

Observational Report LAB 3: ARP CACHE POISIONING ATTACK

ACS 545| Cryptography and Network Security

January 31, 2024

Ashutosh Mishra

Professor: ZesHeng Chen

* Observation Criteria: -

Note: -

1. Used Ubuntu Seed (v20.04 Focal) on Google Cloud Platform
2. Used Real VNC Viewer to perform the actions on GUI
3. All actions are being performed under username - **seed**

* Environment Setup: -

A folder named Crypto was created and the Labsetup file was downloaded from the Seed Documentation site. Post unzipping the folder we find a docker-compose.yml file which contains the configurations for three machines that are connected to the same LAN. The three machines are: -

1. Host A (10.9.0.5)
2. Host B (10.9.0.6)
3. Host M (10.9.0.105)

To create the three containers using the .yml file, firstly one can use either **docker-compose build** or **dcbuild** (alias).

A screenshot of a computer

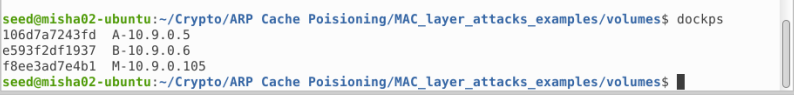
Description automatically generated

As per the attached image, we find that the steps are being skipped. The reason for this is because the containers have already been created. Now we will use either the **docker-compose up** or **dcup** (alias) to start the containers.

A screenshot of a computer

Description automatically generated

Now, as per the above screenshot, we can see that the containers are up. However, just to cross check we can use either **docker ps –format “{{.ID}} {{.Names}}”** or **dockps** (alias) to show the status of the containers.



* Task 1 (ARP Cache Poisoning): -
  + Task 1 A (Using ARP Request): -

For the first task, I have referenced the code shared in class. In this script there is a very small change as per the asked question. The requirement is that on host M, we need to construct an ARP request packet that is mapped to B’s Ip address to M’s MAC address, further sending a request to A. So, I have replaced the IP\_T(Target) field with B’s IP and Changed M’s fake MAC address to actual.

**Note: - In a real scenario, it’s wise to keep a random MAC address or changing MAC address to avoid being caught.**

A screenshot of a computer

Description automatically generated

Before running this program, we need to netsh to Machine M and make sure that net cache is cleared in both A and B.

A screenshot of a computer

Description automatically generated

As we can see, right now both the entries are empty. Now, we run the script from Machine M and check the output. Based on output below we can see that now 10.9.0.6 has a MAC of Machine M successfully. Thus the attack was successful.

A screenshot of a computer

Description automatically generated

* + Task 1 B (Using ARP Reply): -

For part B task, I have referenced the code shared in class. In this script there is a very small change as per the asked question. The requirement is that on host M, we need to construct an ARP reply packet that is mapped to B’s Ip address to M’s MAC address, further sending a request to A. So, I have replaced the IP\_T(Target) field with B’s IP and Changed M’s fake MAC address to actual.

**Note: - In a real scenario, it’s wise to keep a random MAC address or changing MAC address to avoid being caught.**

**A screenshot of a computer

Description automatically generated**

**Scenario 1: B’s IP is already in A’s Cache**

We will clear the caches from both A and B and ping A from B to generate an entry.

A screenshot of a computer

Description automatically generated

We can see that both Machine A and B have real entries. Now, we launch the attack from Machine M and check if there is any change.

A screenshot of a computer

Description automatically generated

We can see that the attack was successful as there was a change of MAC for Machine B on Machine A’s cache.

**Scenario 2: B’s IP is not in A’s Cache**

We will delete caches from both Machine A and B and run the attack from Machine M and check the output.

A screenshot of a computer

Description automatically generated

As we can see from the above screenshot, there was no new entry made in A’s cache. The reason is that we sent a Request package without A requesting anything. Hence, the attack failed.

* + Task 1 C (Using ARP Gratuitous Message): -

For part C task, I have referenced the code shared in class. In this script there is a very small change as per the asked question. The requirement is that on host M, we need to construct an ARP gratuitous packet that is mapped to B’s Ip address to M’s MAC address, further sending a request to A. So, I have replaced the IP\_T(Target) field with B’s IP and Changed M’s fake MAC address to actual.

**Note: - In a real scenario, it’s wise to keep a random MAC address or changing MAC address to avoid being caught.**

A screenshot of a computer

Description automatically generated

**Scenario 1: B’s IP is already in A’s Cache**

We will clear the caches from both A and B and ping A from B to generate an entry.

A screenshot of a computer

Description automatically generated

We can see that both Machine A and B have real entries. Now, we launch the attack from Machine M and check if there is any change.

A screenshot of a computer

Description automatically generated

We can see that the attack was successful as there was a change of MAC for Machine B on Machine A’s cache.

**Scenario 2: B’s IP is not in A’s Cache**

We will delete caches from both Machine A and B and run the attack from Machine M and check the output.

A screenshot of a computer

Description automatically generated

As we can see from the above screenshot, there was no new entry made in A’s cache. The reason is that we sent a Request package without A requesting anything. Hence, the attack failed.

* Task 2 (MITM Attack on Telnet using ARP Cache Poisoning): -

For the second task, I have referenced the code shared in class for Task 1 and modified it heavily as shown below.

**Note: I have given a sleep time for 8 seconds to effectively take screenshots.**

A screenshot of a computer

Description automatically generated

Now, we will run the ARP cache poisoning attack from Machine M

A screenshot of a computer

Description automatically generated

We can see that the attack was successful, and both the cache MAC ids refer to Machine M’s. We will now perform the same attack post setting **sysctl net.ipv4.ip\_forward=0**. Also, we will generate a Wireshark report for the same.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

After setting the flag to 0, the telnet crashes and nothing can be typed.

A screenshot of a computer

Description automatically generated

Now we try the same thing by setting the flag to 1.

A screenshot of a computer

Description automatically generated

The telnet session works and we capture the wireshark packages

A screenshot of a computer

Description automatically generated

The script for MIM attack is as shown below. I have referenced to the example code provided by professor in class and created the same. There is a catch to run this code. We need to first run the ARP Poisoning script from Host M. Then we need to create a Telnet session. Change the flag to 0 and finally run the MIM attack. This will then cause a glitch where whenever we type something we get a response of ZZZZZ\*.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Task 3 (MITM Attack on Netcat using ARP Cache Poisoning): -

The task is like Task 2, however, there is a small change. Except for using Telnet in both Machines, A and B, we will instead use Netcat.

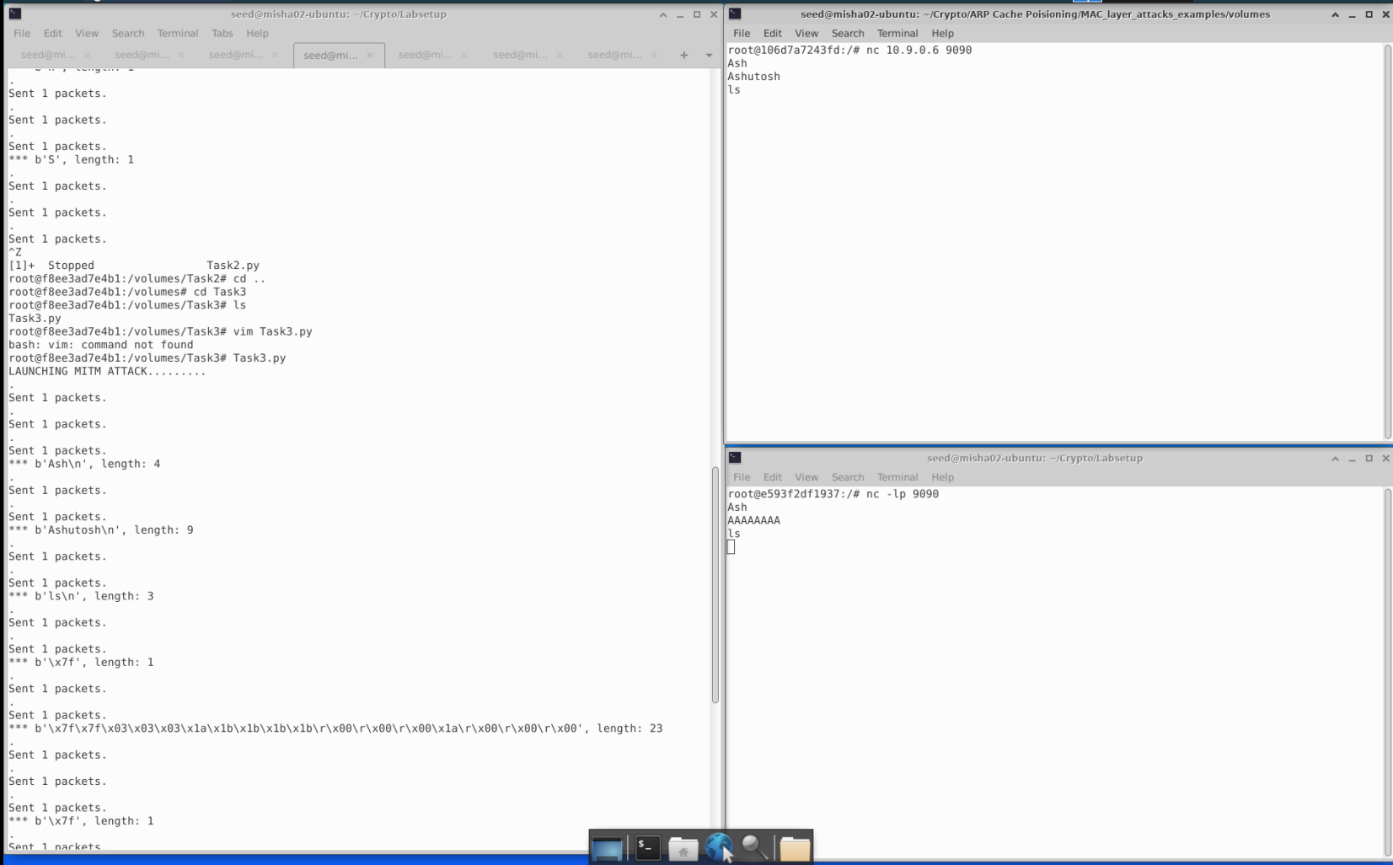
We will first change the script slightly as shown below: -

A screenshot of a computer

Description automatically generated

I have slightly altered and except for using re I used replace function to replace whenever I type Ashutosh (My first name) with AAAAAAAA.

To begin execution for this task, we will simply run the modified code on Machine M and before that generate a nc session from A.



As we can see, the attack was successful.

* Conclusion: -

The overall experience was exciting. I will be going deeper into the subject and try to discover new ways to handle how we had practiced in lab. Having a base knowledge of how ARP Attacks work in general, I will like to do a deep dive in this subject to find out more about how to protect the target machines from attacks that occur in a simple manner.