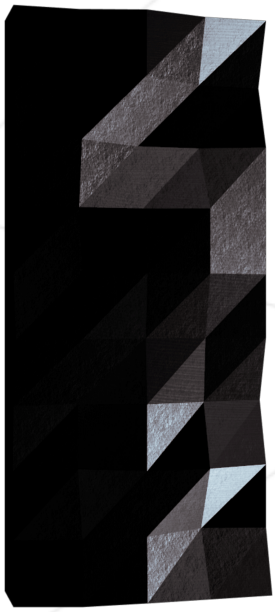

Cybersecurity Piscine

Inquisitor

Summary: ARP Poisoning.

Version: 1.00

Contents

Chapter I

Prologue

An innocent ARP poisoning victim.

Chapter II

Introduction

The so-called *OSI model* is the architecture followed by computer networks from all over the planet. It consists of 7 layers, each of which carries risks and is exposed to different types of vulnerabilities and forms of exploitation.

At the network level, there are elements in charge of deciding where to direct the traffic. Every local network has a default gateway, which receives external traffic and distributes it among its nodes. This gateway is usually also known as *router*.

If a network node is able to impersonate the gateway, it can take control of the traffic, intercept it and decide who to forward it to, as well as being able to modify or block it.

ARP spoofing can also be used legitimately, for example to redirect new connections to a registration page before using a network, as is common in networks open doors of airports, cafeterias and other public places.

Chapter III

Mandatory Part

Since working with *raw sockets* requires low-level permissions, in this project you will work inside a container or virtual machine.

In case of working with one or several containers, in addition to the code of your program you will include the Dockerfile or docker-compose.yaml as well as a Makefile that start the entire environment without user intervention.

You have to create a program called `inquisitor` with the following characteristics:

- It must be developed for the Linux platform.
- It must take at least these 4 parameters:
  - <IP-src>
  - <MAC-src>
  - <IP-target>
  - <MAC-target>
- The program must only work with IPv4 addresses.
- The program should never stop unexpectedly and must handle all input errors.

`The test part will be done using the FTP protocol. You must therefore prepare tests for this specific case.`

Your program must perform several actions described below:

- It must be able to perform ARP poisoning in both directions (full duplex)
- When the attack is stopped (CTRL+C), the ARP tables will be restored.

Your program should also be able to see the names of files exchanged between a client and an FTP server should be displayed in real time.

You can use the `libpcap` library to sniff the packets. Therefore, you can use any programming language that implements it (C, C++, Python, etc).

You have to prove that your program works. That's why you need to prepare a test suite using an FTP connection in addition to the other tests.

Chapter IV

Bonus Part

You can enhance your project with the following feature:

- "Verbose" (-v) mode that shows all FTP traffic and not just filenames. The program must be able to intercept the traffic resulting from the login to an FTP server.

The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter V

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.