Ecole Nationale Supérieure de Bretagne Sud
Spécialité Ingénieur en Cyberdéfense

Phishing-WLAN

Authors

Vannes, le 19 Septembre 2015

Introduction

Our goal is to get the victim's credentials while he is trying to connect to a website.
This project takes place in our network course.
The aim of this project is to practice and learn how a particular aspect of a network works.

This project can be sum up by several steps :

- Scan the Network and choose a victim

- Setup a MITM Attack on the Local Area Network

  - ARP Spoofing

  - DNS Spoofing

  - steal SSL certificate

- Steal credentials

Man In The Middle attack schematized :
(https ://www.multicert.com/en/news/sha1-shellshock-and-poodle-web-threats/)

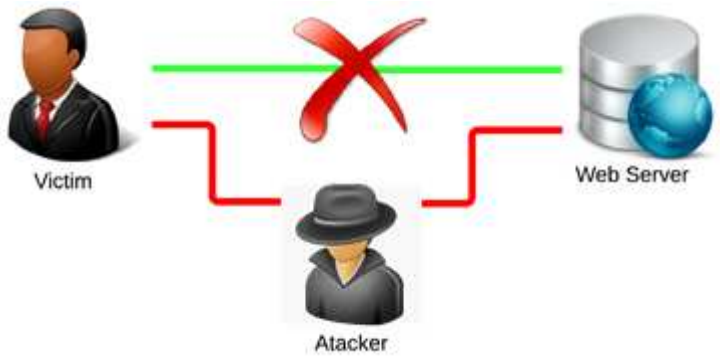

FIGURE 1 – Man In The Middle attack

1 What we are doing

1.1 Global Process

The first step of this attack is to look for a target by scanning the network, and getting his network gateway.
Then, we can make the target's routing table think that we are its gateway via ARPSpoofing,

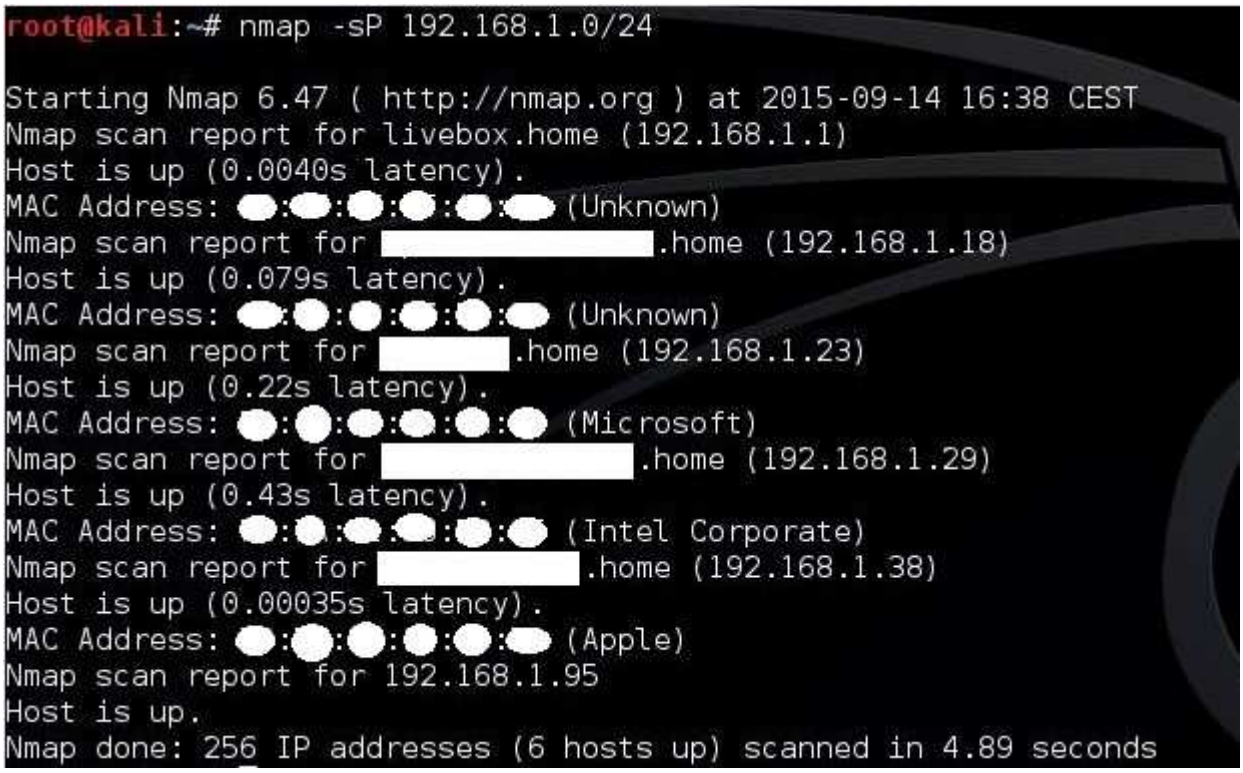

FIGURE 2 – nmap capture

and redirect all the DNS requests with some DNSSpoofing.
For the target, this is just a normal browsing over the internet.
For the attacker, he can see all the packets and get any information about the target.

1.1.1 ARP Step

Before spoofing the target, we can lookup on his ARP table and see the relation between IP adress and MAC adress.
With those informations, we can get the AP IP and MAC adress.

FIGURE 3 – ARP Table before spoof

FIGURE 4 – ARP Table after spoof

We can now start the ARPSoofing, this will have the effect of sending continuously the same ARP packets (level 2 messages) to the target.
The point in doing this, is to force the target to think that the gateway we are sending in the ARP messages is his gateway.
If we open wireshark to sniff the network, we are able to see a lot of those ARP messages from the attacker.

Example of an ARP spoof message :

`<MAC_PIRATE> tell <MAC_TARGET> that <IP_GATEWAY> is at <MAC_PIRATE>`

No.	Time	Source	Destination	Protocol	Length	Info
205	2015-09-14 16:48:58	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
207	2015-09-14 16:49:00	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
212	2015-09-14 16:49:02	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
216	2015-09-14 16:49:04	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
223	2015-09-14 16:49:06	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
230	2015-09-14 16:49:08	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
232	2015-09-14 16:49:10	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
238	2015-09-14 16:49:12	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
239	2015-09-14 16:49:13	00:37: : :	Broadcast	ARP	42	Who has 192.168.1.11? Tell 192.168.1.1
240	2015-09-14 16:49:14	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
245	2015-09-14 16:49:15	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
248	2015-09-14 16:49:17	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
249	2015-09-14 16:49:20	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
271	2015-09-14 16:49:21	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
284	2015-09-14 16:49:23	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
295	2015-09-14 16:49:25	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
299	2015-09-14 16:49:27	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
305	2015-09-14 16:49:29	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
317	2015-09-14 16:49:32	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
320	2015-09-14 16:49:33	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
332	2015-09-14 16:49:36	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
361	2015-09-14 16:49:37	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
375	2015-09-14 16:49:39	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
397	2015-09-14 16:49:41	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
402	2015-09-14 16:49:43	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
416	2015-09-14 16:49:46	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :
424	2015-09-14 16:49:48	Apple_ : :	IntelCor_ : :	ARP	60	192.168.1.1 is at : : : : :

FIGURE 5 – Wireshark capture of the ARP spoofing

Then, if we look to the target's ARP table, we can see that he uses our gateway.
At this point, the target can't reach the internet since he is asking to our gateway.
To make him communicate throw the attacker's gateway, we need to activate the IP forwarding.
So now, the target can surf on the internet, the MITM is in place, the hacker can see what is comming throw his connections.

1.1.2 DNS Step

This step is necessary to exploit a MITM attack.
The purpose of the DNS Spoofing is to intercept the target's DNS requests to be able, if required, to redirect them.

FIGURE 6 – DNS spoofing

Below, we can see that the targets is doing DNS requests aiming fbi.gov.

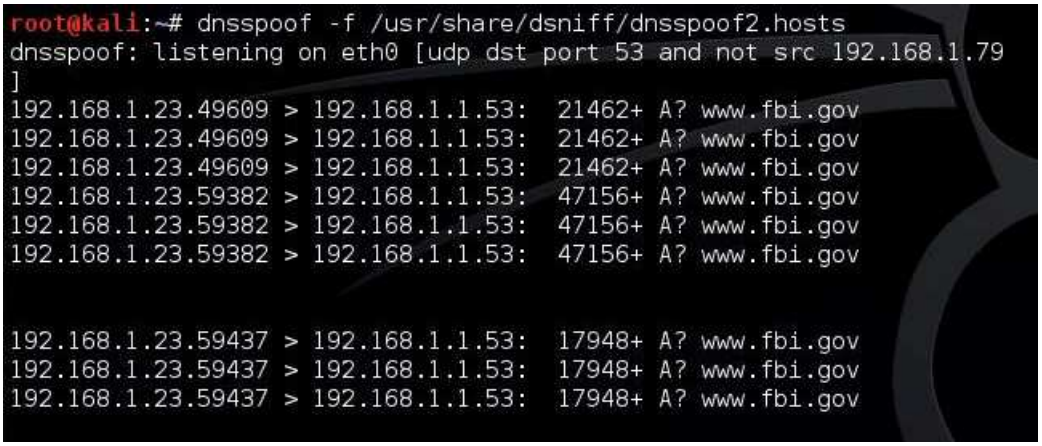

FIGURE 7 – DNS shell capture

Here, we can see the DNS requests on Wireshark.
A query from the target to fbi.gov (No. 66 and 67) then on the reply the IP adress for fbi.gov is the attacker's instead of the real IP adress.

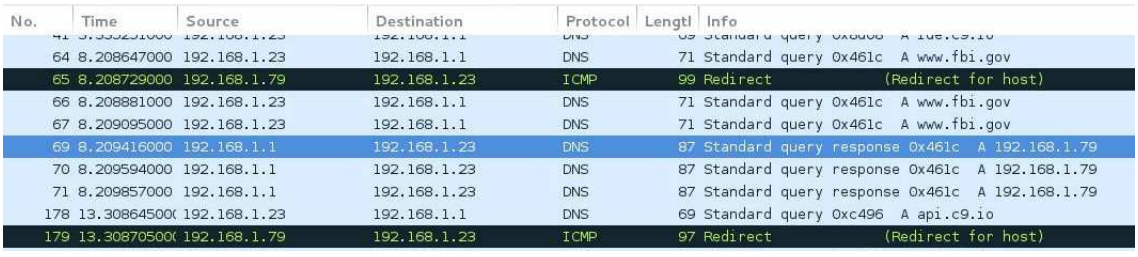

FIGURE 8 – DNS Wireshark capture

1.1.3 HTTP Server Step

Once the DNSSpoofing is runnig, the objective is to redirect the target to our own web service wich is a copy of the website ask by the target.
This copied website will be used to trap the target, he will think he is on the real site.

We have reproduced a fake gmail authentification page, hosted on our own HTTP server.

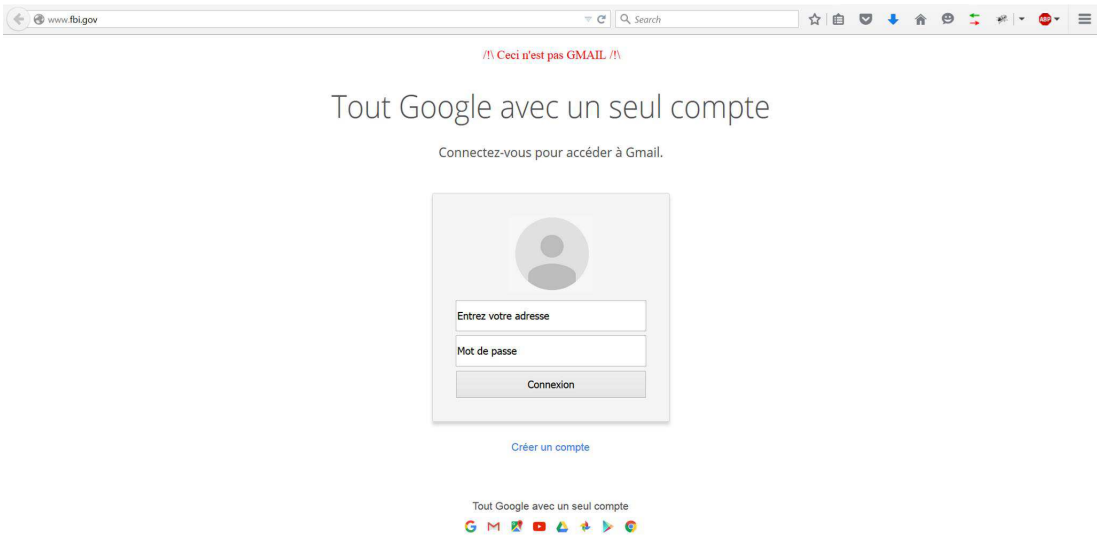

FIGURE 9 – Fake GMail

First, we used the SimpleHTTPServer tool that opens the 80 port and executes a simple HTML webpage.

FIGURE 10 – HTTP Server

We later decided to execute an Apache webserver : we wanted to have a database to store the credentials, but SimpleHTTPServer didn't offer that functionality.
We used PHPMyAdmin for the database.

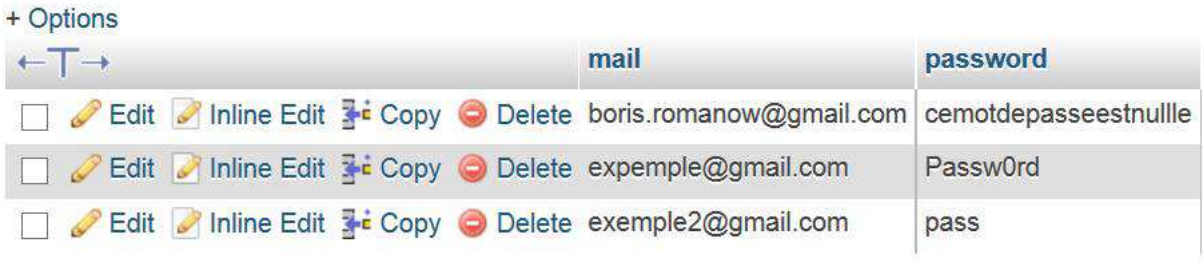

FIGURE 11 – PHPMyAdmin database

1.2 What we did to legally test security flaw

- Work on a personnal WLAN

- Every test was ran on our personnal machines

1.3 Counter measures to prevent this kind of attack

For the website owner :

- not providing an HTTP version of his website

- acquire a legitimate HTTPS certificate

For the internet user :

- always browse on a secured network you trust (ie not on a public WI-FI)

- always browse on a HTTPS website with trusted and verified certificate

2 Encountered problems

2.1 What they were

- HTTPS protocol, while in the MITM, we can 'read' everything as long as the target is on HTTP. If we ask for HTTPS website, all the packets are protected.

- HTTPS with SSLStrip : Even with ssltrip, we still have problems to get credentials on specific websites (facebook.com and gmail.com for exemple). Those website, do not use HTTP at all, so sslstrip does not have eny effects.

2.2 How we actually dealt with them

- HTTPS protocol, the solution to get readable informations with this protocol is to use SSL-Strip.

- HTTPS with SSLStrip : we can't use it on facebook.com or gmail.com, we had to find an other website with less protection. That's why we chose fbi.gov, just to demonstrate our handling.

3 My Ethercap project

We began to code a sniffer-like program : my ethercap.
You can see the source code on the following link :
https ://github.com/as3nds/phishing-wlan/tree/my-ethercap/my-ethercap

4 Differences with the Overview's Objectives

4.1 What are those differences and why did we not met our objectives ?

- We wanted to get the credentials from gmail, but since this site is well protected against this sort of attack, we can't use it as our target.

- We started to code our own tools sniffer (ARP spoofing, etc. Available on our GitHub).

4.2 What are the possible solutions to meet them.

- Find a website less protected and/or having HTTP and HTTPS protocols.

5 What we wanted to learn with this project and what did we really learn

We wanted to learn how to use some basic tools provided natively with kali linux.

6 How did we share the code

We used Git (and especially GitHub) to share the code. (https ://github.com/as3nds/phishing-wlan)
We also used c9.io to work in a collaborative way.

Conclusion :

This projet let us studied a MITM attack. We worked on how to set up and exploit the attack. At the beggining we wanted to use a fake authentification page on 'gmail.com' to trap a target and retreve his credentials.
During the project we realize that this attack is not possible on 'gmail.com' (and others like 'facebook.com') since they have protection against SSLStrip.
On normal HTTP websites there is no problems to catch eny informations and with some HTTPS site, SSLStrip can get rid of the secured connection.

We used some well-known tools to catch and trap targets but also developped our own websniffer and started to develop much more (see GIT).

Definitions :

AP Acces Point, device necessary to connect client on a wireless network.

MITM Man In The Middle, this attack is based on relaying secretly the communication between a client and the AP.

ARP Table It's a database with all the relation between IP adress and MAC adress.

Used Programs :

nmap Scan the network in order to find a target

arpspoof Allows to usurp the target's gateway

wireshark Network analyzer, shows all the packets flowing on the network including protocols and many informations

dnsspoof Listen and intercept the target's DNS requests in order to apply a fake DNS table.

sslstrip This tool, used in a MITM attack, can change requests protocols from HTTPS to HTTP. Only works with website that have HTTP 'AND' HTTPS connection