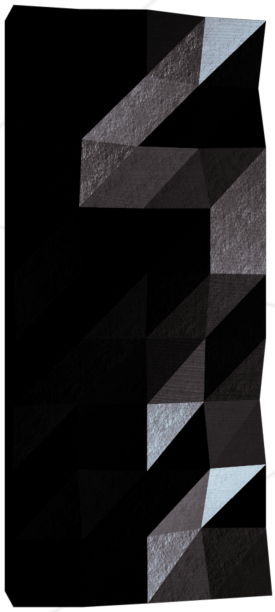

ft_malcolm

An introduction to Man in the Middle attacks

Summary: This is the first project of a network security branch created by maabou-h.

Contents

Chapter I

Foreword

Yes, no, maybe
I don't know
Can you repeat the question?
You're not the boss of me now
You're not the boss of me now
You're not the boss of me now, and you're not so big
You're not the boss of me now
You're not the boss of me now
You're not the boss of me now, and you're not so big
You're not the boss of me now, and you're not so big Life is unfair, so i just stare at the stain on the wall where
The tv'd been, but ever since we've moved in it's been empty
Why i, why i'm in this room
There is no point explaining
You're not the boss of me now, and you're not so big
You're not the boss of me now
You're not the boss of me now
You're not the boss of me now, and you're not so big
Malcolm in the middle, and i confess
I like this mess i've made so far
Grade on a curve and you'll observe
I'm right below the horizon
Yes,...

This subject has nothing to do with Malcolm in the middle, but rather with Man in the middle!

Chapter II

Introduction

In this first project of network security, you will implement the ARP spoofing/poisoning method, which is one of the most basic Man In The Middle attacks using a vulnerability present in the ARP protocol.

Chapter III

Goals

In this first project aimed at introducing you to network security, you will discover in details what is the Address Resolution Protocol, that you may have heard during your days of shell as ARP, and how it is used. You will find that this implementation has several vulnerabilities and while protections might have been put in place in large and/or important networks, it remains widely unsafe and unprotected in most cases.

`You should really start by reading the RFC 826`

Chapter IV

General instructions

• This project will be corrected by humans only. You're allowed to organise and name your files as you see fit, but you must follow the following rules

• You must use C and submit a Makefile

• Your Makefile must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.

• You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).

• Within the mandatory part, you are allowed to use the following functions:

  ◦ printf and its family.

  ◦ Your libft functions.

  ◦ You are allowed to use other functions to complete the bonus part as long as their use is justified during your defense. Be smart.

Chapter V

Mandatory part

Example usage:

```
$./harpoon --interface enp0s3 --ip 192.168.0.35 --mac 00:AA:11:BB:22:CC
```

or

```
$./harpoon --interface enp0s3 --ip 3232235555 --mac 33:DD:44:EE:55:FF
```

Your program must take into account the following parameters:
–interface
–ip
–mac

Where –interface is the network interface, –ip is the internet protocol address, either in dotted notation or in decimal notation, and –mac is the physical address you wish to force.

A help menu, via the –help option, or when no option is specified must be available in your program.

For this project, you will only have to spoof your own ARP table, but it should work on other IPs.

Beware! Don't try to spoof IPs outside your local network, as this is not legal in most if not all countries!!!

Your program must not stop poisoning the ARP cache until the user interrupts it with a Ctrl+C.

Chapter VI

Bonus part

We will look at your bonuses if and only if your mandatory part isEXCELLENT. This means that your must complete the mandatory part,beginning to end, and your error management must be flawless, even incases of twisted or bad usage. If that's not the case, your bonuseswill be totally IGNORED.

Find below a few ideas of interesting bonuses:

- IPv6 management.

Chapter VII

Turn-in and peer-evaluation

• Submit your work on your GiT repository as usual. Only the work on your repository will be graded.

• You have to be in a VM with a Linux kernel > 3.14. Note that grading was designed on a Debian 7.0 stable.