ICMP Redirect Attack Lab

Contents

Lab Setup

Please download the Labsetup.zip file from the below link to your VM, unzip it, enter the Labsetup folder, and use the docker-compose.yml file to set up the lab environment.

https://seedsecuritylabs.org/Labs_20.04/Files/ICMP_Redirect/Labsetup.zip

In this lab, we need several machines. The lab environment setup is depicted in Figure 1. We use containers to set up this environment.

We will use the attacker container to launch attacks. We assume all these machines are on the same LAN.

Note: When we use the attacker container to launch attacks, we need to put the attacking code inside the attacker container. Code editing is more convenient inside the VM than in containers, because we can use our favorite editors. Hence it is advisable for you to place your respective codes in the "volumes" folder directly (using gedit for example).

Lab Overview

An ICMP redirect is an error message sent by a router to the sender of an IP packet. Redirects are used when a router believes a packet is being routed incorrectly, and it would like to inform the sender that it should use a different router for the subsequent packets sent to that same destination. ICMP redirect can be used by attackers to change a victim's routing.

The objective of this task is to launch an ICMP redirect attack on the victim, such that when the victim sends packets to 192.168.60.5, it will use the malicious router container (10.9.0.111) as its router. Since the malicious router is controlled by the attacker, the attacker can intercept the packets, make changes, and then send the modified packets out. This is a form of the Man-In-The-Middle (MITM) attack.

This lab covers the following topics -

- The IP and ICMP protocols
- ICMP redirect attack
- Routing

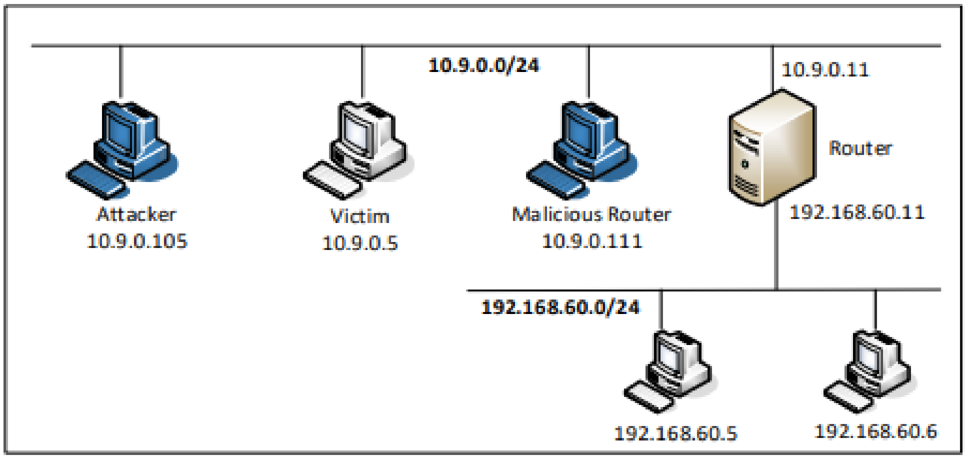

Figure 1

Task 1: Launching ICMP Redirect Attack

In the Ubuntu operating system, there is a countermeasure against the ICMP redirect attack. In the Compose file, we have already turned off the countermeasure by configuring the victim container to accept ICMP redirect messages.

```
// In docker-compose.yml
       sysctls: - net.ipv4.conf.all.accept_redirects=1
// To turn the protection on, set its value to 0
      # sysctl net.ipv4.conf.all.accept_redirects=0
```

For this task, we will attack the victim container from the attacker container. In the current setup, the victim will use the router container (192.168.60.11) as the router to get to the 192.168.60.0/24 network.

To check this run the following on the Victim Machine -

Command:

```
# ip route
```

Run the following command on the Victim Machine to remove the countermeasure -

Command:

```
# sysctl net.ipv4.conf.all.accept_redirects=1
```

Take a screenshot of the same.

Task 1A - In order to perform the attack i.e make the Victim Machine route its packets through the Malicious router, follow the steps mentioned below.

1. First we ping the Host - 192.168.60.5 from the Victim Machine

   Command:

   ```
   # ping 192.168.60.5
   ```

2. Then we run the following code on the Attacker Machine

   Command:

   ```
   # python3 task1A.py
   ```

3. ICMP redirect messages will not affect the routing table; instead, it affects the routing cache. Entries in the routing cache overwrite those in the routing table, until the entries expire. To check if we have successfully executed the attack, check the routing cache on the Victim Machine.

   Command:

   ```
   # ip route show cache
   ```

4. Now run a traceroute on the victim machine, and see whether the packet is rerouted or not.

   Command:

   ```
   #  mtr -n 192.168.60.5
   ```

Take a screenshot of the attacker and victim machines, explain your observations.

After you have succeeded in the attack, flush the router cache on the Victim Machine

Command:

```
ip route flush cache
```

Questions. After you have succeeded in the attack, please conduct the following experiments, and see whether your attack can still succeed. Please explain your observations:

- Question 1: Can you use ICMP redirect attacks to redirect to a remote machine? Namely, the IP address assigned to **icmp.gw** is a computer not on the local LAN. Please show your experiment result, and explain your observation.

Perform the earlier mentioned steps (Task 1A), but now instead of running **task1A.py run task1B.py**. Students can change the IP address assigned to icmp.gw to one of their own liking. Flush the router cache after each step.

- Question 2: Can you use ICMP redirect attacks to redirect to a non-existing machine on the same network? Namely, the IP address assigned to icmp.gw is a local computer that is either offline or non-existing. Please show your experiment result, and explain your observation

Perform the earlier mentioned steps (Task 1A), but now instead of running **task1A.py run task1C.py**. Students can change the IP address assigned to icmp.gw to one of their own liking. Flush the router cache after each step.

Take screenshots and illustrate your observations.

- Question 3: If you look at the docker-compose.yml file, you will find the following entries for the malicious router container. What are the purposes of these entries? **Please change their value to 1, and launch the attack again. Please describe and explain your observation.**

```
sysctls:
   - net.ipv4.conf.all.send_redirects=1
   - net.ipv4.conf.default.send_redirects=1
   - net.ipv4.conf.eth0.send_redirects=1
```

Restart the docker containers then follow the initially mentioned steps i.e **Task1A** (Ignore question 1 and 2 for this step).

Before proceeding to the next task, restore the docker-compose files to the original. Then execute Task 1A in order to make the Victim Machine route its packets through the Malicious router. Check the Victim's Cache in order to verify the same.

Task 2: Launching the MITM Attack

Using the ICMP redirect attack, we can get the victim to use our malicious router (10.9.0.111) as the router for the destination 192.168.60.5. Therefore, all packets from the victim machine to this destination will be routed through the malicious router. We would like to modify the victim's packets.

Note - You will need Wireshark for this task, capture the packets on the container interface for Hosts (192.168.60.x) . Also note the victim's router cache expires quickly, so please check the cache to make sure it has been redirected, else perform Task 1A again.

Task 2A - Netcat Connection

Before launching the MITM attack, we start a TCP client and server program using netcat.

On the destination container 192.168.60.5, start the netcat server:

Command:

```
# nc -lp 9090
```

On the victim container, connect to the server:

Command:

```
# nc 192.168.60.5 9090
```

Once the connection is made, you can type messages on the victim machine. Each line of messages will be put into a TCP packet sent to the destination, which simply displays the message. Take a screenshot of both the terminals (victim and host) and the wireshark packet capture.

Task 2B - To launch the MITM Attack

Your task from now is to replace every occurrence of your first name in the message with a sequence of A’s. The length of the sequence should be the same as that of your first name, or you will mess up the TCP sequence number, and hence the entire TCP connection. You need to use your real first name, so we know the work is done by you.

Now disable IP Forwarding - In the setup, the malicious router’s IP forwarding is enabled, so it does function like a router and forward packets for others. When we launch the MITM attack, we have to stop forwarding IP packets; instead, we will intercept the packet, make a change, and send out a new packet. To do that, we just need to disable the IP forwarding on the malicious router.

In the mitm.py code, change “seedlabs” to your name, and add or reduce the number of ‘A’ characters accordingly. Check the victim router’s cache, if empty perform Task 1A again, then establish the netcat connection before proceeding further.

On the malicious router terminal turn off ip forwarding

Command:

```
# sysctl net.ipv4.ip_forward=0
```

MITM code. Once the IP forwarding is disabled, our program needs to take over the role of packet forwarding from the victim to the target, of course after making changes to the packets. Since the packet’s destination is not for us, the kernel will not give the packet to us; it will simply drop the packet. However, if our program is a sniffer program, we will get the packet from the kernel. Therefore, we will use the sniff and spoof technique to implement this MITM attack. In the following, we provide a sample sniff-and-spoof program, which captures TCP packets, makes some changes, before sending them out.

On the malicious router terminal run the mitm attack.

Command:

```
# python3 mitm.py
```

Now on the Victim Machine's netcat connection window **type in the name you previously entered in the mitm.py code**, you should be able to see the respective 'A' characters on the Host - 192.168.60.5 window.

Take screenshots of all the terminals and that of wireshark. Explain your observations.

Questions. After you have succeeded in the attack, please answer the following questions:

- Question 4: In your MITM program, you only need to capture the traffic in one direction. Please indicate which direction, and explain why.

- Question 5: In the MITM program, when you capture the nc traffic from A (10.9.0.5), you can use A's IP address or MAC address in the filter. One of the choices is not good and is going to create issues, even though both choices may work. Please try both, and use your experiment results to show which choice is the correct one, and please explain your conclusion

- **For using A's IP address as a filter, change the variable 'f' (mitm.py) value to - 'tcp and src host 10.9.0.5'**
- **For using A's MAC address as a filter, change the variable 'f' (mitm.py) value to - 'tcp and ether host 02:42:0a:09:00:05'**

Perform the above steps again (establish the netcat connection and launch the attack) and state your observations with appropriate screenshots.
Incase of any error, you may have to execute Task 1A again, as the router cache might have been invalidated. Make a new netcat connection and repeat the attack again.

Submission

You need to submit a detailed lab report to describe what you have done and what you have observed; you also need to provide explanations for the observations that are interesting or surprising. Please also list the important code snippets followed by an explanation. Simply attaching code without any explanation will not receive credits.