**MITM DE-Authentication Using False Handshake Manipulation**

Ammar

(SP18 – BCS – 093)

Faiqe

(SP18 - BCS – 096)

Shehryar

(SP18 – BCS – 109)

Salman

(SP18 – BCS – 106)

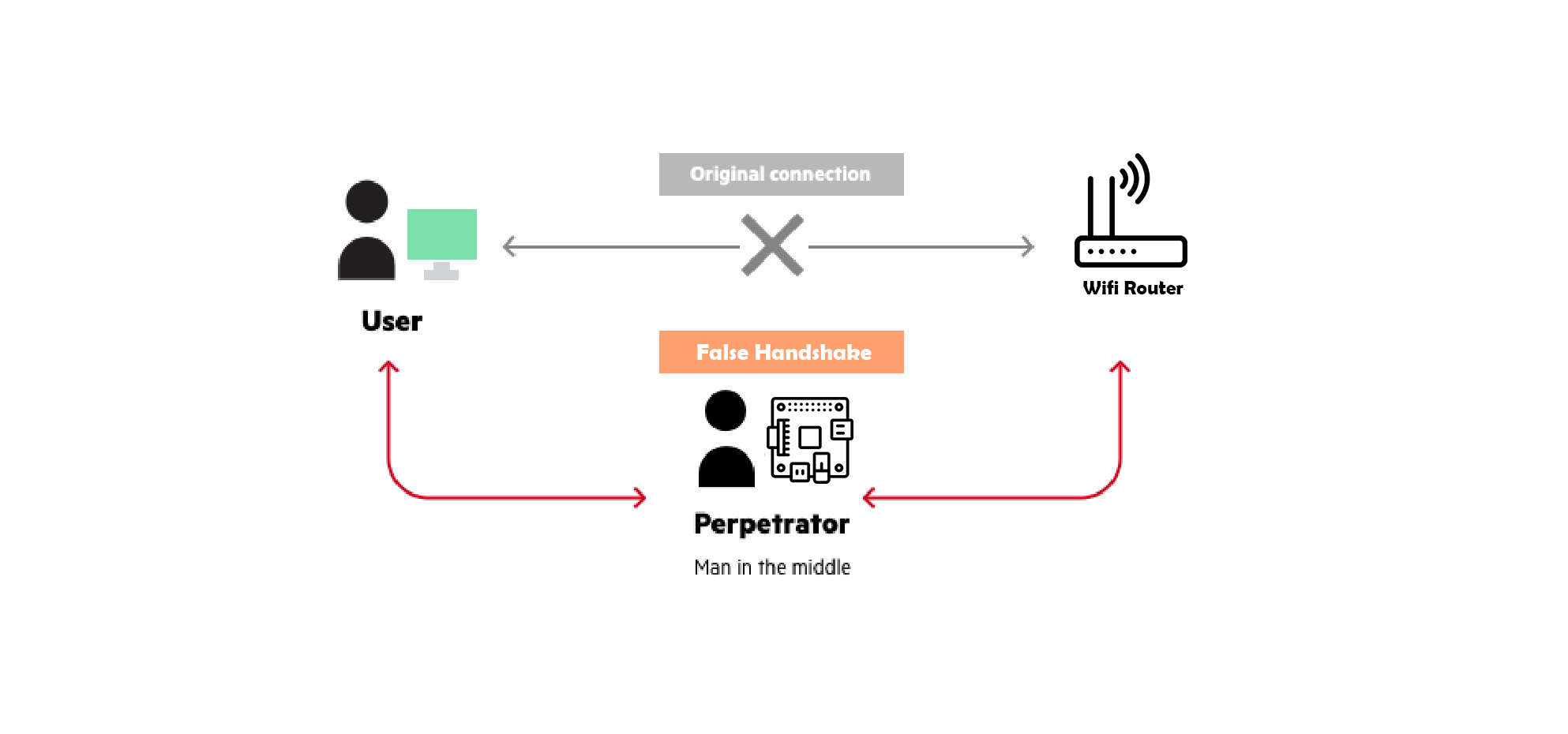
**Submitted to:**

Ma’am Syda Fatima Usman

COMSATS University Islamabad

**MITM DE-Authentication Using False Handshake Manipulation**

**Abstract:**

Hardware used Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz. SSH connectivity is established via 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless (WLAN0) and Monitor mode capable Wi-Fi Antenna for False handshake Authorization (WLAN1). Functionality: Using Linux Bash and Some third-party Dependencies. Man, in the middle disguise attack is pretending to be someone which we are not. Here we are talking about our ARM processor which first interprets the MAC address of a device i.e. connected to the wireless network. While the connection between the device/client and server remains established the exchange of data takes place. But, when the perpetrator establishes the connection towards the client and the wireless access point, the original established connection between the client and the access point is compromised. The perpetrator the communicates with the access point, pretending to be the device which is connected to it, which in reality is the perpetrator. Same happens on the other side, the man in the middle/perpetrator communicates with the device/client showing itself the access point, which in realty is the man in the middle. Some of functionalities MAC address spoofing D-auth hand shake Here is the illustration how the man in the middle attack works

**Hardware:**

The Hardware which is used to process the MITM False handshake Manipulation is in Simplified form is known as Raspberry pi V4, WIFI Monitor, AtTiny85 Digi-spark and A 5V/2A Power supply. Whereas in the programmatic term Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz. SSH connectivity is established via 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless (WLAN0) and Monitor mode capable Wi-Fi Antenna for False handshake Authorization (WLAN1).

**Raspberry pi:**

The RPI (Raspberry Pi) is used for the processing of the script in the Linux Environment.

It is the Core of our module whereas it provides the execution of scripts in a sequential manner. It helps us interact with the outside world, and has been used in a wide array of digital products, here we are using this as a WIFI jammer. Our Raspberry PI uses Debian based Linux operating system therefore we can manipulate our hardware to perform any task we desire. In this scenario we manipulated our Raspberry pi to Monitor every connection packet and de-authenticate using the False handshake.

**WIFI Monitor:**

WIFI Dongles are used for TX, RX Packets but there are special WIFI Modules which can not only connect to a particular access point but can also used to Read the transmitting packets in a Specific or Global channel. These two Modes are called:

* Managed Mode.
* Monitor Mode.

The commands which are used to change the mode of a WIFI module is using a Library known as “Aircrack-ng” or a Debian method known as “iwconfig”

* sudo airmon-ng start wlan1
* sudo airmon-ng stop wlan1mon

Here you can clearly see when our WIFI interface is in managed mode its name is “wlan1” but when it mode is monitor mode its name is appended by “mon” which is to describe that our Interface is now in monitor mode. To check if our interface is in either monitor mode or in managed mode you can check using “ifconfig” or “iwconfig” command both will work

**How to change Interface mode to Monitor Mode using IW config:**

1. sudo ifconfig wlan0 down
2. sudo iwconfig wlan0 mode monitor
3. sudo ifconfig wlan0 up

**Reverse Process:**

1. sudo ifconfig wlan0 down
2. sudo iwconfig wlan0 mode managed
3. sudo ifconfig wlan0 up

[(Full Code is available on GitHub)](https://github.com/AmmarTee/handshake)

**AtTiny85:**

AtTiny85 is an Arduino based microprocessor which in our case is used to automate the execution of the script. Arduino code is available on GitHub the link is in the footer. There are two methods which allow us to pursue the False Handshake attack first is using the Bash scripts which is more of a manual script and second is Using this Digispark AtTiny85 Arduino Based Micro controller the procedure to execute the code is given below.

**Procedure**

**MITM handshake Using Airmon-ng and mdk3 dependencies**

**This is a program to automate the False handshake MITM Protocol**

**How to use:**

1. Install dependencies and packages using install.sh
2. Run master.sh in terminal
3. Enable monitor mode
4. Select Channel to initiate the attack
5. Use CTRL+C to exit and stop the attack

**Using AtTiny85:**

1. Select the channel you desire and upload the code to Digispark AtTiny85.
2. Boot your raspberry pi in CLI/Autologin Mode (Wait for 2 Minutes).
3. Plug AtTiny85 into Raspberry pi.
4. Raspberry Will start the False Handshake in approx. 1 Minute.
5. Unplug Raspberry to stop the execution.

This program uses WIFI interface to replicate the identities of User and Router DE authenticate it and send disassociate packets whenever User tries to reconnect

**PLEASE DO NOT HARM SOMEONE WITH THIS TOOL**

[GITHUB](https://github.com/AmmarTee/handshake)