

Observational Report LAB 7: Local DNS Attack Lab

ACS 545| Cryptography and Network Security

March 13, 2024

Ashutosh Mishra

Professor: ZesHeng Chen

* Observation Criteria: -

Note: -

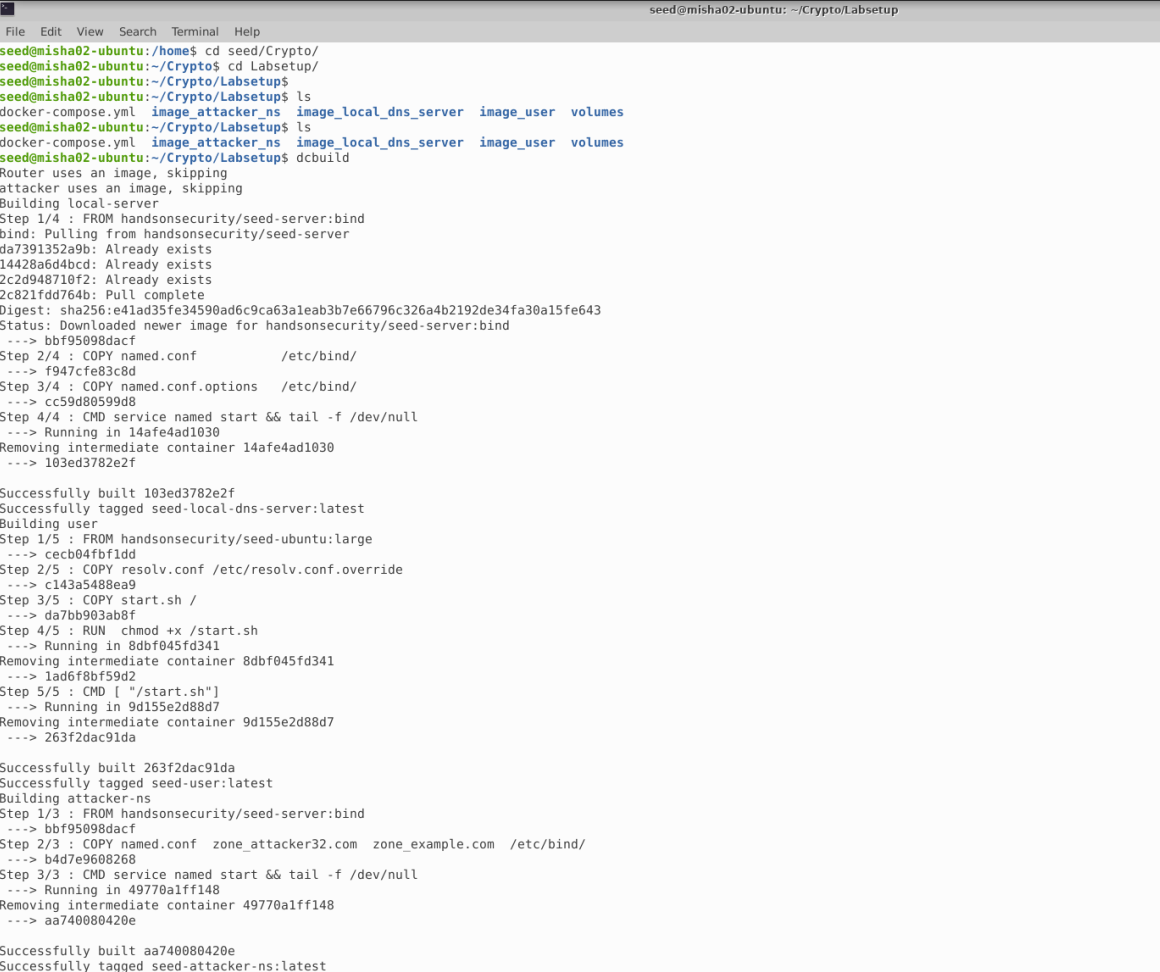
1. Used Ubuntu Seed (v20.04 Focal) on Google Cloud Platform
2. Used Real VNC Viewer to perform the actions on GUI
3. All actions are being performed under username - **seed**

* 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 four machines are: -

1. Attacker-ns-10.9.0.153
2. Local-dns-server-10.9.0.53
3. Seed-attacker
4. Seed-router
5. User-10.9.0.5

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



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.

A screenshot of a computer

Description automatically generated

* Testing the DNS Setup: -

In this task, we need to check if we can fetch the Ip addresses from the user container. Let’s first docksh to user container.

A screenshot of a computer

Description automatically generated

* + Get the IP address of ns.attacker32.com

Let’s try to find the Ip address for ns.attacker32.com

A screenshot of a computer

Description automatically generated

As we can see, we get 1 answer which is the ip address mentioned in the attacker nameserver.

* + Get the IP address of www.example.com

Let’s try to find the Ip address for www.example.com.

A screenshot of a computer

Description automatically generated

As we can see that the answer section has the actual Ip address for the site.

* + Get the fake IP address of www.example.com

A screenshot of a computer

Description automatically generated

This is the fake Ip address of www.example.com.

* Task 1 (Directly Spoofing Response to User): -

For this task, I have used the reference solution mentioned in the class. Please find a screenshot of the code below.

A screenshot of a computer

Description automatically generated

Let’s run the attack on the attacker machine

A screenshot of a computer

Description automatically generated

Now let’s run the dig command in user.

A screenshot of a computer

Description automatically generated

The attack failed as the local dns server already had the cache. Let’s flush the cache.

A screenshot of a computer

Description automatically generated

Now, let’s rerun the dig command in user system.

A screenshot of a computer

Description automatically generated

We have successfully performed the attack and now ip is pointed to the fake one.

* Task 2 (DNS Cache Poisoning Attack – Spoofing Answers): -

For this task the code will remain the same. The only change will be to change the call of the python script in attacker to point to DNS Server rather than the user.

A screenshot of a computer

Description automatically generated

Now let’s check output in the user terminal.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

We can observe that the attack is successful every time.

Also the cache file looks as following: -

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Task3 (Spoofing NS Records): -

We will modify the python script slightly as shown below.

A screenshot of a computer

Description automatically generated

Now let’s clear the cache in local dns server and relaunch the attack.

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

Now let’s dig example.com and see if the attack was successful.

A screenshot of a computer

Description automatically generated

The attack is successful and let’s check the records in local dns server.

A screenshot of a computer

Description automatically generated

To test let’s check aaaaa.example.com. This will point to 1.2.3.6.

A screenshot of a computer

Description automatically generated

* Task 4(Spoofing NS Records for Another Domain): -

In this task the script is slightly modified as shown below.

A screenshot of a computer code

Description automatically generated

Now, let’s clear cache in local DNS Server and relaunch the attack.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Now, let’s run the dig command and check if the attack was successful.

A screenshot of a computer

Description automatically generated

The attack was successful. Let’s check the records in local DNS Server to see if there is record for google.com.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The record is not present. The reason for the same is because previously we didn’t call google.com. Hence, even if there is an additional record it won’t be published here.

* Task 5(Spoofing Records in the Additional Section): -

For this task the modified code is as below.

A screenshot of a computer

Description automatically generated

Let’s clear the cache and rerun the attack

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Now let’s perform dig www.example.com.

A screenshot of a computer

Description automatically generated

The attack was successful. Let’s check the records at the local dns server.

A screenshot of a computer

Description automatically generated

Like the previous task, the attack is successful. However, the additional records don’t pop up here as Facebook and attacker have nothing to do with www.example.com.

* 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 DNS works, I would like to do a deep dive into this subject to find out more about how to protect the target machines from attacks that occur in a simple manner.