 2 – Initiate attack**

1. Open a terminal window in the attacker machine, you will use the following command to find the machines’ IP address:

**Ifconfig**

*Hint: the address will be listed under the eth0 network*

**Attacker IP Address \_\_\_\_\_\_\_\_\_\_\_\_**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The python script being run will act as a proxy, this will allow easier implementation of the POODLE attack.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. In the same attacker machines terminal you will write the following command:

**python3 httpserver.py https://*step1IPAddressHere*//**

*Hint: the IP address will be the same one found as in step 1*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The shell code will set up the environment for a POODLE attack by incorporating bettercap arp.spoof and enabling port forwarding. In most cases, this can be done manually.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. In a second terminal tab within the attacker machine you will write the following commands:

**iptables -i eth0 -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 4443**

**bettercap -iface eth0**

**set arp.spoof.internal true**

**arp.spoof on**

*Keep the two terminal windows OPEN during the duration of the workshop*

1. Open a third terminal in the attacker machine where you will execute the following command:

**python3 ./poodle-exploit.py *attackerIP* 4443 *serverIP* 443 --start-block 49--stop-block 52**

Based on your observations, in your own words, do your best to briefly describe **what you think the --start-block and --stop-block is used for:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 3 – Create a user/session cookie**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A ***user/session cookie*** is used because webpages have no memory; session cookies allow the webpage you are visiting to keep track of your movement from page-to-page so the information you’ve already given the site to refresh and recurrently ask you for your information.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Open ***https:VulnerableServerIP*** address in a browser window on the victim machine and sign in with the following credentials:

Username: **John**

Password: **mypasswd**

Based on your observations, in your own words, do your best to briefly describe **why you think a session cookie is user/created for this attack:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 3 – Validating MitM attack**

1. Open ***https:AttackersIPAddress*** in another browser window on the victim machine and click the button “Ping HTTPS Server”.
2. Go back to the attacker machine and the script ran in Step 2 should now receive the package length and output it to the user.

**Received package length:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Step 4 – TLS/SSL Downgrade**

1. If the protocol is not SSLv3.0 type ***downgrade*** into the terminal window with the received package length to initiate a TLS attack.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In some cases, the downgrade may not work; so for the purposes of this scenario we must manually downgrade the browsers TLS protocol. Open ***about:config*** in a browser tab in the victim machine and set ***security.tls.version.max*** to ***0.***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Optional Steps*: Decryption attack**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A ***decryption attack*** is a separate exploitation of information that is used by an attacker to gain information from a user. In this case, it is a supplementary attack that you can choose to do. In this attack you will decrypt the CBC information that you have gained from the previous attack.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. In the same terminal on the victim machine that has taken input such as ***downgrade***, type the ***search*** command and click find **CBC block length**.

**Block Length:\_\_\_\_\_\_\_\_\_\_\_\_**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CBC** (Cipher Block Chaining) is a mode of operation that provides message dependence for generating ciphertext and makes the system non-deterministic( an algorithm that, even for the same input, can exhibit different behaviors on different runs).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. Once the script has found the block length, type the ***active*** command into the terminal and choose **Run Decryption**.

**Decrypted CBC:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**