COMP 4203 Project Proposal

Title: **Lightweight Solutions for Detecting Deauthentication Attacks**
Authors: **Matthew Nitschke, Steven Rhodes**

Research Papers:

1) Kumar & Singh (2019). A Lightweight Solution for Detecting Deauthentication Attack:
https://aircconline.com/ijnsa/V11N1/11119ijnsa02.pdf

2) Mital, Nguyen, Nguyen, Tran (2008). A Lightweight Solution for Defending Against Deauthentication/Disassociation Attacks on 802.11 Networks:
https://www.researchgate.net/publication/221092095_A_Lightweight_Solution_for_Defending_Against_DeauthenticationDisassociation_Attacks_on_80211_Networks

Brief Description:

Comparing two research papers that explore different methodologies of detecting deauthentication attacks in WLAN/802.11 wireless networks.

Description of Domain:

The current project is to be done on a topic in wireless network security. Wireless security is important as it ensures protection to a Wi-Fi network. In terms of wireless security authentication, the papers mainly focus on the deauthentication aspect. Deauthentication is used to gracefully terminate connections between a connected client and access point. One flaw of wireless networks is that they are vulnerable to management frames attacks. This is due to the fact that the management frames are not encrypted nor authenticated. Attacks that fall under this area of management frames attacks include deauthentication/disassociation frame attacks.

Description of the Research Question:

Our research question is to find a way to detect deauthentication attacks made towards a connected client and access point. Deauthentication attacks are a type of attack that targets communication between a user and wireless access point and then denies the user of service. This denial of service is done when an attacker spoofs the deauthentication and disassociation frames acting as if the client has left the server. Thereby disconnecting the client from the access point. Since these frames are unencrypted and unauthenticated this can be done pretty easily by an attacker. The goal of the paper is to identify if there is a deauthentication attack on the client or not.

Outline of Development Technology:

There are going to be 3 main things we need to do to reimplement the work required in the paper *A LIGHT WEIGHT SOLUTION FOR DETECTING DE-AUTHENTICATION ATTACK* (Paper #1). The first thing we need to do is create a client and server to interact with each other. Secondly we need a way to monitor the messages that are being sent back between client and server. With the final step being a way to create deauthentication attacks against the client and server.

List of Potential Technology:

- Languages: Python (for client and server, and simulation deauthentication attack)
- Operating System: Windows or Linux; an attempt at using Kali Linux will be considered, as the paper uses this distribution
- Libraries: PyShark (monitoring wireless packets)
- Other: Wireshark, tcpdump for network monitoring
- Other technologies will be considered as problems arise over the course of the project

We have both used wireshark during the course of Principles of Computer Network (COMP 3203). Matthew and Steven both used tcpdump last semester during the course Computer Systems Security (COMP 4108). Matthew's python experience includes courses such as; COMP1405, COMP3109 and COMP 3203. Steven's python experience includes; COMP 1405, COMP 3203 and COMP 4201.

Outline of Study Methodology:

We will be following the methodology in the above mentioned research paper and developing an application that monitors all the wireless traffic data between a client and server. The client and server will be created using the language Python. The purpose of the client and server is to simulate the client application and the access point interaction with each other over a wireless network. If the application detects deauthentication packets, it will then send out alerts of a possible deauthentication attack against the client. For monitoring the wireless packets Python's PyShark library will be used. In addition, wireshark and tcpdump will be used to monitor network traffic between the client and server. The reason for using wireshark and tcpdump is to help monitor the network traffic. We will also develop a Python application to simulate a deauthentication attack to verify that our application for detecting deauthentication attacks is working properly.

The following is the base outline for detecting a deauthentication attack against a client. First would be to check for management frames. The next step taking place would be checking for subtype of the management frames. If the management frame is a deauthentication frame then extract mactime. If the mactimes are constant then check their reason for exit status. If their reason for exit status does not change or there are data frames after deauthentication then send an alert. If the mactimes are not constant then check if there are data frames after deauthentication, if so then send an alert. If there is no alert sent then there is no evidence of a deauthentication attack.

```
check if management frames
check subtype
if(subtype == deauth) then
extractmactime
if(mactime[intervals] is constant) then
        check reason_code
        if(reason_code == same)
                alert()
        else if ( data_frames_after_deauth )
                alert()
        end if
else if ( data_frames_after_deauth )
        alert()
end if
else
return
end if
```

Figure 1: Sample Algorithm for Deauthentication Detection

Once we have developed our simulation for the first paper (see research paper #1 above), we will then make a comparison with paper #2; specifically, a comparison will be made between which methodology achieves the answer to the research question (detecting deauthentication attacks), and which is more accessible for usage. A report will be written comparing the 2 papers and a recommendation of which method should be used will be made.