# Comp 6841 Something Awesome Project

## Network Security

## My Goals

The goal was to increase awareness among University Students and working-class individuals about the danger for students and individuals of the working class who frequenlty use LAN networks at offices and universities which if not correctly configured can lead to a man in the middle exploit where attackers can insert themselves between you and the network not only to observe but also modify packets to redirect you to malicious websites.

My Project aimed to simulate a man-in-the-middle attack through ARP Poisoning and DNS spoofing. I would first corrupt the ARP tables of both my victim machine and the default gateway router of my home network. After successfully performing ARP Poisoning, I would also perform DNS spoofing by analysing packets received as the man-in-the-middle and modifying DNS packets to redirect users to my own hosted webpage.

## What I achieved

I was able to create a Python script (See Appendix D in deliverables) that successfully performed ARP poisoning, allowing my machine running the script to be inserted between the connection of my victim machine and the router. I was also able to inspect and modify DNS packets to perform DNS spoofing; however, I successfully redirected users on the victim machine to my hosted webpage before disabling security measures on the browsers of Linux Machines that I had not considered.

## What I did

Before implementing any scripts, I spent time researching and reviewing Network Security protocols. I had taken COMP 3331 Computer Networks last term and was interested in deepening my understanding through simulating a network attack regarding the ARP and DNS protocols.

I own a Lenovo ThinkPad laptop running Windows 11, and the first major challenge was determining the best setup for running two operating systems simultaneously on my computer. My setup would involve one machine acting as the man-in-the-middle, running my malicious Python script. In contrast, the other machine would be the victim, communicating with my home network's router.

The first iteration involved WSL (Windows Subsystem for Linux), which virtualises a Linux environment on Windows; however, it proved to be inadequate, as it had network limitations that prevented me from sending raw packets. I then switched to a virtual Ubuntu Linux Machine using VirtualBox, which would target my Windows machine. This seemed to work for ARP poisoning.

However, I encountered another roadblock in that I wasn't receiving DNS response packets from the router, which I could then modify and send back to the victim. I spent a couple of hours debugging and logging all packet information, as well as examining the fields of DNS packets that would indicate a response, but I was unable to do so. I then thought that maybe it was due to browser security measures (See Appendix B, Figures 1.1 and 1.2 in Deliverables), which may be preventing this due to encryption or some other measure, and disabled browser security. However, that didn't work either, which led me to suspect it must be a security implementation of Windows 11 at the operating system level. My suspicions were proven correct as I discovered that Windows 11 encrypts DNS packets, which meant I would be unable to modify them. (See Appendix B, Figures 2.1 and 2.2 in Deliverables)

So, in the end, the most optimal setup included two virtual machines, each running a different Linux Distro. In my case, I used Ubuntu and Kali Linux, which allowed me to perform ARP poisoning and DNS spoofing successfully.

One final challenge was that I had difficulty being able to redirect a user on the victim machine to my own ‘maliciously’ hosted webpage. Many Browsers, including Chrome and Firefox, use HTTPS with digital certificates as an added layer of security over HTTP, which means that the browser can detect suspicious activity and prevent users from being redirected. (See Appendix C, Figure 2 in Deliverables)

The primary concern was that my own hosted website did not contain a valid digital certificate, and thus, the browser detected it as untrusted. After obtaining a self-signed certificate, the website displayed a warning advising the user to proceed at their own risk. Many individuals would not have had any technical background to understand the significance of digital certificates and would most presumably accept the risk without any concern. I also examined Ettercap and its ability to simulate a man-in-the-middle attack, comparing it to my solution.

In my project, I am particularly proud of DNS spoofing, which causes users on the victim machine to be redirected to my website, hosted on an Apache2 Web Server. Even though the browser was able to detect suspicious activity, I was pretty proud of having implemented a somewhat successful DNS spoofing technique on top of ARP poisoning.

The aforementioned is evident in what I did for my project, including problems and challenges that I faced and how I solved or found alternative ways to recover from setbacks.

## Reflections and What I learnt

**Overall, the project outcomes turned out to be slightly different from what I envisioned, as I encountered new concepts such as virtualisation and security measures applied to DNS and web browsers.**

This project provided me with a deeper understanding of network security and how attackers can exploit network protocols, such as ARP and DNS, to gain access as a man-in-the-middle.

I learnt a lot about virtualisation and how many machines running different operating systems can be run on one physical machine as virtual machines. Software such as VirtualBox allows resources to be allocated accordingly to all virtual machines running on the physical machine. I also learnt how virtualisation is commonly used in enterprise settings, such as on servers in data centres, which utilise underutilised resources that would otherwise be wasted, with traditional machines running only one operating system.

I learnt about DNS security measures, such as encryption of DNS with HTTPS and/or TLS, which weren't discussed when I took Computer Networks at Uni last term.

Overall, I found that man-in-the-middle attacks are widespread and highly likely due to their low technical barrier. Creating a simple script like mine or using Ettercap are both simple solutions for orchestrating an attack. In conclusion, students need to be aware of the potential risks associated with connecting to local networks that may be configured poorly. To mitigate these risks, it is recommended to use security measures such as VPNs and browsers that implement DNS and HTTPS security correctly.

What I would do differently next time is to further research methods for workarounds to effectively simulate a man-in-the-middle attack for DNS packets encrypted over HTTPS and/or TLS. As my current solution for ARP poisoning and DNS spoofing is only effective against Linux machines, I would like to know how DNS spoofing can occur with Windows Machines, which provide additional security measures, including encryption.