Cybersecurity bootcamp | 42 Madrid

Inquisitor

Summary: ARP Poisoning.

Version: 1

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 using a virtual machine, in the delivery repository you will only include a file `signature.txt` with the checksum of your machine's .vdi. During the evaluation, the signature of the repository will be compared with the real signature of your machine, and if they do not match, your note it will be a 0.

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 Bash script called `start.sh` that start the entire environment without user intervention.

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

- Will receive four parameters: <IP-src> <MAC-src> <IP-target> <MAC-target>
- Will be able to perform ARP poisoning in both directions (full duplex)
- When the attack is stopped (CTRL+C), the ARP tables will be restored.
- Will only work with IPv4 addresses.
- The program will be able to intercept the traffic resulting from the login to an FTP server.
- The names of the files exchanged between the client and the FTP server will be displayed in real time.
- The program will never stop unexpectedly and will handle all input errors.

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

Chapter IV

Bonus Part

The evaluation of the bonuses will be done `IF AND ONLY IF` the mandatory part is `PERFECT`. Otherwise, the bonuses will be totally `IGNORED`.

You can enhance your project with the following features:

- "Verbose" (-v) mode that shows all FTP traffic and not just filenames.

Chapter V

Peer evaluation

This project will be corrected by your peers. Deliver the files to the Git repository and make sure everything works as expected.