NTNU

Norwegian University of
Science and Technology

# Session hijacking in WLAN based public networks

**Ørjan Bækkelund**

Master of Science in Communication Technology
Submission date: June 2009
Supervisor: Stig Frode Mjølsnes, ITEM
Co-supervisor: Martin Eian, ITEM
Thomas Jelle, Trådløse Trondheim

Norwegian University of Science and Technology
Department of Telematics

# Problem Description

A challenge now and in the future in all publicly available wireless networks is the threat of session hijacking. Incidents of abuse and illegal activities have occurred where the wrong person have been held accountable. How can technical evidence of session hijacking be gathered so it can avoided that an innocent user is accused?

The thesis' analysis should be based on the existing structure and access mechanisms that exists today in Wireless Trondheim. The task is to find attack methods that work, and then suggest and experiment with mechanisms that can provide technical evidence for unauthorized access.

Assignment given: 15. January 2009
Supervisor: Stig Frode Mjølsnes, ITEM

# Abstract

The background for this masters thesis is the threat of session hijacking in public wireless networks. A public wireless network in this context is a network such as Wireless Trondheim where users with WLAN enabled devices can connect for a small fee for a given period of time. These kind of networks relies on having a high degree of user friendliness to reach users with average knowledge in computers and wireless networks.

There is always a struggle between user friendliness and security and the downside to the user friendliness in these kind of networks is poor security. Many of these networks only use the unique identifier (MAC address) of the network device to identify users and grant them access. A person with some technical knowledge about wireless networks and less then honest intentions may exploit this weak security barrier and impersonate the legitimate user by duplicating the MAC address.

The practical part of this master thesis starts with the setup of a test bench with three computers, an attacker, a legitimate client and a passive monitor. A MAC spoofing attack was performed on the production network to prove that this kind of attack is easy to perform. The attack was first done with Backtrack which is a specialized penetration testing OS and the same type of attack was done in Windows to also prove that it does not require specialized tools. The attacker was able to gain access to the Internet without going through the web page for authentication.

The thesis also proposes some countermeasures against this kind of attack. They are session ID, MAC sequence number tracking and monitoring physical properties such as received signal strength and RTS-CTS handshake round trip times. The thesis presents some thoughts on how they can be implemented in the wireless Trondheim network and what the major difficulties of each of them might be. The thesis also makes an evaluation

of how well each of them fit with Wireless Trondheims requirements for countermeasures against the attacks done in this thesis.

# Preface

This thesis was written as the final part of a master's degree in Communications technology with specialization in information security. It was carried out at the Department of Telematics (ITEM) at the Norwegian University of Science and Technology (NTNU) during spring 2009.

During november and december 2008 the choice of of subject for the master's thesis had to be made. The subject about security in public wireless networks seemed interesting and was part of my list of prioritized subjects. This was the subject I was appointed and in January 2009 I met with my main supervisor Thomas Jelle and my co-supervisor Martin Eian to work on a problem description. It had initially a broad focus on general security threats against public wireless networks, but after input from the professor responisble for the thesis Stig F. Mjølsnes it was refined to only cover session hijacking attacks.

The work on this thesis has been interesting and educational and I have gained useful insight in wireless networks and security surrounding them.

# Acknowledgments

Many people have contributed with insight and advice during the work on this master's thesis.

First of all I would like to thank my co-supervisor Martin Eian for taking the time for scheduled weekly meetings during the work on this thesis, and thanks for useful input on important things to cover in the thesis.

I would also like to thank Thomas Jelle and the rest of Wireless Trondheim for input on their wishes for the master thesis, and my responsible professor Stig F. Mjølsnes for helping to define a more specific problem description for my thesis.

And last, but not least friends and fellow students at NTNU and Brage Rønning Tukkensæter and Jens Wiel Monrad-Hansen for the fun working environment at the Wireless Trondheim office.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| WEP | Wired Equivalent Privacy |
| WPA | Wi-fi Protected Access |
| LAN | Local Area Network |
| SSID | Service set identifier |
| VLAN | Virtual local area network |
| WLAN | Wireless Local Area Network |
| RSN | Robust Security Network |
| CUWN | Cisco Unified Wireless Network |
| WLC | Wireless LAN Controller |
| IDS | Intrusion detection system |
| RSS | Received signal strength |
| RTS | Request to send |
| CTS | Clear to send |
| RTT | Round trip time |
| BT | Backtrack |
| ROM | Read only memory |
| OS | Operating system |
| AP | Access point |
| SP2 | Service pack 2 |
| ACK | Acknowledgment |
| HTTP | Hyper Text Transfer Protocol |

# Chapter 1

# Introduction

## 1.1 Background for the thesis

In the later years the use of wireless networks has become more and more widespread because of the flexibility and ease of connection they offer. Security in wireless networks have some additional challenges compared to wired networks. This is due to the fact that the traffic is transferred as radio waves in the air and anyone close enough with an antenna can receive them.

Wireless Trondheim is a provider of Internet access through wireless LAN in the city center of Trondheim. Anyone with a laptop or other WLAN enabled device can connect to Wireless Trondheim and after connecting the users will be redirected to a web-page with several options to gain access to the Internet. A public wireless network such as Wireless Trondheim is dependent on having a high degree of user friendliness to attract users with average knowledge about computers and computer networks. The problem with this is that there is always a compromise between security and user friendliness.

As a consequence of its user friendliness the access solution used by most of the users of Wireless Trondheim has security flaws that are easily exploitable by a person with some technical knowledge about wireless networks. The traffic is unencrypted and is vulnerable to eavesdropping and with an unencrypted network users are also vulnerable to a session hijacking attack.

When an attacker performs a session hijacking attack he fools the network into believing that the attacker is the legitimate user. From Wireless

Trondheims viewpoint there is up to date no way to tell the attacker and the client apart. The concern of Wireless Trondheim is that an attacker can use this for criminal purposes and that an investigation of those activities will point to the legitimate user.

Wireless Trondheim would therefore like to have a system that can detect the presence of a session hijacking attack. Such a system could prevent innocent users from being accused of criminal activity if they have been victim to a session hijacking attack. Wireless Trondheim does not, however, wish to reduce its current level of user friendliness. Such a system could also have the ability to throw users off the network immediately when session hijacking is discovered, but Wireless Trondheim does not wish to do this. The system should make logs of the registered activity that can be analyzed for suspicious activity at a later point in time.

Wireless Trondheim also offers access to eduroam which is built on more secure technology then the more popular network. This network is only available to students or staff associated with NTNU and other academic institutions abroad and requires a lot more configuration to use. This network will not be in focus in this thesis.

## 1.2 Thesis goals

The first goal in this thesis should be to find practical session hijacking attacks that can be executed in experiments. The experiments should then be conducted to prove the statement that such attacks are easy to perform.

After the attacks have been successfully performed research should be conducted to find countermeasures that can detect the presence of such attacks. These countermeasures should be evaluated based on Wireless Trondheims wishes. These wishes are no decrease in user friendliness and the ability to make logs that can be analyzed at a later time to uncover suspicious activity.

## 1.3 Thesis outline

The following is a list of the chapters in the thesis and a short summary of what they contain.

**Chapter 1 - Introduction**

This introduction chapter

**Chapter 2 - Theory**

Basic theory behind the technologies this thesis builds on, such as 802.11, Intrusion detection systems, MAC addresses and other things.

**Chapter 3 - Penetration testing**

This chapter describes the practical penetration testing attacks that were conducted in this thesis.

**Chapter 4 - Countermeasures**

Describes the countermeasures against session hijacking that were discovered. It also contains some thoughts on how they can be implemented into the Wireless Trondheim network and the difficulties that might occur.

**Chapter 5 - Results**

A quick summary of the results from this thesis

**Chapter 6 - Discussion**

Discusses some aspect of an IDS in Wireless Trondheim and makes recommendations on how well the proposed countermeasures fit with Wireless Trondheims wishes.

**Chapter 7 - Conclusion and future work**

A summary of the results from chapter 6 and future work that can be based on the results of this thesis.

# Chapter 2

# Theory

## 2.1 Introduction to Wireless Trondheim

The goal of Wireless Trondheim is to supply wireless Internet access in Trondheim and to make the city a laboratory for development of wireless and mobile solutions. [7] Wireless Trondheim is a cooperation between NTNU and a few local government branches and businesses.

### 2.1.1 Access methods in Wireless Trondheim

When writing this thesis there are two main methods of gaining access to Wireless Trondheim. The less secure portal solution and the more secure WPA2 solution.

#### 2.1.1.1 Portal

The wireless network with SSID tradlosetrondheim in downtown Trondheim is the SSID for the portal solution. It is encryption free and anyone can connect. However if an unauthenticated user makes a web request to any other web page then a small number of white listed urls he will be redirected to the web-portal where the user is presented with a few authentication options. Among these are the option to pay 10 NOK by SMS for 3 hours access and to log in with your NTNU user name and password. This initial authentication on the web-portal is done over a secure SSL transaction, but after the authentication the traffic between STA and AP is unencrypted. The Nomadix gateway uses MAC filtering to allow authenticated users to be routed to the Internet.

#### 2.1.1.2 WPA2

Wireless Trondheim also supports a more secure connection for users that have access to Eduroam. Eduroam is a roaming infrastructure in use by research and academic communities. It allows its users to connect to WLANs at each others institutions across Europe and Australia. [5] NTNU is a part of eduroam so it is available through the wireless network on the campuses of Gløshaugen and Dragvoll. Wireless Trondheim also offers access to eduroam through their network.

The security mechanisms in use for eduroam are 802.11i compliant. The security is therefore considered as very good by todays standards.

### 2.1.2 Wireless Trondheim architecture

Building and configuring large wireless networks can be a difficult challenge. A large network consists of many access points and if all of these had to be configured manually one by one large wireless networks would have serious scalability issues. That's why centralization of configuration and access control is the way to go.

Wireless Trondheim is built on the Unified Wireless Network architecture from Cisco [4]. The Cisco Unified Wireless Network uses both wired and wireless components to tackle the challenges in creating a large scalable wireless network. Cisco can deliver all parts of an enterprise wireless network from client devices to network management devices. In the description of CUWN, Cisco states that their solution cover all the 5 areas they have defined that a fully functional enterprise wireless network should consist of. These 5 areas are:

- Client devices
- Access points
- Network unification
- Network management
- Mobility services

Wireless Trondheim has no need to cover all these areas and has taken a couple of these to create a network that satisfies its needs.

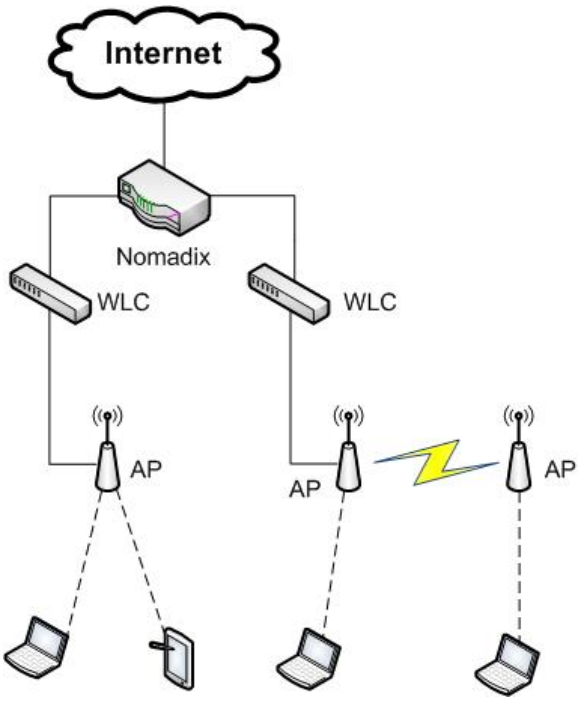


Figure 2.1: Simple overview of the portal solution network in Wireless Trondheim

#### 2.1.2.1 Lightweight access point

Figure 2.1 gives a general overview of the network architecture in wireless Trondheim, but note that it is only accurate for the portal solution. The main benefit this architecture has is the use of Lightweight access points. In a simple wireless home network the access point is in reality a multifunction device. It handles the wireless channel, authentication of clients and routing of IP traffic. All the lightweight access point does is to send and receive wireless link layer frames between the client stations and Wireless LAN Controllers (WLC). Most access points have wired connections to the rest of the network, but they can also be connected to each other via a wireless link in case wiring is difficult to perform.

#### 2.1.2.2 WLC

The WLCs centralize the configuration of the wireless network and makes the configuration a lot easier in large networks. The access points are configured automatically by a WLC when it is connected to the network. That

removes the need to configure each access point manually one by one.

The WLCs delivered by Cisco supports several SSIDs and on the wired part of the network traffic from different SSIDs are kept apart by using VLANs. When each SSID has its own VLAN it is easy to e.g. deploy different security mechanisms to different wireless networks which is what Wireless Trondheim has done with Tradlosetrondheim and eduroam. To a user they are two different networks, but they are handled by the same infrastructure.

#### 2.1.2.3 Nomadix gateway

Anyone can connect to the SSID TradloseTrondheim in downtown Trondheim, but surfing on the web is restricted to a small number of white listed sites. The nomadix gateway keeps a list of MAC addresses for the authenticated users. When a user with a non listed MAC address connects and makes a request for a web page that is not white listed the nomadix gateway performs some http trickery to make the users browser display the Wireless Trondheim authentication portal. Nomadix is a common solution to achieve this kind of functionality in public access wireless networks. Nomadix is not a product of Cisco, but it can be integrated into their unified network solutions.

## 2.2 Basics of the wireless MAC layer

This section will briefly explain some of the basics of the 802.11 MAC layer to prepare the reader for the following sections which will use many of the basics concepts.

The widely used 802.11 standard operates on layer 2 and layer 1 in the TCP/IP model 2.2. The purpose of layer 2 is to ensure error free data transfer between two network interfaces often on a shared medium. The purpose of layer 1 is to define the way bits are transformed into physical signals that can be transferred over a medium. In 802.11s case the data transfer is being done via radio waves and the shared medium is the air. Using the air as a shared medium creates some challenges that 802.11 has to take care of to ensure the wireless clients can communicate with as little interference as possible, but that is not in the scope of this thesis.

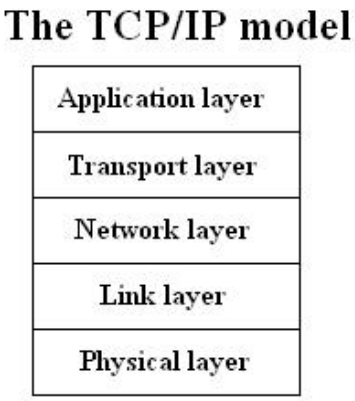


Figure 2.2: The 5 layer TCP/IP model

The data that a wireless client wants to send is separated into frames and a MAC header 2.3 is added to each frame. This header contains important information about the data transfer where the sender and receiver addresses are the two most important fields.

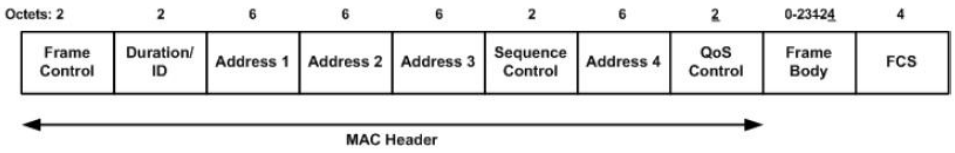


Figure 2.3: The MAC header fields, from [2]

The addresses are 48-bits or 6 bytes long and are known as the MAC addresses. 48 bits is a long string of 1s and 0s so a MAC address is most often written as 6 bytes in hexadecimal form to make it more human readable. E.g. **00:11:22:33:AA:BB**. Each network interface device has its own MAC address which is stored in the hardware. The MAC address is in principle supposed to be unique for every network device in the world and manufacturers makes sure that each of their network interface devices are shipped with a unique address. The problem is that it is easy to change this address most often by bypassing the hard coded MAC address in the network device via functions in the computers OS. This opens for MAC spoofing attacks which are the basis of this thesis.

### 2.2.1 MAC spoofing

MAC spoofing basically means to change the MAC address your network card uses to identify itself on the network. By changing the MAC address an attacker can pose as another user or conceal his own MAC address on a wireless or wired network. In Windows this can be done by editing a value in the registry that contains the MAC address. After changing the value the wireless interface just needs to be restarted and it will start using the new MAC address on the network.

In Linux it is even easier. In Ubuntu a program named macchanger is available through the package manager and this program will change the MAC address of an interface with 3 simple commands.

At the time of writing this thesis Wireless Trondheim has no means of detecting the presence of attacks where the attacker changes his MAC address to impersonate a user.

## 2.3 TCP connections

TCP is located in layer 4 of the TCP/IP model 2.2. Layer 2 offers data transfer between two nodes on a link, layer 3 offers data transfer between two arbitrary nodes in a network and layer 4 deals with flow control and connections between nodes. A basic knowledge of how TCP connections are set up is needed at a later point in this thesis so a quick explanation follows.

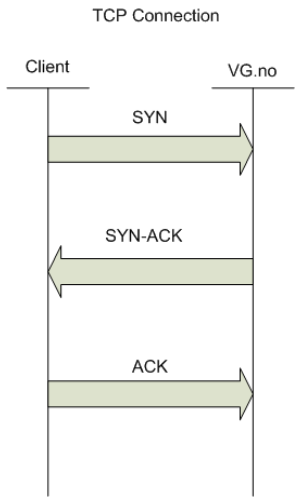


Figure 2.4: TCP connection sequence diagram

Lets say a user is browsing the web and wants to check vg.no which is the web edition of Norway's largest newspaper. The users computer first needs to set up a TCP connection to the vg.no webserver to be able to request data. The first step is that the users computer sends a SYN message shown in figure 2.4 to indicate to the vg.no webserver that it wishes to initiate a connection. The webserver responds with a SYN-ACK to indicate it is ready for the connection. Then the users computer responds with an ACK to finalize the connection setup. After the connection is successfully set up the users browser can request data from the vg.no webserver.

## 2.4 Previous work

The following is a short summary of the articles this thesis is based on. Their main focus is countermeasures against MAC spoofing attacks in wireless networks.

### 2.4.1 Article on MAC spoofing types and countermeasures

The article "Detecting and Blocking Unauthorized Access in Wi-Fi Networks" [10] is written by Xia and Brustolini at the University of Pittsburgh. This article separates MAC spoofing into three approaches an attacker might choose. These approaches are hijacking where the attacker actively throws the victim off the network, freeloading where the attacker uses the network simultaneously with the victim and waiting for availability where the attacker waits until the user is not using the network. After describing these variations on MAC spoofing attacks the article describes some countermeasures and they also do practical experiments with the countermeasures.

### 2.4.2 Articles from QUT

The articles "Passive Techniques for Detecting Session Hijacking Attacks in IEEE 802.11 Wireless Networks" [8] and "Experiences in Passively Detecting Session Hijacking Attacks in IEEE 802.11 Networks" [9] are written by Gill, Smith, Looi and Clark from Queensland University of Technology. These articles take a different approach then the previous one in detecting MAC spoofing attacks.

They are looking at physical properties in the radio transmission to

detect if there are more then one client using the same MAC address. These two properties are the received signal strength and the round trip time for a specific type of message. By correlating between these two methods they get promising experimental results in [9].

## 2.5 Attacks

### 2.5.1 MAC spoofing

This attack is aimed at the portal solution in Wireless Trondheim and it is in theory simple. When connecting with the portal solution in Wireless Trondheim a user must first go through the authentication procedure on the web-portal. The authentication procedure is done via an SSL protected web-page and cannot be listened to by an eavesdropper. After the authentication is completed the authenticated user is white listed in a MAC filter maintained by the Nomadix gateway and its web requests are routed to the Internet. The theory is that the attacker can spoof the MAC address of a connected user to gain access.

The attackers access will only last until the legitimate users white listing in the Nomadix MAC filter times out or the user uses the log off function after he is finished using the network.

### 2.5.2 Variations of MAC spoofing

There are some variations on how this type of attack can be performed as explained in [10]. The variation is on how the attacker treats the already existing client with the MAC address that the attacker is spoofing. It is difficult to detect all the variations using a single detection method so effective intrusion detection against all of them is dependent on a combination of methods.

#### 2.5.2.1 Session hijacking

In this variation the attacker listens to the target network for a client MAC address and the APs MAC address. The attacker uses the APs MAC address to send fake deauthenticate messages to the client and the client will then terminate its association with the AP. The attacker can then spoof the clients MAC and have the session for himself. A problem with this method is that

normally the client will try to reconnect quite often and the deauthenticate procedure will have to be repeated frequently.

Another problem is how an attacker will be able to send deauthenticate frames at the same time as he is using the network. To be able to use the network normally the network interface would need to be in managed mode, but when sending fake traffic the network interface must be in monitor mode. One solution could be to have one interface for each task, but then another problem arises.

The attacker has spoofed the MAC address of the user to be able to gain access to the network and the fake deauthenticate frames have the AP as sender and the client MAC as the receiver. With this setup the same deauthenticate frames that is intended to keep the legitimate client off the network will also cause the attacker to be thrown off. The attacker would have to modify his protocol stack to ignore deauthenticate frames, but this might have unfortunate side effects since deauthenticate frames have other legitimate purposes.

#### 2.5.2.2 Freeloading

In the freeloading variant the attacker assumes the same MAC address and IP as the attacker. The difference from the hijacking attack is that the client is not thrown off the network, but the attacker communicates simultaneously with the client. On the MAC layer this works fine, but when using TCP problems may occur in the transport layer.

When setting up a TCP connection the initiator sends a SYN message to the target system which responds with a SYN-ACK message as explained in section 2.3. The attacker and client are using the same MAC address so both will receive the SYN-ACK response, but only one of them has initiated that connection. The standard TCP procedure is to send a TCP-RST (reset) message when receiving a SYN-ACK for an unknown connection. The TCP-RST message then terminates the connection. This means that a TCP connection initiated by either the client or attacker will be terminated by the other.

Most systems however have some sort of firewall, especially since Windows XP with service pack 2 or above and Windows Vista has one integrated into the OS. If this firewall is configured to ignore traffic from unknown connections the TCP-RST message will not be sent. The attacker can then set

up his own firewall for this purpose and hope that the victim has a firewall that will allow his own traffic to stay uninterrupted.

#### 2.5.2.3 Waiting for availability

If an attacker don't want to risk detection by actively faking traffic to throw the client off the network or to communicate simultaneously with the client the attacker can simply wait until the client is no longer using its session. A users session in Wireless Trondheim lasts from 3 to 24 hours. If an authenticated user browses to www.tradlosetrondheim.no there will be a button to log off the network which will remove the user from the Nomadix white list. If a user uses this function the wait for availability approach will not work, but many user are probably unaware of this function or does not remember to use it.

## 2.6 Countermeasures

### 2.6.1 Basics on Intrusion detection systems

Network intrusion detection systems are systems designed to detect attacks against a network or system in a network. The IDS can be either passive or reactive. A passive IDS will only log suspicious activity and make alarms for an operator to evaluate. A reactive IDS will take action against an attack and can reset connections or reprogram a firewall to block the harmful traffic. A reactive IDS is also called an Intrusion prevention system (IPS).

Network intrusion detection can like everything else not be perfect and errors occur when trying to separate normal and harmful behavior in a network. The ideal IDS will always detect all kinds of attacks and never flag normal behavior as suspicious. A real world IDS can only try to come as close to this goal as possible.

#### 2.6.1.1 False positive and false negative

There are two important terms in relation to IDSs that will be used in the following sections. The first term false positive is used to describe a situation where the IDS has detected suspicious behavior when there is in reality no harmful activity. The second term false negative is used when an actual attack remains undetected by the IDS.

There are two different method an IDS can use to separate normal behavior from harmful behavior.

#### 2.6.1.2 Statistical anomaly

A statistical anomaly IDS has a baseline for what is considered normal network traffic. It will then compare the current network traffic with the baseline. If the sampled traffic is outside the boundaries for the baseline behavior the IDS will respond.

#### 2.6.1.3 Signature based

Signature based IDSs are configured with patterns that are based on what an attacker will do to attack the protected network. Since attackers will adapt their attacks in new ways to avoid the current signatures the signature based IDS is in need of constant upgrades to detect new attacks.

### 2.6.2 MAC spoofing countermeasures

The two following countermeasures are described in [10] and they are aimed at detecting the various types of MAC spoofing attacks covered earlier.

#### 2.6.2.1 Session ID

When a user logs into a public wireless network it is common to give the user a session management page in a small pop-up window. The session ID countermeasure uses a cookie that is associated with this web page. The cookie contains a cryptographically random session ID. The web page is tagged with an HTTP directive that makes the clients browser periodically request a refresh. For every refresh the client browser also sends the cookie containing the session ID. The web page is secured with SSL to keep the session ID hidden from eavesdropping attackers.

This countermeasure is effective against the session hijacking and waiting for availability types of MAC spoofing described earlier. When the attacker either waits for the real client to stop using its session or actively throws the user off the network the IDS system will detect that it is no longer receiving the refresh requests from the client. The attacker will not be able to fake the random session ID if it has sufficient length and cryptographic strength.

#### 2.6.2.2 MAC frame sequence number

This countermeasure revolves around the sequence numbers in the MAC frame headers [2]. When a wireless station sends layer 2 frames the sequence number is incremented by one for each frame sent. With only one legitimate client on a link the sequence number should be a steadily increasing value.

However if an attacker launches a freeloading attack there will be some anomalies in the sequence number values. An IDS system should be able to detect two distinct counters for MAC frame sequence numbers if the the attacker also follows the standard. This is shown in figure 2.5.

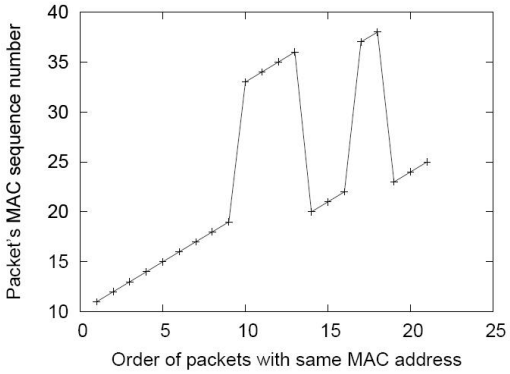


Figure 2.5: Figure from [10] showing two different counters for a set of layer 2 frames

An attacker might decide not to follow the standard for MAC frame sequence numbers. The attacker could try to always send frames with the last sequence number sent by the client incremented by one to try and mimic a normal counter. The client who does not know anything about the attacker will just continue its own counter and this leads to duplicate values within a smaller timeframe then usual. An IDS could be designed to also takes this event into account.

This countermeasure is only effective against the freeloading type of attack. This is because it is dependent on two clients communicating using the same MAC address in the same time frame. In the case of the hijacking and waiting for availability variants there will at best be a jump in the sequence numbers when the attacker takes over the session. After the attacker has taken over its MAC sequence number counter will look like any other.

#### 2.6.2.3 Combining countermeasures

Because the two countermeasures describes earlier only works for separate varieties of MAC spoofing attacks they should be combined. If an IDS uses analyzes network traffic with both methods it can provide a full coverage of the MAC spoofing attacks described in this thesis.

Combining detection methods is generally a useful approach in intrusion detection systems. When two overlapping detection techniques are combined where each of them has a certain percentage for success, the sum of the two will be greater then the two by them self. If two or more detection techniques detects an anomaly at the same time an IDS system can be much more certain that it is real even though the detection methods by them self is not very reliable.

The downside of combining one or more detection methods can be that the IDS might overlook an actual attack because it did not trigger the various detection methods to the degree needed to create an alarm.

In the case of the MAC sequence number analysis and session ID countermeasures this approach can not be used to increase the accuracy of detection. This is because these two countermeasures only works for separate variations of the MAC spoofing attack. An alarm from both these detection methods at the same time will not be possible. If an attacker freeloads, the MAC sequence number analysis will raise an alarm, but the real client will still send responses to the session ID refreshes keeping the session ID checking from detecting anything. If an attacker performs a hijacking attack only one MAC sequence number counter will be present which prevents MAC sequence number analysis from detecting anything, but session ID checking will react to the lack of session ID refreshes.

## 2.7 Other MAC spoofing countermeasures

The following section describes the detection methods from [8] and [9].

### 2.7.1 Received signal strength

The first of the two methods revolves around measuring the signal strength of the frames received by an AP or a wireless sensor. The idea is that when a user is connected to the network his signal strength will be relatively stable

and only vary within a certain range. However if an attacker is using the same MAC address as the user his signal strength will be noticeably different then the clients. Figure 2.6 is an illustration of this. The signal strength can vary for many different reasons such as differences in radio equipment, reflections and refractions in the radio path and the distance between sender and receiver. If the signal strength from frames sent by a single MAC address varies more then normal the IDS can flag this as suspicious and make an alarm.

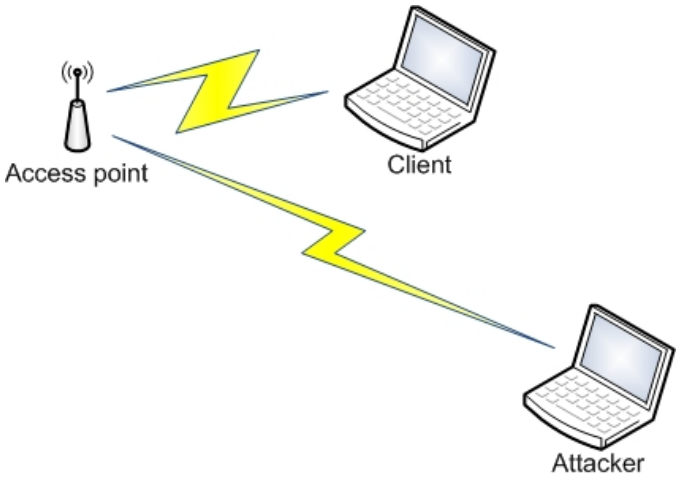


Figure 2.6: Attacker and client with different signal strength seen by the AP

The challenge with this method is that even under normal conditions with only the user connected the signal strength will vary significantly. In conditions where the variation in signal strength under normal operation is bigger then the expected variation generated by an attacker a false positive response will be triggered by the IDS. The tuning of this intrusion detection method to separate normal variation from suspicious variation is challenging.

### 2.7.2 RTS-CTS handshake

The second method revolves around the Request to send - Clear to send (RTS-CTS) mechanism in 802.11. This mechanism is made to avoid collisions between data frames. If two frames are sent at the same time they will interfere with each other and the receiver will not be able to make sense of any of them. This is because of the shared medium properties of the air.

When a node is sending a frame every other node in range can "hear" it and not only the receiver.

This is a simplified explanation of the RTS-CTS mechanism, [8] goes through it in more detail. When a wireless node wants to send data it can send an RTS frame to its intended recipient. When the target node receives the RTS frame it will transmit a CTS frame back to the node sending the RTS. If any other nodes in range receives either the RTS or CTS frame it will halt its transmissions for a given period of time. The time a node needs to halt transmissions is included in both the RTS and CTS frames.

Gill et. al. proposes to use this mechanism for IDS purposes. The RTS and CTS frame have fixed sizes and at a fixed bit rate the only thing that can create variations in the time it takes to complete the RTS-CTS handshake is the radio path between the sender and the receiver. When an attacker has spoofed the MAC address of a client the RTS-CTS handshake will always have a different round trip time for the attacker then for the client, as illustrated in figure 2.7.

If a large enough difference in RTS-CTS RTT is detected by the IDS an alarm can be raised. This method also has some tuning issues since the RTS-CTS RTT can vary even in normal conditions with only a legitimate client.

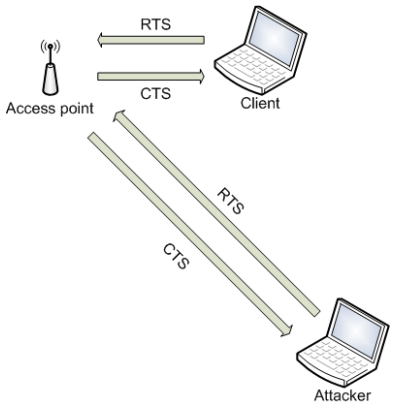


Figure 2.7: Attacker and client with different RTS-CTS times seen by the AP

### 2.7.3 Correlating between the two methods

Gill et. al. uses the principle mentioned earlier about correlating the results from two or more detection methods to make more reliable decisions. In [9] they have made a correlation engine that combines the output of the two detection methods. When one of the methods detects something suspicious the correlation engine checks the result from the other method. An alarm is raised only if both detection methods create an alert at the same time. This way the rate of false positives is decreased.

# Chapter 3

# Penetration testing

This chapter describes the practical part of this thesis which comprises of the test network setup, installation of the necessary software and the MAC spoofing attacks in Backtrack and Windows.

## 3.1 The test network

This section gives an overview of the test network and what the purpose of the components are. The components are summarized in figure 3.1.

### 3.1.1 Wireless Trondheim AP

After an evaluation of the risks to the production network in wireless Trondheim it was decided to perform the attack on the production network itself. The attack were to be performed on a dedicated client computer and none of the other users of Wireless Trondheim would be affected by this. A fully functional access point for Wireless Trondheim was therefore put up in Wireless Trondheims office in room 241 in the IT building at NTNU.

### 3.1.2 Client

This is the computer that will play the role of the innocent user of Wireless Trondheim. The PC is a Dell GX270 desktop, table 3.1 contains its most important specifications.

| CPU | Pentium 4 2.60 GHz |
|---|---|
| RAM | 1.25 GB |
| OS | Windows XP Service pack 3 |
| WLAN interface card | ASUS 802.11g |

Table 3.1: Client computer specifications

### 3.1.3 Attacker

The attacker computer is the authors personal laptop which is an HP pavilion dv9000. A laptop is flexible in where it can be used so it is a natural choice when attacking a WLAN. During the work on the thesis I found out that the network chip set on the laptop was not ideal for work on wireless penetration testing. To start it in monitor mode is easily done by loading a different driver, but traffic injection is harder to do and requires a lengthy process of patching the driver. No traffic injection is needed for the attack in this thesis so it did not become an issue. The most important specifications of the attacker computer is shown in table 3.2

| CPU | Intel Core 2 duo 1.83GHz |
|---|---|
| RAM | 2GB |
| OS | Windows XP |
| WLAN interface card | Intel PRO/Wireless 3945ABG |

Table 3.2: Attacker computer specifications

### 3.1.4 Passive monitor

A 3rd computer is useful to listen to the traffic on the network when the attack is in progress. This computer has a network interface card with the Atheros chip set. This chip set is popular in wireless penetration testing because it is easy to make the card do packet injection.

| CPU | Pentium 4 2.60 GHz |
|---|---|
| RAM | 1.25 GB |
| OS | Ubuntu 8.10 |
| WLAN interface card | Atheros Communications Inc. AR5413 802.11abg |

Table 3.3: Passive monitor computer specifications

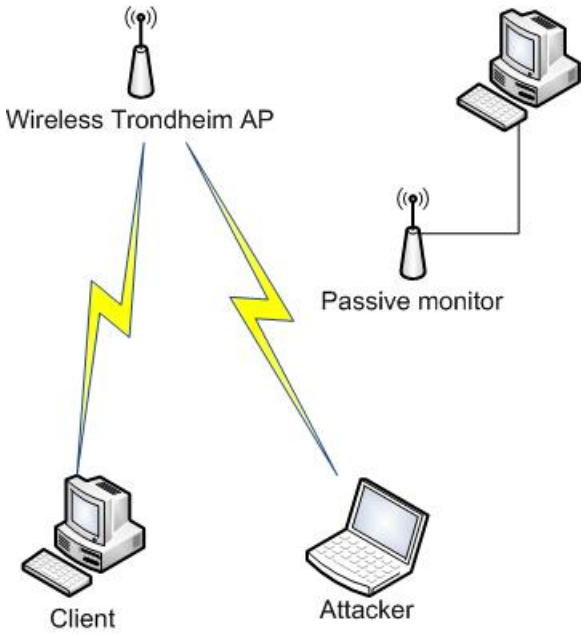


Figure 3.1: Overview of the test network

## 3.2 Tools

### 3.2.1 Backtrack

Backtrack is not a single tool, but an operating system which is focused on network penetration testing. It contains a wide selection of penetration testing oriented tools of which this master thesis only scratches the surface. Backtrack is built on the live CD Linux distribution Slax. A live CD distribution does not require to be installed on a computers hard drive. Instead it can be run from a bootable CD or USB memory stick and does not do any changes to the system it is run on. Slax is built on Slackware which is the oldest Linux distribution that is still maintained. Backtrack is built on Slax and Slax is built on Slackware as illustrated in figure 3.4.

| Backtrack |
|---|
| Slax |
| Slackware |

Table 3.4: Backtrack layers

### 3.2.2 Kismet

Kismet is a wireless traffic analysis tool written by Mike Kershaw. Kismet separates itself from many other wireless network analysis tools in that it is entirely passive. This means it does not send any traffic and can not be detected by other wireless nodes in range. It is an open source program and is licensed under the GNU GPL license [6].

Among its wireless monitoring capabilities are associating wireless clients with the network they are connected to, hidden SSID De cloaking, channel hopping and the possibility of GPS coordinate logging of WLANs.

Kismet also has some IDS functionality. It can detect many active wireless network monitoring programs such as Netstumbler. The active network monitoring programs actively send traffic to trigger responses from the target networks and can be detected by the traffic they send. Kismet can also detect some simple wireless attacks, but Kismet does not detect the kind of attack this thesis revolves around.

### 3.2.3 Aircrack-ng

Aircrack-ng is a suite of many programs written by Thomas d'Otreppe that are used for wireless penetration testing. Aircrack-ng contains programs that can capture wireless traffic, detect wireless APs and clients, crack WEP keys and WPA TKIP pass phrases. Aircrack-ng is free software licensed under GPL. In this thesis Aircrack-ng is only used to capture traffic and dump it to a pcap file which is readable by Wireshark.

### 3.2.4 Wireshark

Wireshark can also capture traffic like the two previous tools, but this is not its most important functionality. Wireshark has a graphical front-end that displays the captured traffic and has powerful functionality to sort and filter data. Wireshark started as Ethereal which was written by Gerald Combs.

Combs needed a tool that could capture and analyze packet data so he wrote the original Ethereal. This protocol analyzer became popular and today more then 500 people have contributed to the Wireshark project. In 2006 Combs changed job and since his old employer owned the name Ethereal the name would have to be changed and the new name is Wireshark.

Wireshark is a tool that "understands" many network protocols and can therefore extract useful information such as header values for various protocols and what type of packet or protocol data is encapsulated in. Wireshark is most often used for network debugging and analysis, network application and protocol development and education.

### 3.2.5 Installing Backtrack

For my masters thesis I chose to install Backtrack (BT) on a USB stick since it is much faster then running it from a CD. BT can also be installed on the hard drive like operating systems usually are. When running Backtrack from a CD it is obvious that no changes will be saved since a CD is a read only memory (ROM). When running Backtrack from a USB stick it has by default the same behavior even though a USB stick is not a ROM. Some small modifications must be made before Backtrack will save changes made to configuration and new files that are made. The steps to install Backtrack and make it save changes is summarized in the list below.

- Prepare partitions on the USB stick
- Download the Backtrack image and unpack it to the USB stick
- Make the USB stick bootable
- Make a folder for changes and modify syslinux.cfg

I used a tutorial on the remote exploit forums which goes through all of these steps [1]. The tutorial is originally for BT2, but as it says in the tutorial it can be easily modified for BT3. When I was going through the steps in the tutorial I found it a bit confusing since I did not understand the purpose of some of the steps. The purpose of this section is therefore to summarize the main steps and explain their purpose. I recommend to read read this part first and then go through the tutorial for the full walkthrough of what to do.

#### 3.2.5.1 Prepare partitions on the USB stick

The first step in the tutorial is to make 4 partitions that will be used for various purposes. These partitions are listed below along with the recommended minimum sizes from the tutorial.

- FAT32 (1-1.5GB)
- SWAP (1GB)
- ext2 (5GB)
- FAT32 or ext2 (Remaining space)

These partitions can be made with the partition manager program qtparted which is standard in Backtrack or available through the aptitude package manager in Ubuntu. Note what reference the third partition has as this is important to get exactly right at a later point. E.g. sda3 or sdb3.

#### 3.2.5.2 Download the Backtrack image and unpack it to the USB stick

The tutorial suggest to unpack the BT image after creating only the first partition. I created all the partitions first and then unpacked BT3 to the first partition. BT is available in an iso image which can be opened in Winrar and then simply unpacked like any other compressed archive.

#### 3.2.5.3 Make the USB stick bootable

Backtrack has now been copied to the USB stick, but when starting a PC with the USB stick inserted Backtrack will not boot because the USB stick is not bootable. This is where bootinst.bat comes in. This file is now on the partition where Backtrack was unpacked and only needs some small modifications before it can be run.

At this point in the tutorial there is a warning written in caps with red color. If bootinst.bat is located on the hard drive of the OS running it it will ruin the boot capabilities of the OS and render the computer useless. How hard it will be to fix this error is not known to me, but the tutorial makes a clear warning against it.

#### 3.2.5.4 Make a folder for changes and modify syslinux.cfg

The last step is to make a folder for changes and then make changes to syslinux.cfg to tell backtrack to use that folder for changes. When a Linux distribution is started it uses a command string that determines some important startup parameters. Syslinux.cfg contains several of these and when the

computer boots from the USB stick a list of different startup alternatives appears. Each of these startup alternatives has its own command string that determines its startup parameters. It is useful to figure out which the startup alternatives that works best after making the USB stick bootable, but before making the modifications to include changes.

After figuring out which alternative works best backtrack can be told to store the changes that the user makes while using the OS. The first step to doing this is to add a folder named changes on the third partition (ext2). The second step is to add an argument to the startup command string of the startup alternative chosen earlier. This is as mentioned located in syslinux.cfg It is important to match the reference to the partition in the argument to what it says when booted in Backtrack, sda3, sdb3 or sdc3 and so on.

The preferred startup alternative can also be moved to the top of syslinux.cfg so it will be selected as default. Timeout can also be set to a lower value so the computer does not wait so long to start Backtrack if the user is absent.

## 3.3 MAC Spoofing

The easiest way to connect to Wireless Trondheim is to use the portal solution. The authentication at the portal is done via an SSL encrypted session so it is not feasible to break this and recover the authentication information. After the authentication however the only access control mechanism is the MAC address filtering at the Nomadix gateway.

This section describes the MAC spoofing attack in detail with screenshots that illustrates how the important data needed to setup the spoofed connection with Wireless Trondheim was found. I used Backtrack to perform the attack the first time, but I realized that it was simple enough to be done in Windows as well. Windows has a much more limited selection of penetration testing tools freely available so this is not the case for more advanced attacks.

### 3.3.1 MAC spoofing in Backtrack

The following is a detailed description of how the MAC spoofing attack was performed using the tools in Backtrack. It should be noted that this

attack can easily be performed in other Linux distributions which are less specialized towards penetration testing. Ubuntu for instance has all the tools used available through the aptitude packet manager. The goal of the attack is that the attacker is able to access the Internet through Wireless Trondheim without authenticating through the web-portal. The attacker should acquire all the information needed to execute the attack by passive monitoring the traffic in Wireless Trondheim.

#### 3.3.1.1 Necessary information

To be able to spoof the MAC address of the wireless client and to set up a connection with Wireless Trondheim the attacker needs some information about the client and the network in Wireless Trondheim. This information is summarized in the list below.

- The channel the AP is on
- The APs MAC address
- Clients MAC address
- Clients IP address
- Subnet mask
- Default gateway
- DNS server address

#### 3.3.1.2 Wireless card in monitor mode

The first thing to do to start listening to traffic is to make the wireless card start in monitor mode. In normal operation when the user is connected to a wireless network a wireless card is in managed mode. In this mode it only passes packets with its own MAC address as destination to the OS. All other traffic to other nodes is discarded.

In monitor mode a network card will pass all the frames it receives regardless of MAC address to the OS which can then be stored and analyzed. After struggling with it I found out that with the laptops network card this could be done with two simple commands.

```
bt ~ # modprobe -r iwl3945
bt ~ # modprobe ipwraw
```

Modprobe is an application that adds or removes modules from the Linux kernel. The first command unloads the driver used under normal use of the network card. The second loads a driver that supports monitor mode.

#### 3.3.1.3 Start sniffing for information

A good place to start is the MAC address of the Wireless Trondheim access point and the channel it uses. Kismet is a good tool to get an overview of nearby wireless networks and to get more info on a specific network. When Kismet is started it will continuously capture traffic and list the networks it finds in real-time. After sorting the list the networks become selectable. Figure 3.2 is a screenshot from Kismet where TradloseTrondheim has been selected and it displays some information for the Wireless Trondheim network.

From this screenshot two of the points from the list of required information can be find. These are:

- The channel the AP is on: **11**
- The APs MAC address: **00:23:5d:0e:01:90**

With this data a more specific capture of traffic can be done where only traffic to and from the wireless Trondheim access point is captured. For this purpose I use airodump-ng which displays clients connected to the APs it finds while it captures traffic. To run airodump-ng I used this command:

```
bt ~ # airodump-ng -w report --bssid 00:23:5d:0e:01:90 --channel 11 wifi0
```

This command contains three parameters. The first one "-w report" is the file airodump should store the captured data in. Airodump will append -01.cap to the string or -02.cap, -03.cap and so forth if airodump has been run before with the same parameter. A file with .cap ending is a capture file that is compatible with wireshark and is the standard capture file format for airodump.

The second parameter "–bssid 00:23:5d:0e:01:90" tells airodump to only capture traffic to and from this MAC address. Without this parameter airodump will capture traffic on other networks as well which is not necessary.

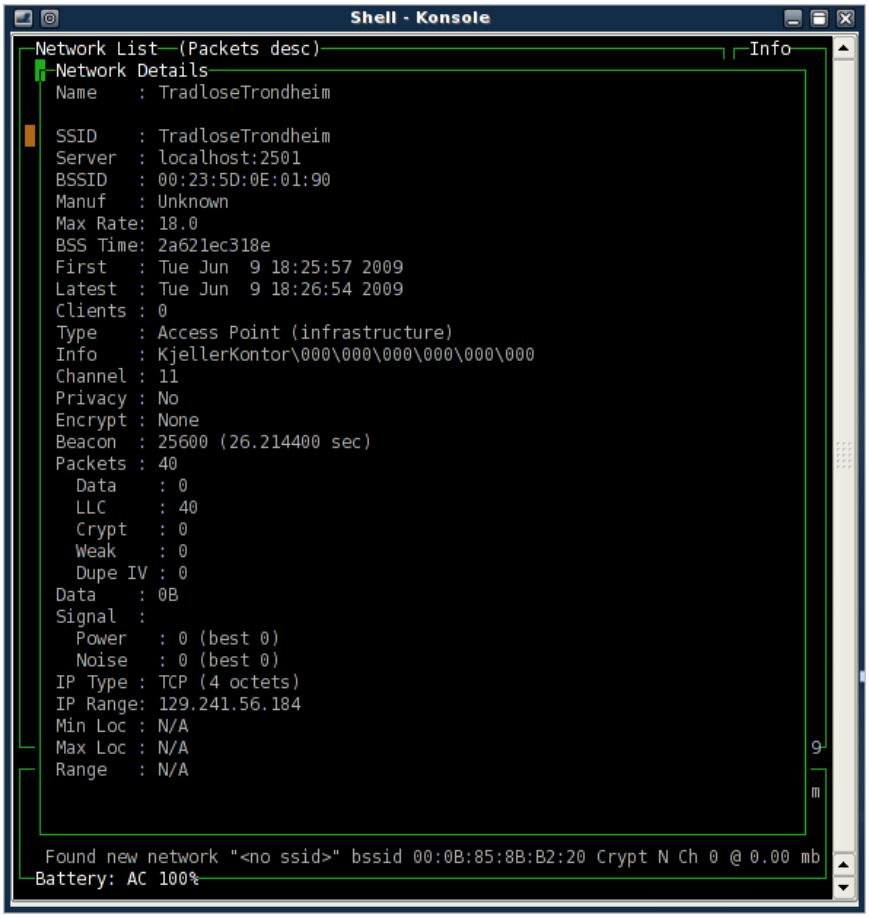


Figure 3.2: Screenshot from Kismet displaying information on the TradloseTrondheim WLAN

The third parameter "–channel 11" tells airodump to only listen on channel 11. When listening to a single AP, tuning in on its channel is very useful. Without this parameter airodump uses channel hopping by default. This means that airodump only listens to each wireless channel for a short amount of time before moving to another channel. This means that much of the traffic to and from the target AP will not be captured while airodump is listening to other channels.

The last part of the command is the interface airodump will capture from.

The screenshot in figure 3.3 shows the clients currently associated with the AP supplied as input. The client listed in this screenshot is the computer that plays the role of the innocent victim. We now have the clients MAC

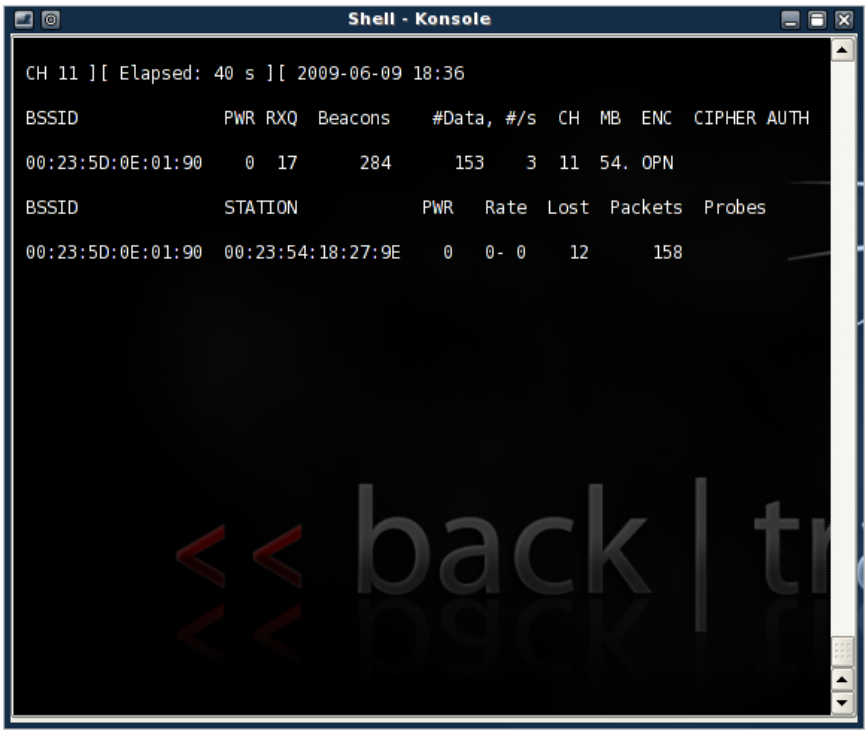


Figure 3.3: Screenshot from airodump displaying traffic on the Wireless Trondheim access point and an associated client

address and can add another point to the list of required information.

- Clients MAC address: **00:23:54:18:37:9E**

The next step is to find the clients IP address. This can be found through the dump file which has now been made by airodump. The program I used to open the file was Wireshark. Wireshark has many advanced filters that can be applied to the packet dump. The filter strings are entered into the text box at the top of the window. I only typed "ip" which filters away all the unnecessary 802.11 packets that does not contain a payload of data.

Figure 3.4 is a screenshot from wireshark that contains all the information needed to establish the IP address of the host that has the known MAC address. When selecting a packet from the upper display window more detailed information appears in the window below. The category IEEE 802.11 Data, Flags: can be expanded and more information on the layer 2 part of the packet can be found. The senders MAC address matches the clients MAC found earlier so the senders IP address is the clients IP address.

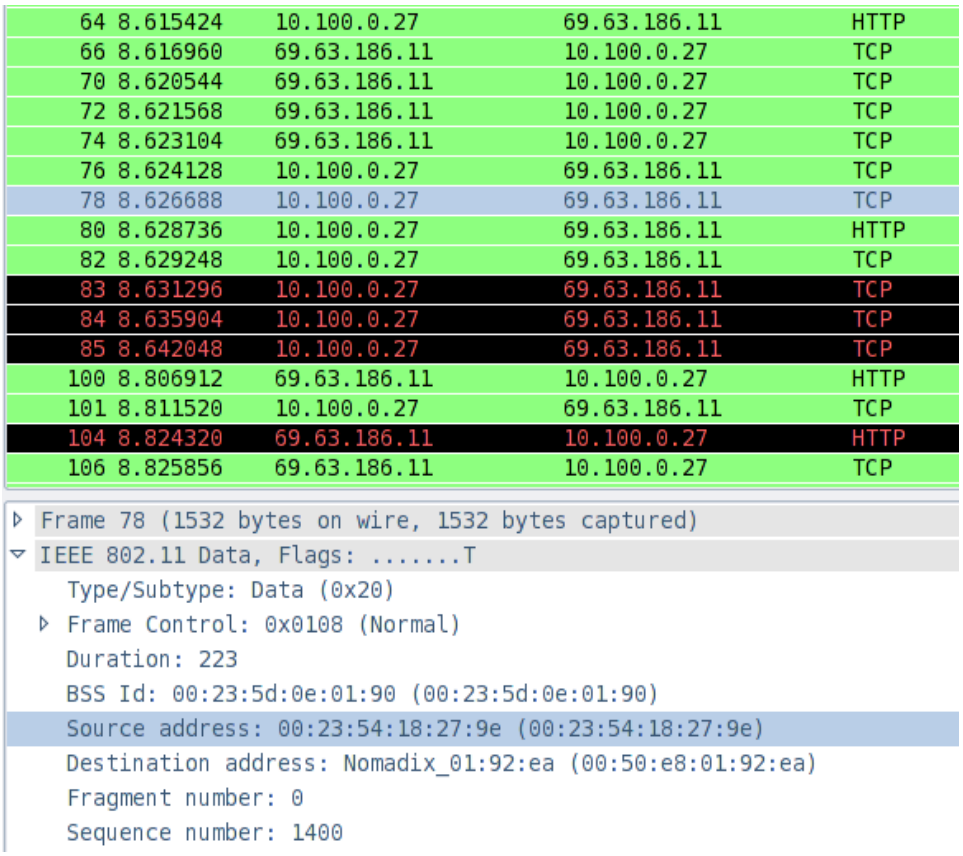


Figure 3.4: Screenshot from Wireshark that shows a captured IP packet and its sender MAC address

- Clients IP address: **10.100.0.27**

4 out of 7 points in the list of required information has now been covered. The three remaining points are parameters for the Wireless Trondheim network that are needed to be able to connect to and use the Internet through Wireless Trondheim. All of these parameters can be found by making a legitimate connection to Wireless Trondheim once. It is sufficient to make a connection without going through authentication to get this information. Figure 3.5 shows the result of entering "ipconfig /all" in a DOS prompt window in Windows.

From here the three last points on the list can be found.

- Subnet mask: **255.255.254.0**
- Default gateway: **10.100.0.1**

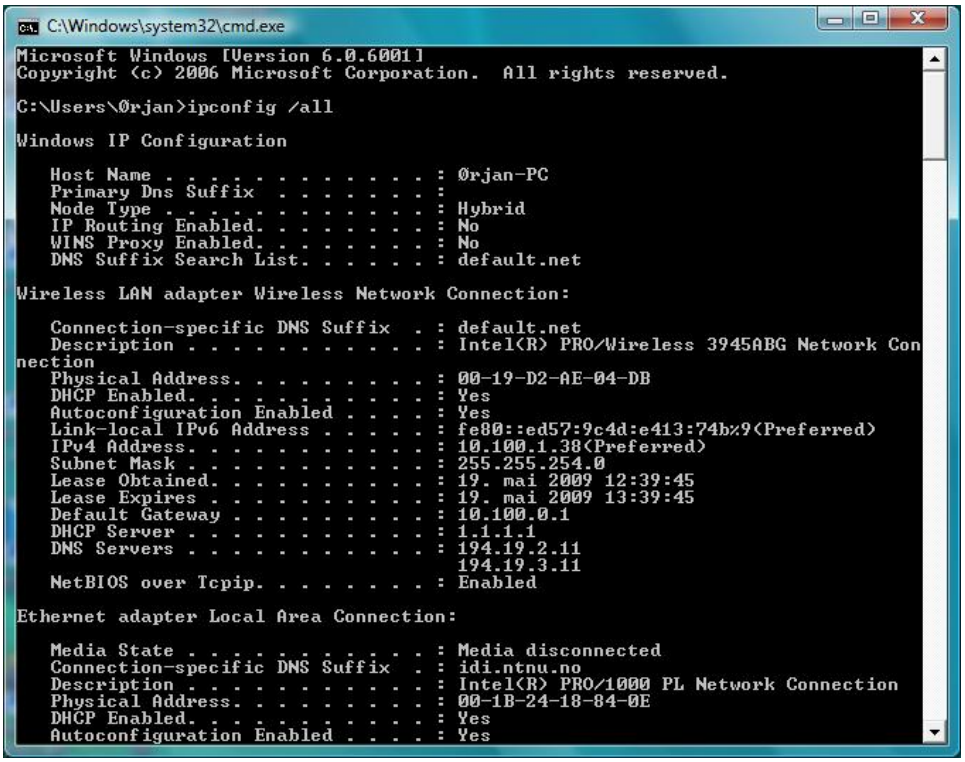


Figure 3.5: Screenshot from ipconfig in Windows with information about the Wireless Trondheim network

- DNS server: **194.19.2.11** and **194.19.3.11**

#### 3.3.1.4 Executing the attack

All the information needed is now known and the actual attack can be performed. The network interface card is currently in monitor mode and has to be switched to managed mode to be able to make a connection to Wireless Trondheim. This is done by reversing the procedure from earlier.

```
bt ~ # modprobe -r ipwraw
bt ~ # modprobe iwl3945
```

The first step is to enter two lines into iptables to avoid the problems with tcp sessions explained in section 2.5.2.2. This is done with the following commands.

```
bt ~ # iptables -A INPUT -i wlan0 -m state --state ESTABLISHED
,RELATED -j ACCEPT
bt ~ # iptables -A IMPUT -i wlan0 -p tcp -j DROP
```

Iptables can be considered as the equivalent to the Windows firewall in Linux. By entering rules into iptables a user can make his own customized firewall. The first command enters a rule into iptables that tells Linux to accept all packets that belongs to currently known TCP sessions. The second command enters a rule that tells Linux to drop all packets from unknown TCP sessions. This way the attackers computer will not send a TCP-RST when receiving a TCP-SYN from a session initiated by the client. This will prevent the user from noticing anything about the attack.

The second step is to spoof the MAC address of the client network card. This is done with the following commands.

```
bt ~ # ifconfig wlan0 down
bt ~ # macchanger --mac=00:23:54:18:27:9e wlan0
bt ~ # ifconfig wlan0 up
```

The last step is to set up the connection itself, which is done with the following commands.

```
bt ~ # /sbin/iwconfig wlan0 mode managed channel 11 key off
essid TradloseTrondheim
bt ~ # /sbin/iwconfig wlan0 ap 00:23:5d:0e:01:90
bt ~ # /sbin/ifconfig wlan0 10.100.0.27 netmask 255.255.254.0 up
bt ~ # route add default gw 10.100.0.1
```

The first two commands tells the wireless interface to associate with the Wireless Trondheim access point. The third command sets a static IP to the wireless interface. Normally this is done via DHCP, but that creates complications with the already existing client so a static IP has to be configured. The last command adds a default route to the Linux routing table.

The only thing left now is to start a browser and hope the targeted client has a firewall enabled on his computer so the client does not interrupt the attackers traffic by sending TCP-RST messages. If the attack is successful the attacker can browse the web like a normal user of Wireless Trondheim

without having to authenticate himself through the web-portal. The free access will only last until the legitimate users MAC address is removed from the Nomadix white list either by time out or when the user uses the log off function.

### 3.3.2 MAC spoofing in Windows

When doing the attack in Backtrack it became clear that there was no need to use a highly specialized OS to perform the simple tasks needed. The same attack can be performed in Windows which is normally a very limited OS when it comes to penetration testing. This section is a walkthrough of how the freeloading type of MAC spoofing can be performed in Windows.

When using the wireless network manager in Backtrack to connect to a network it will automatically make a DHCP request to Wireless Trondheim to setup the required IP settings. This is why a completely manual connection to the Wireless Trondheim network has to be made. In Windows this can be avoided by choosing the "Use the following IP address" option in the properties window of TCP/IP setup. There is no need to find specific information about the APs MAC and channel.

The following list is a summary of what is needed:

- Clients MAC address
- Clients IP address
- Subnet mask
- Default gateway
- DNS server

The first step is to capture some network traffic to get a hold of a clients MAC and IP addresses. I did this with Wireshark in the Windows based attack. In Windows it is not possible to switch the wireless card to monitor mode like in Backtrack. Instead Wireshark uses promiscuous mode to capture other clients traffic. The purpose of both promiscuous mode and monitor mode is that the network card passes along all packets it receives to the CPU and not only the ones addressed to it. The difference between them is that promiscuous mode requires the wireless network card to be associated with the network the user wants to listen to.

To capture some traffic I connected to Wireless Trondheim and started Wireshark. I had problems with packet capture in Vista on my laptop so for the Windows attack the attacker and client switched places. The victim was the laptop running Windows Vista and the attacker was the former client desktop running Windows XP SP2. Figure 3.6 is a screenshot from a packet dump done on the attacking computer. It shows a client with a highlighted packet containing the IP and MAC address of the client.

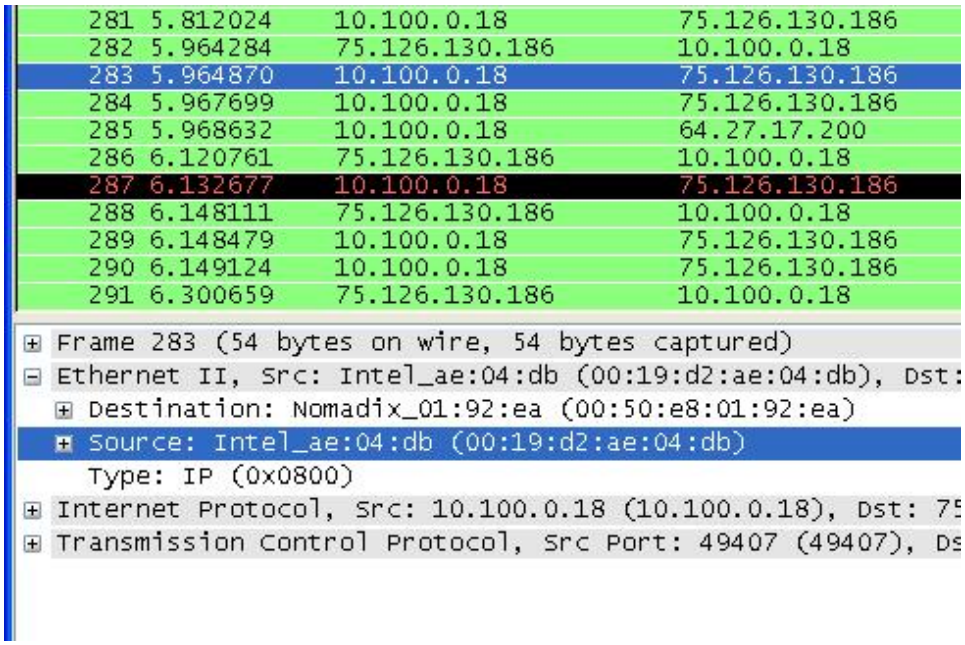


Figure 3.6: Screenshot from wireshark in Windows showing a client with IP and MAC address

- Clients MAC address: **00:19:D2:AE:04:DB**
- Clients IP address: **10.100.0.18**

The rest of the information needed can already be looked up in ipconfig. I started a command prompt window and entered "ipconfig /all". From this window which is similar to figure 3.5 the rest of the information needed about the Wireless Trondheim network could be found.

- Subnet mask: **255.255.254.0**
- Default gateway: **10.100.0.1**
- DNS server: **194.19.2.11 and 194.19.3.11**

#### 3.3.2.1 Executing the attack

The necessary information has been found and the attack can now be performed. First the MAC address of the client have to be spoofed. in Windows there is an easy way and a hard way to do this. The easy way is to configure the MAC address of the network interface from the network configuration window in Windows. The hard way is to change a value in the registry containing the MAC address. Note that when changing the MAC address in Windows it will not reset back to the factory value stored in the hardware of the network card at reboot like it will in Linux.

**The easy way**

Before changing the MAC address in this way go to the desktop and right click my network places and select configure. Then right click the wireless interface and choose disable. This is to stop Windows from renewing the IP address via DHCP automatically after the MAC address has been changed.

Then right click on My network places and select properties. Right click the wireless interface and select properties. Select configure to the right of the name of the network card and go to the advanced tab. If the network cards drivers support it the list will contain a field called Locally administered MAC address or something similar. I clicked the radio button for value and entered the clients MAC address found in the packet capture in the format written below.

```
0019D2AE04DB
```

If the locally administered MAC-address option is not in the list, the harder option of changing MAC address will have to be used.

**The hard way**

The hard way is to change the MAC address in the registry. The following procedure is a short version of the guide from [3].

Some information needs to be gathered about where to look in the registry first. I went to command prompt and enter "ipconfig /all" and took note of the name of the network interface card. I entered "net config rdr" in the same window and took note of the long string of numbers and letters. In my case it was the following.

```
{E09127F0-E706-462E-87BF-829ACD208A88}
```

Then I started the registry editor by typing "regedt32" in the run window from the start menu. The registry contains a large tree of important values used by Windows and it is always useful to make a backup in case something is done wrong. Right click a tree node in the registry editor and choose export to make a backup of all the sub nodes of that node.

Now the field containing the MAC address for the wireless interface has to be found. The first path of the registry path is the following. Make sure the last folder is correct as there is a long list of almost identical folders.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\
{4D36E972-E325-11CE-BFC1-08002BE10318}
```

A list of folders named 0000, 0001, 0002 and so forth should appear and one of these contains the field with the MAC address. To determine which of the folders it is they have to be checked one by one. Most of the folders contains among others a field called **DriverDesc** and another called **NetCfgINstanceId**. When these two fields match the interface name and the string of numbers and letters found earlier the correct folder has been found. In my case the correct folder was 0012.

The field **NetworkAddress** contains the MAC address Windows will use for the network interface. I changed this to the value obtained by capturing network traffic.

### Configuring TCP/IP

Now the attacker computer had the same MAC as the client computer. The next step was to configure the attacker computer to use a static IP. This was done by going to My network places -> properties -> wireless interface -> properties. Then find Internet protocol (TCP/IP), select it and click on properties. Then I selected "use the following IP address" and "Use the following DNS addresses". I entered the information found in the information gathering phase. The last step was to connect to the wireless Trondheim network like usual through the built in wireless netowkr manager in Windows. The attacking computer now had full access to the Internet without going through the authentication procedure, but when the clients MAC address is removed from the Nomadix white list the free access for the attacker is lost.

# Chapter 4

# Countermeasures

This chapter contains a recap of the proposed countermeasures and some thoughts about how to implement them into the Wireless Trondheim network.

## 4.1 Session ID

Session ID is a countermeasure targeted towards the session hijacking and waiting for availability types of MAC spoofing. A network using the session ID countermeasure creates an SSL encrypted connection to a web page with a cookie associated to it. The cookie contains a session ID and is sent to the client after the connection has been set up so eavesdroppers can't find out what the ID is. The web page tells the clients browser to request refreshes with a given interval and the cookie is sent along with each refresh request. As long as the Wireless Trondheim IDS keeps getting the refresh requests with the correct ID it can assume the original client is still active on the network. This was a quick recap of the session ID countermeasure and the following subsection is a suggestion on how it can be implemented in the Wireless Trondheim network.

### 4.1.1 Implementing Session ID

This countermeasure is the easiest one to implement because the elements needed are already present in the Wireless Trondheim network. The main reason for this is that there is a web server involved in the authentication procedure. Figure 2.1 is only a simplified overview and does not show the

web server. In the authentication phase the user is communicating with the web server and the web server tells the Nomadix device to add the user if the authentication is successful. Figure 4.1 illustrates the web servers role in the authentication phase.

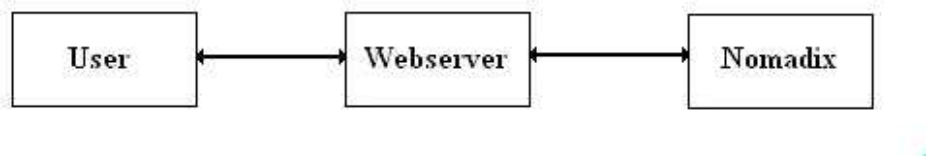


Figure 4.1: The webserver as an intermediary in the authentication phase

It is the web server that sets up the SSL protected communication and verifies the user input such as in the case where NTNU students use their user name and password. The input here is sent by the web server to an authentication system at NTNU.

Only some added functionality in the web server is needed to implement the session ID countermeasure.

Figure 4.2 is a simplified sequence diagram illustrating how the session ID countermeasure would work. The first part which is not shown in the figure is the same as before with the user being redirected to the web-portal authentication page after connecting to Wireless Trondheim. The first step in the sequence diagram is that the user sends his authentication information to the web server. If the authentication succeeds the web server sends a message about adding the user to the Nomadix device.

The next step is were the difference is. Either the user has to be redirected to a new web-page or a new pop-up window will have to be opened. This new web-page will be secured with SSL and get an associated cookie that contains a session ID generated for the current session. The web page will also contain the following HTML code that will make the users browser request refreshes with a given interval.

```
http-equiv="refresh"
```

The associated cookie containing the session ID will be sent along with each refresh request. The first message in the bottom half of the sequence diagram shows that the users browser sends a refresh request containing the session ID. The web server should run a script or program with a certain

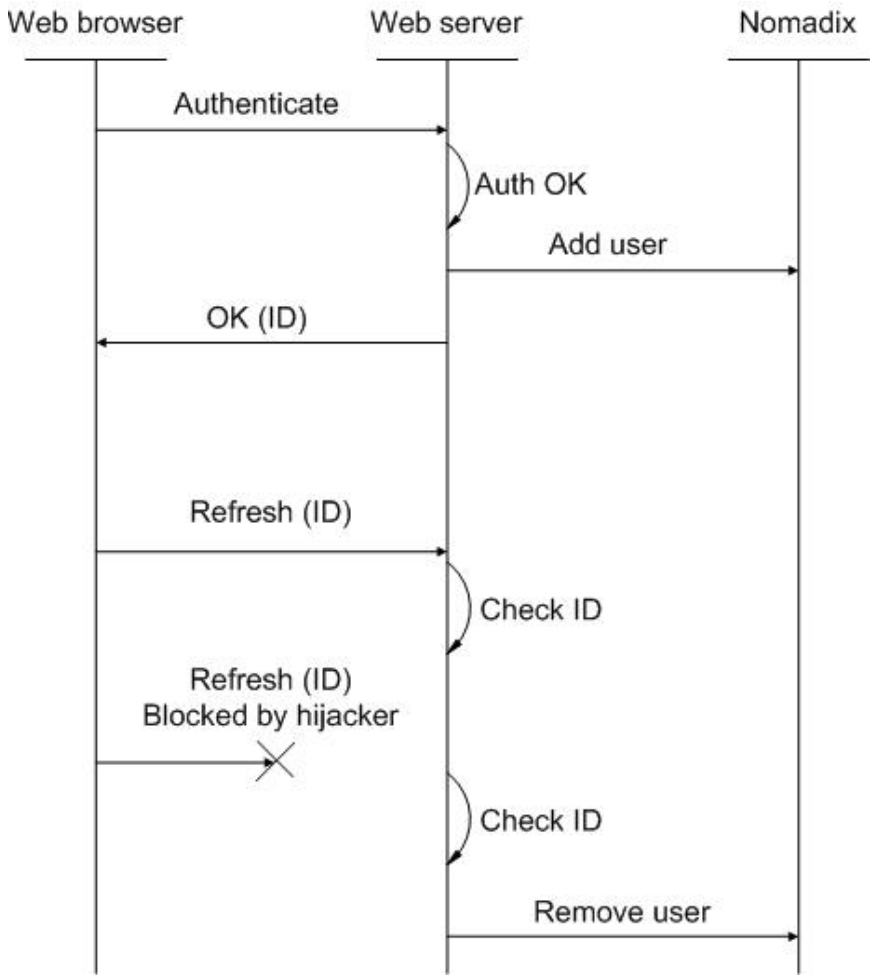


Figure 4.2: Sequence diagram that roughly illustrates how the session ID countermeasure will work

interval and cycle through the list of users and check if all of them have refreshed their ID recently. The last two messages in the diagram shows that a hijacker has thrown the user off the network and the refresh request can not be sent to the web-server. When the web-server runs its check it will discover that the hijacked user has not refreshed his ID and can take som sort of action. This could be either to tell the Nomadix device to remove the user as illustrated in figure 4.2 or to note the event in a log.

## 4.2 MAC sequence number analysis

The MAC sequence number analysis countermeasure is targeted towards the freeloading type of MAC spoofing attack. An IDS using this countermeasure analyzes the sequence numbers in MAC frames. The MAC frame sequence number is incremented by one for every layer 2 frame sent by a wireless interface card. Under normal circumstances with one client using the same MAC address a plot of the MAC sequence number will be a straight line. If an attacker freeloads using the clients MAC address its MAC sequence number counter will be different from the clients. An IDS should be able to detect two different counters in the stream of sequence numbers.

### 4.2.1 Proof of concept for MAC sequence number analysis

This master thesis does not include any practical implementations of the proposed countermeasures. However in the case of the MAC sequence number analysis a simple proof of concept experiment has been conducted. In addition a control experiment was conducted with only the client connected.

The idea is to make a packet dump of the network traffic while the freeloading attack is being performed. By generating traffic simultaneously at both the attacker and the client it should be easy to browse through the packet dump in Wireshark and see that the MAC sequence numbers are making big jumps between two different counters.

The packet dump was done by the passive monitor machine with airodump and the packet dump was started with the following command.

```
root@ubuntu:~# airodump-ng -w maqseqtest
--bssid 00:23:5d:0e:01:90 --channel 11 ath1
```

After a short while with generating traffic in both the clients and attackers browsers the packet dump was stopped and the dump file was opened in Wireshark.

The filter string "ip.src == 10.100.0.80" was used to only show IP packets sent by the client and attacker. The sequence number of the MAC frames these IP packets were sent with can be viewed by expanding the information tab below beginning with IEEE 802.11. By using the arrow keys to scroll down the list of packets it could be observed that two different MAC frame counters was in use.

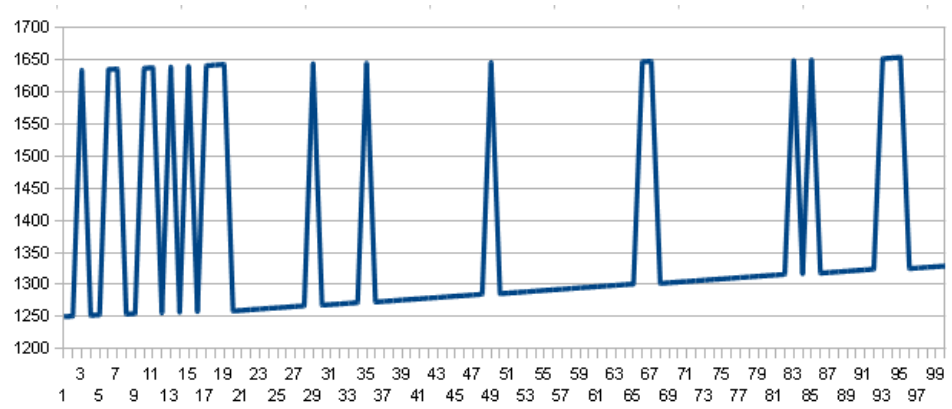


Figure 4.3: The first 100 MAC sequence numbers from the proof of concept experiment

Figure 4.3 is a plot for the first 100 observed MAC sequence numbers in the dump file. Many of the entries had identical MAC sequence numbers and in these cases the first occurrence of the sequence number was recorded. Further studies are needed to determine why the MAC sequence numbers appears in several IP packets. The duplicate sequence numbers also occurred in the control experiment with only the user using the connection, but to a lesser extent. The duplicate sequence numbers also seemed to be connected to duplicate TCP ACKs which also was present in both the control test and the main experiment.

The figure clearly shows that the MAC sequence number makes big leaps between two groups of values. On both the top and bottom values it can be seen that the counter is continuously rising which is the normal behavior of the counter. The computer using the lowest sequence numbers is sending more traffic so its counter is rising more rapidly. For the computer using the highest sequence numbers the increase is not as easy to see, but still visible. It is not easy to tell which of the counters belongs to the client and attacker, but the fact that there is two counters is sufficient to tell that an attacker is present. An interesting note is that if the assymetry in transmitted traffic were to continue the lower curve would cross the higher one and continue upwards.

The graph only shows the order of the packets and not the time between them. Any information about how close together the packets were is lost in this representation, but it clearly shows the two different MAC sequence

counters. It might be useful to know however that the packets in the plot were captured in a timeframe of about 2,4s.

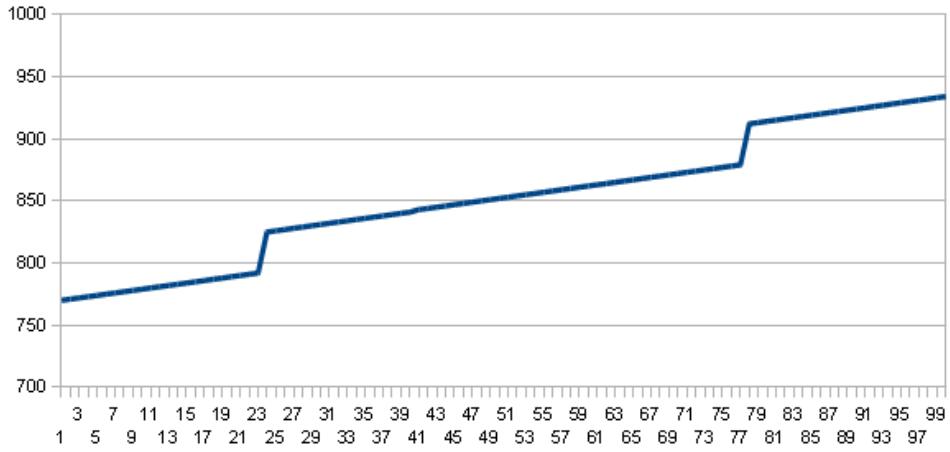


Figure 4.4: 100 MAC sequence numbers from the control test

Figure 4.4 shows a group of 100 consecutive MAC sequence numbers from the control test with only the client using the connection. The expected result was a straight continuously increasing line, but in the control test there were two jumps in the MAC sequence number counter. The reason for this is also subject to further studies. These MAC frames was collected in a timeframe of about 90 seconds which is significantly longer then in the proof of concept test.

### 4.2.2 Implementing MAC sequence number analysis

The most critical aspect of implementing the MAC sequence number analysis will be how to capture and store the MAC sequence numbers from all the frames being sent by clients in Wireless Trondheim.

#### 4.2.2.1 Capturing sequence numbers

If the 802.11 header is removed by the access point before sending it into the wired part of Wireless Trondheim it will be difficult to capture and store them in a central location.

The ideal situation would be that the 802.11 header used on the air between the client and AP is available in a central point in the network. A computer could listen to the the traffic in this central point and record the

MAC sequence numbers of all the frames sent by clients. How this will be done in an implementation is subject to further studies.

#### 4.2.2.2 Storing and analyzing sequence numbers

When a user authenticates a unique session should be created and all the MAC sequence numbers sent by that user is associated with that session. The IDS system can then analyze each session at a later point in time and determine if more then one MAC sequence number counter is in use.

It may seem like a huge task to monitor all the traffic being sent by the users of Wireless Trondheim, but there are a few aspects of the tasks that might make it more feasible. The first thing is that only the traffic being sent by the clients needs to be monitored. The traffic pattern of the majority of Internet users is that they download a lot more then they upload so the amount of traffic that requires analysis might not be as large as one might expect. Another aspect is that the IDS only needs the MAC sequence number from each MAC frame. The sequence number value is 12 bits long so the required storage capacity will be 12 bits pr frame being sent by clients connected to Wireless Trondheim. In addition there will be some overhead depending on how the sequence numbers are being stored and associated with their respective sessions.

## 4.3 Physical parameters

The physical parameter countermeasures base themselves on physical properties of the radio signals. These physical properties are the time it takes to complete a RTS-CTS handshake and the signal strength of received frames from clients. When an attacker performs a MAC spoofing attack the IDS detects changes in these values and takes action. The IDS correlates between the two methods and a detection must be made by both before an alarm can be raised.

### 4.3.1 Implementing physical parameters countermeasures

These countermeasures are perhaps the most difficult to implement in a public wireless network such as Wireless Trondheim. The important information here is the received signal strength of frames received by all the access points

and round trip times for RTS-CTS handshakes. The authors of [9] uses a passive sensor with a prism2 chip set. This chip set has the ability to append an additional header to each frame in addition to the 802.11 header. This extra header contains physical parameters such as received signal strength and timestamps for the frames.

The information about physical parameters are only available to the access points and have to be recorded in the same way the prism2 chip set does it or in some other similar way. Once the frames are sent into the wired part of Wireless Trondheim the required information can not be recreated by any means. The implementation of these countermeasures are therefore reliant on what functionality the chip set in the wireless Trondheim access point can offer.

In the practical experiment done in [9] they use a separate sensor to record the information needed by the IDS. In Wireless Trondheim it is not practical to deploy sensors in addition to the already existing infrastructure of access points covering the city of Trondheim. The implementation of these countermeasures depends on the access points ability to record this information when receiving frames from the users.

# Chapter 5

# Results

This chapter contains a summary of the work that has been done in this masters thesis. The main goal stated in the problem description his how to get technical evidence of a session hijacking attack that has taken place. The first step towards this goal is to find ways that session hijacking attacks can be performed. And the target of these attacks as stated in the problem description is the existing infrastructure in Wireless Trondheim. Given the two access mechanisms in Wireless Trondheim which is the less secure portal solution and the more secure RSN based eduroam solution the portal solution is the most natural candidate for a session hijacking attack.

## 5.1 Penetration testing

The portal solution in Wireless Trondheim uses only MAC filtering so the attack of choice is a MAC spoofing attack. In [10] Brustolini and Xia describes three variants of MAC spoofing attacks. These are the hijacking variant where the client is actively thrown off the network, the freeloading variant where the attacker communicates simultaneously with the client and the third variant of waiting until the client is not using the network.

To do practical experiments a test bench was needed. A suitable test bench illustrated in figure 3.1 was set up which consisted of a laptop computer acting as the attacker, a desktop computer as the regular client and a desktop computer acting as a passive sensor to monitor the traffic between client attacker and AP. The test bench was set up in Wireless Trondheims office in room 241 in the IT-building at NTNU. A Wireless Trondheim AP

was installed in the office to provide access to the production network in Wireless Trondheim for the test bench.

Backtrack was installed on the attacker computer and the freeloading variant of MAC spoofing attack was performed successfully. The same type of attack was also performed in Windows XP to prove that it can also be done with a less specialized OS.

## 5.2 Countermeasures

The second part of the problem description is to look at ways to gain technical evidence proving that a session hijacking attack has taken place. Some possible countermeasures were found in [10] and [8]. The four proposed countermeasures are session ID, MAC sequence number analysis, received signal strength and RTS-CTS handshake. Some thoughts have been made about how each of these countermeasures can be implemented in the Wireless Trondheim network and what the difficulties are. A proof of concept experiment was conducted for the MAC sequence number analysis countermeasure.

# Chapter 6

# Discussion

## 6.1 Real-time analysis vs logs

Most intrusion detection systems do real time analysis of the network traffic to detect suspicious behavior. That way they can take action immediately when an attack occurs. Both a reactive IDS and a passive IDS does real time analysis and the difference is that the reactive IDS takes direct action to stop an attack and the passive IDs makes log entries or alarms.

What I suggest for an implementation in Wireless Trondheim is that there is no real-time analysis of the network traffic. The necessary data should be recorded in logs for analysis at a later point in time Wireless Trondheims main concern is not to block attackers that only want free access, but if a user is accused of criminal activities the IDS should be able to tell that his network identity was spoofed or not. Blocking users leeching on the network requires real-time analysis, but if any user is accused of criminal usage of the network it will be days or weeks after the actual incident.

If any criminal activity is reported the logs can be analyzed with the appropriate methods to see if any suspicious activity can be found. The assumptions one can make from the result of the analysis depends on the reliability of the IDS. Lets say Wireless Trondheim is asked by the police to supply information about a suspect in a criminal investigation. If the IDS is perfect and the suspects activities does not show signs of identity theft the suspect will be guilty of using Wireless Trondheim for criminal activity. If the suspects activities does show signs of identity theft the user is either innocent or made an attack on himself to avoid suspicion.

However an IDS system can never be perfect and any results from the analysis of network traffic can only be used as circumstantial evidence. More solid evidence will need to be gathered to be sure that a suspect is guilty, but the result from an IDS in Wireless Trondheim can help to either strengthen or weaken the suspicion towards the user.

## 6.2 False positives vs false negatives

In Wireless Trondheims case the most important aspect is to avoid false negative situations. A false negative situation is equivalent to an attacker doing MAC spoofing without being detected by the IDS. The main goal of Wireless Trondheim is to detect MAC spoofing in the case of criminal usage of the network. Minimizing false negatives which is a good thing requires an IDS to be more strict, but that also increases false positives which is a bad thing. Too many false positives makes the IDS loose its credibility, but avoiding false negatives should still be the priority of an IDS solution in Wireless Trondheim so a balance needs to be found.

## 6.3 Countermeasures

This section will discuss the proposed countermeasures and how well they fit into Wireless Trondheims needs and wishes for a possible IDS solution.

### 6.3.1 Session ID

The major drawback of the session ID countermeasure is the pop-up window that will contain the web-page with the associated session ID cookie. The page could contain a warning to the user that his connection to Wireless Trondheim will be terminated shortly after closing it. This would be annoying for many users which might want to close it and many users will probably close it from forgetfulness or from being unaware of the warning. Wireless Trondheim does not wish to decrease the user friendliness of their current access solution.

To throw clients off the network in real time is easy to implement with this countermeasure. This is because it relies on the continuous check to see if the clients have updated the session ID recently. However Wireless

Trondheim does not wish to throw clients off the network in real-time because their main focus is any long term criminal consequences of session hijacking.

Another solution could be to make logs where it is recorded if a client fails to make its session ID refreshes. The downside to this is that this log would contain a lot of false positives from clients that e.g. forgets to keep their pop-up window open.

This countermeasure is the easiest one to implement, but is has some drawbacks that makes it unsuitable for a practical implementation. The decrease in user friendliness from the pop-up window and the disputable usefulness of making logs instead of throwing users off in real-time is the reason why this countermeasure is not recommended for implementation in Wireless Trondheim.

### 6.3.2 MAC sequence number analysis

This countermeasure is the one that fits the best into Wireless Trondheims wishes about a possible IDS solution against hijacking attacks. The presence of this countermeasure will not be noticeable by normal users of Wireless Trondheim at all. All the countermeasure does is to extract the MAC sequence number from the frames they send. The countermeasure does not store any of the data the clients send so an implementation will not be affected by legal issues surrounding data storage.

The countermeasure is also ideal for the logging approach that Wireless Trondheim wishes to follow. The IDS can make logs containing the MAC sequence numbers for each user session and the relevant logs can be analyzed at a later point in time in the event of any criminal activity.

The only uncertain aspect of the countermeasure is how the IDS will gather the MAC sequence numbers from frames sent by the users. Given that this can be implemented in the Wireless Trondheim network I recommend this countermeasure for further study and implementation.

### 6.3.3 Physical parameters

The two countermeasures in this category are the ones with the most uncertainty tied to them. The data about received signal strength and RTS-CTS handshakes will have to be recorded by the access points and stored in some

central location. Unless the chip set in the access points have functionality to do this it could be very hard to gather this data.

It is also uncertain how this countermeasure will work in a full scale network in an urban environment such as wireless Trondheim. The practical testing in [9] was done in a small scale network involving office landscapes, but more information on how it would work in a large network like Wireless Trondheim is needed.

On the plus side these countermeasures fit well with Wireless Trondheims wish to make logs instead of actively kicking users. As with the MAC sequence number countermeasure the received signal strength and RTS-CTS values can be stored and analyzed at a later time if necessary.

These countermeasures are overall not recommended for a possible implementation because of the uncertainty concerning how well they will work in a full scale urban network and how the IDS will gather the necessary information.

# Chapter 7

# Conclusion and future work

## 7.1 Conclusion

The thesis has proven that a session hijacking attack in the form of MAC spoofing against Wireless Trondheim is easy to perform. No specialized OS is required to be able to perform the attack. At the time of writing this thesis there is no mechanism in Wireless Trondheim that can detect a MAC spoofing attack. A person with criminal motives may use an innocent users session to perform criminal activities and Wireless Trondheim will not at the present date be able to tell that the innocent user was hijacked and might not be guilty.

### 7.1.1 Relevant countermeasures

The thesis has proposed these countermeasures and evaluated if they are recommended for a possible IDS solution in Wireless Trondheim that will protect against session hijacking attacks. The basis for the evaluation is Wireless Trondheims wishes for the IDS and what the difficulties are for implementing them in the Wireless Trondheim network.

#### Session ID

This countermeasure is the easiest to implement, but it has two drawbacks that does not fit into Wireless Trondheims requirements for an IDS. It decreases user friendliness by requiring a browser window to stay open to avoid triggering the IDS countermeasure. This countermeasure is also best suited for throwing suspected users off the network in real time which is something

Wireless Trondheim does not wish to do. Based on these two drawbacks the countermeasure is not recommended for further work.

#### MAC sequence numbers

This countermeasure fits best with Wireless Trondheims requirements for a possible IDS implementation. It is suitable for making logs that can be analyzed a long time after an incident took place. This countermeasure should be feasible to implement given that the MAC sequence numbers are available at a central point in the wireless Trondheim network. This is the countermeasure recommended by the thesis for a possible future IDS solution.

#### Physical parameters

The countermeasures in this category have the most uncertainty tied to them. They will probably be the hardest to implement because the data required by the IDS can only be recorded by the APs and an implementation relies on their ability to do so. It is also uncertain how well this countermeasure will work in an urban environment with many moving clients and highly unpredictable paths and reflections for radio waves. This countermeasure is not recommended for further work.

### 7.1.2 Additional security needed

This thesis recommends one countermeasure that will protect against the freeloading variant of MAC spoofing attack. In addition to this variant there is the hijacking type and the waiting for availability type that will not be detected without additional IDS mechanisms. Ideally all the types of attacks should be protected against, but with the lack of a countermeasure that fits with Wireless Trondheims wishes the thesis can only recommend protection against the freeloading type.

## 7.2 Future work

The initial ambition for this thesis was that more substantial work would be done on the countermeasure part of the thesis. This was also the wish from Wireless Trondheims viewpoint and the countermeasures part is represented

more strongly in the problem description then in the thesis itself. The recommendations for future work is therefore focused on countermeasures and on the MAC sequence number countermeasure in particular.

An evaluation of how the MAC sequence numbers can be gathered and tied to its respective sessions should be conducted. If this is found to be possible to realize in the existing Wireless Trondheim network work can begin on implementing the MAC sequence number analysis countermeasure.

More information on the behavior of the MAC sequence number counter will be needed to make reliable analysis of the MAC sequence number logs possible. This thesis has for the most part assumed that the MAC sequence number counter is always incremented by one for each frame, but the control test in 4.2.1 shows a deviation from this.

# Bibliography


[1] BT USB stick with changes tutorial. http://forums.remote-exploit.org/showthread.php?t=7844, 2007.

[2] IEEE standard 802.11. http://standards.ieee.org/getieee802/download/802.11-2007.pdf, 2007.

[3] Changing MAC addres in windows. http://www.nthelp.com/NT6/change_mac_w2k.htm, 2009.

[4] Cisco unified wireless network. http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps430/prod_brochure09186a0080184925_ps6305_Product_Solution_Overview.html, March 2009.

[5] Eduroam homepage. http://www.eduroam.org/, February 2009.

[6] GNU GPL. http://en.wikipedia.org/wiki/Gpl, 2009.

[7] Wireless Trondheim home page. http://www.tradlosetrondheim.no, 2009.

[8] Looi Gill, Smith and Clark. Passive techniques for detecting session hijacking attacks in IEEE 802.11 wireless networks. *AusCert 2005*, 2005.

[9] Smith Gill and Clark. Experiences in passively detecting session hijacking attacks in ieee 802.11 networks. *ACM International Conference Proceeding Series; Vol. 167*, 2006.

[10] Brustolini Xia. Detecting and blocking unauthorized access in wi-fi networks. *NETWORKING 2004*, 2004.