7/7/2019

Vanye Wadawasina

1228377

ARP SPOOF / DETECOR PROJECT

Intrusion Detection System: Yasir Malik

Table of Contents

Abstract 2

Problem Statement 2

Review of related work 2

Proposed solution 2

Methodology 2

Experiment 3

Phase 1 (Attack) 3

Phase 2 (Defense) 3

Test Procedure design / Flow chart 3

Flow chart 4

Data Analysis & Discussion 5

Attack 5

Defence 5

Conclusion 5

Reference 6

Appendix 6

Attack 6

Defence 7

# Abstract

This project demonstrated the security flaw of address resolution protocol. The project showed how and ARP spoofing attack can be carried out, disrupting the flow of network between a client and the gateway / router. In addition, the project also highlighted possible counter measures of this security problem, by developing a program that actively sniffs packets and compare the mac address of each packet to the original MAC address. During entire course of the project, both programs were developed using python programming language with Scapy library.

# Problem Statement

Address resolution protocol is a method that maps IP address to a hardware MAC address. (geeksforgeeks, 2019). It is used when devices in a network communicate with each other. Arp can be found in the data link layer of the OSI (open systems interconnection). One of the drawbacks of ARP is security. When two devices on the same network want to communicate with each other, device A sends an ARP request to the broadcast Ethernet which is visible to all devices on the network asking for the MAC of Address of device B through its IP address. Once the broadcast is made, device B answers and returns in MAC Address by acknowledging the IP Address broadcasted by the Ethernet. The drawback is, a 3rd party acknowledging the transmitted IP address and sending its own MAC address can carry out a man in the middle attack. By doing this, all traffic flow from A to the 3rd party.

# Review of related work

This entire project was based on a tutorial course bought on Udemy. During the tutorial, the instructor showed the students how to manipulate packets, send and intercept requests. In the tutorial, the instructor created both programs, but they were not collecting using input. The programme which was created asks for user inputs, making it more dynamic because you don’t have to open the code and input the IP address manually into the function.

# Proposed solution

The main goal of this project is to design an ARP spoofer that would intercept ARP request between two systems. The Attack carried out would be a Man in the middle attack, which would direct all traffic from a to the evil attacker. The 2nd part of the project which is the proposed solution would involve creating an ARP spoofing detector which would alert the victim of any potential attack by continually monitoring the MAC address packets received from the gateway.

# Methodology

Both programs would be created with Python programming language. The core library which would be used is;

Scapy. This is a packet manipulation library written in python. It allows the user to create, sends and manipulates packets in a network. (Udemy, 2019)

Sys: This is a module that provides access to some variables used and maintained by the python interpreter. It will be used to flush out values from the buffer. (Udemy, 2019)

Time: This module will be used to create a timer variable, which will continuously send spoofed packets to both victim and gateway. (Udemy, 2019)

# Experiment

The experiment will be carried out in the following way.

### Phase 1 (Attack)

* Target: Kali Linux clone machine.
* Gateway: Virtual Router network (Same network address).
* Attacker: Kali Linux.

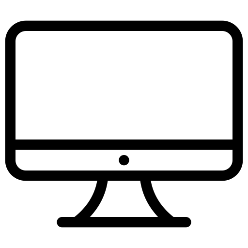
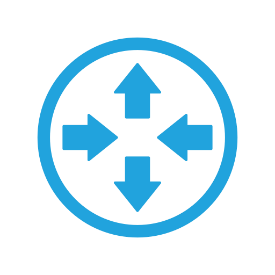
1. The 1st step of the experiment will be to build the spoofing program on python. Once both programmes are developed; ARP Spoofing program and ARP Spoof detector program
2. The next step will be to run the programme as the attacker.
   1. The code will be executed on the Kali Linux terminal by executing the following command
      1. Python {programme name}.
3. Once the programme has been executed, the attacker will be required to input the target Ip address as well as the gateway Ip address, fooling both the target and the gateway into believing that he is the target and at the same time the gateway. Once this happens, the flow of the network will be through the attacker system, making him the man in the middle. Figure 1 depicts the attack

### Phase 2 (Defense)

* Host-based ARP detector
  + The detector would constantly be running on the victim’s machine.

1. The program will be executed on the victim’s terminal window. It would be running on the terminal until a keyboard interrupt.
2. Once an ARP spoof has been detected, the victim would get a notification message warning of a potential attack.

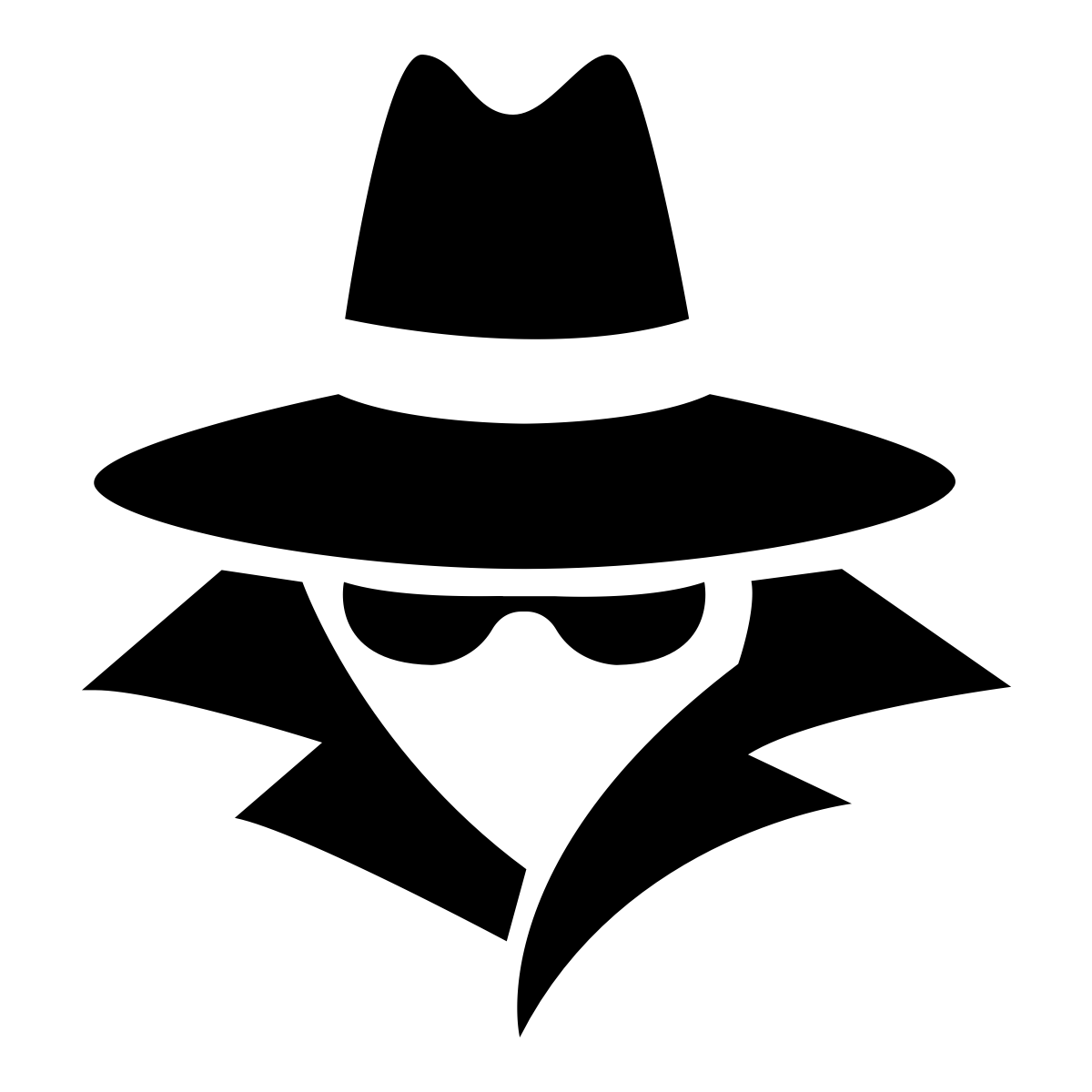
## Test Procedure design / Flow chart



**Normal Flow of packets**

**Gateway (vm)**

**Victim (Kali Linux clone VM)**



**Spoofed flow**

**Attacker (Kali Linux)**

##### Figure 1

### Flow chart

#### Attack

Execute python script

Enable port forwarding to allow the packets flow through your system

Intercept and study the flow of packets between victim and gateway

##### Figure 1.1

#### Defence

##### Figure 1.2

No

Yes

Sniff response packet only (Packet with OP filed set to 2)

Get valid mac address by performing a separate broadcast

Is Valid mac address the same with mac on the sniffed response packet

Raise alarm

Keep running

# Data Analysis & Discussion

## Attack

The result from the experiment gave an in-depth view of how an ARP Spoof attack works. By executing the python program, the packets sent to victims and the gateway device were both spoofed with the attacker MAC address which was automatically inserted by the Scapy library fooling both the victim system and the gateway into believing they were the other. In other for this to work, the OP field of the packets was set to 2 instead of 1. This specifies if it’s a request or a response. By setting the OP field to 1; this becomes a request, e.g. (who has……) while 2; is a response which is sent to the broadcast (ff: ff:ff:ff:ff:ff). See attack appendix for more details.

## Defence

The ARP spoof detector programme was able to detect the spoofing attack once executed on the victim’s machine. It identified the attack by sniffing packets with OP field set to 2 and comparing the source MAC address with the valid mac address. It was able to get the correct MAC address by executing its broadcast packet which had the op field set to 1. By doing this, the programme retrieved the valid mac address of each response packet and comparing it to the MAC address of the packet field. See defense appendix for more details.

# Conclusion

In conclusion, this project provided me with a hands-on experiment on how to manipulate IP Packets using Scapy library on python. I was able to manipulate the packets destination and source mac address as well as create a function that could retrieve mac address of any user on the same network just by providing the IP address of the target. Furthermore, this project gave me an in-depth view of how the entire Address resolution protocol structure works, which is based on a broadcast address (ff:ff:ff:ff:ff:ff). This address is the heart of the ARP protocol. By sending a packet to this address, all devices on the same network will receive the broadcast. If the packet OP field is set to 2, then it’s a response, but if it's set to 1, then it’s broadcast. With this Knowledge, I was able to understand fully how the ARP Spoof attack and Detector worked, and this, in turn, led me into building both programmes without any major setback.

# Reference

Udemy: Python ethical hacking (2019).  Retrieved from <https://www.udemy.com/learn-python-and-ethical-hacking-from-scratch/learn/lecture/10273212?start=450#questions>

Computer Network: How ARP works? (2019, April 23). Retrieved May 2, 2019, from https://www.geeksforgeeks.org/computer-network-arp-works/

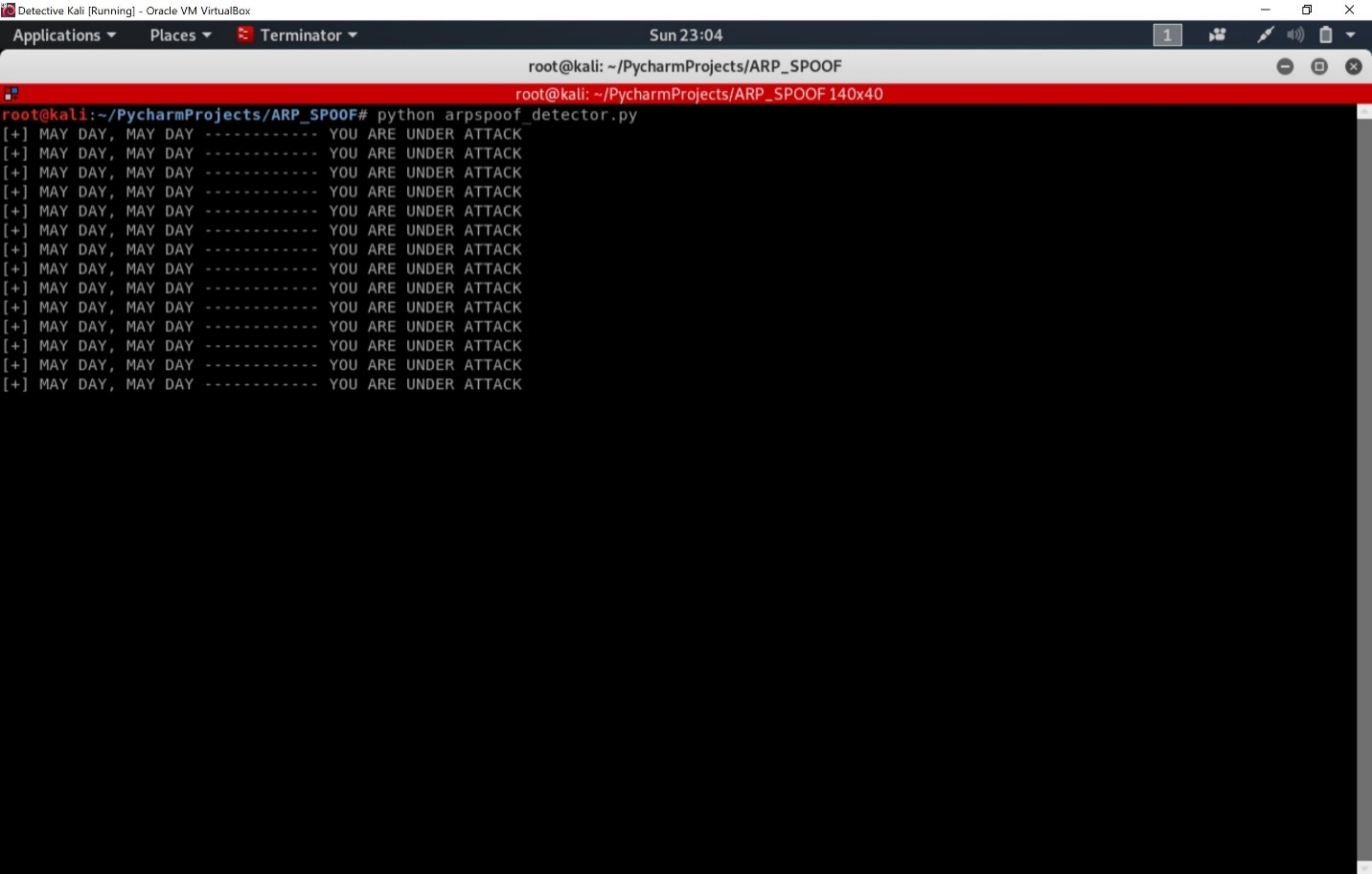
# Appendix

# Attack

##### Figure 2.1

1. Figure 2.1 illustrates the actual attack, which shows the number of spoof packets sent to the victim and the router system.
2. The 2nd part shows illustrate the packets flowing between the victim and the router. This demonstrates that the attack was a success. From the diagram, you could see that the attacker is receiving packets sent by both devices.
3. The last section is the command that enables port forwarding on the attacker system. Without this command, the packets won't flow out through the attacker machine, therefore, blocking the packets coming from the victim’s network. The attack will work, but the victim will not be able to access the network because packets are not flowing from the victim to the router

## Defence



##### Figure 2.2

1. Figure 2.2 illustrates the defence mechanism which was able to detect that the victim is under attack

The Project can be cloned on github https://github.com/vanye007/ARP\_SPOOFING