NTNU
Norwegian University of
Science and Technology

Wireless LAN auditing procedure for industrial environments

Magnus Andreas Ohm

Master of Science in Communication Technology
Submission date: January 2017
Supervisor: Danilo Gligoroski, IIK

Norwegian University of Science and Technology
Department of Information Security and Communication

NTNU – Trondheim
Norwegian University of
Science and Technology

Wireless LAN auditing procedure for industrial environments

Magnus A. Ohm

Submission date: January 2017
Responsible professor: Danilo Gligoroski, ITEM
Supervisor: Mate J. Csorba, DNV-GL

Norwegian University of Science and Technology
Department of Telematics

Title: Wireless LAN auditing procedure for industrial environments

Student: Magnus A. Ohm

Problem description:

Having internet access has these days become a necessity for industry as well as social activities. This has also become a fact for offshore and maritime environments. These environments are increasing the use of wireless network communication. The use of these WLANs ranges from sensors communicating with control systems, to crew using wireless networks for entertainment. Installing these networks may introduce challenges we don't normally see in onshore networks. These challenges might lead to bad decisions and shortcuts when installing offshore WLANs. Seeing as these networks might affect crucial industrial equipment as well as sensitive data, it's important to test the security, robustness and availability of these networks in these environments.

The main goal of this project is to create a well defined procedure for testing the security, robustness and availability of offshore WLANs. This procedure needs to be specifically sutured for the test circumstances. If you were to perform these kinds of tests on a ship, there would be several non technical aspects that would need to be considered. You would have to avoid disrupting the daily business and follow strict safety rules. This will among other things lead to the testers only being granted a limited amount of time for testing. All the restrictions related to these types of testing environments will have to factor into the creation of a suitable testing procedure.

Due to the fact that the testing needs to be done in a limited amount of time, then most of the work needs to be done before actually entering the testing area. The procedure will therefore need to contain simple steps where all tools and scripts are ready to run, once you enter the testing environment. There will be done testing at DNV GLs offices at Tiller Trondheim, simulating real maritime and offshore WLANs. The tests done here will focus on finding the best software and hardware tools for getting the best test data in the limited amount of time that is given. Supplementing software will also be developed if it's deemed necessary. There will be Both software and hardware will be tested in various network setups. This is to see how well these tools perform under different circumstances. Although most of the time will be spent evaluating, testing and creating tools, there will also be a significant amount of time spent setting up and configuring test networks. It's important for the project that the final procedure is based on results from various realistic scenarios. This is so that the results will reflect a variety of different offshore environments and not a single

test setup.

The test results found in this project will, in combination with how much time and effort the tools require, determine the final outcome of the procedure. Additionally, the procedure should try to avoid disrupting the daily business and should not break any rules set in the offshore/maritime environment. There are of course a lot of things in the offshore environments that will be difficult to predict. The created procedure should therefore also try to be as flexible as possible.

Responsible professor: Danilo Gligoroski, ITEM
Supervisor: Mate J. Csorba, DNV-GL

Abstract

Today's industry is dependent on computer networks. These computer networks are a vital part of how industrial environments operate. They are used for a variety of different tasks. Networks are needed to do everything from operating possibly dangerous equipment, to support employees in their every day activities. Having to support such a variety of tasks means that these networks will need to fulfill a lot of different requirements to function in a proper and safe way. DNV-GL has seen that these requirements are often not upheld in industrial environments. They have therefore seen a business opportunity when it comes to testing networks that operates in these types of environments.

This project focuses on finding good ways to test WLANs in industrial environments. This has primarily been done by testing different tools and methods for assessing WLAN security and quality. Every network in industrial environments will be different. This means that a possible testing procedure will vary from project to project. It's therefore important to define a testing procedure that captures the most important testing aspects. How to define a testing scope and identifying WLAN requirements have therefore been necessary. The methods and tools used in this project try to cover the most important aspects of WLAN quality and security testing that testers may face in industrial environments. The tests and theory that this document contains should provide testers with the means to detect flaws and shortcomings in clients WLANs.

WLAN quality testing has primarily focused on different ways to perform site surveys. WLAN security testing has focused on ways to access devices that are crucial for a companies safety and daily business. The results and experiences gained from this project has been used to create two testing procedures. These testing procedures are step-by-step guidelines that can be used to test WLAN quality and security.

Sammendrag

Dagens industri er avhengig av datanettverk. Disse datanettverkene er en viktig del av hvordan industrielle miljøer opererer. De er brukt til en mengde forskjellige oppgaver. Nettverk er nødvendig for å gjøre alt fra å styre potensielt farlig utstyr, til å støtte ansatte i daglige gjøremål. Det å måtte støtte et så bredt spekter av forskjellige behov betyr at disse nettverkene har mange forskjellige krav som må oppfylles for å kunne operere på en skikkelig og trygg måte. DNV-GL har sett at disse kravene ofte ikke blir oppholdt i industrielle miljøer. De har derfor sett en forretningsmulighet når det kommer til testing av nettverk som opererer i denne typen miljøer.

Dette prosjektet fokuserer på å finne gode måter for å teste WLAN som opererer i industrielle områder. Dette har hovedsakelig blitt gjort ved å teste forskjellige verktøy og metoder for å evaluere kvaliteten og sikkerheten til disse WLAN-ene. Hvert nettverk i industrielle miljøer vil være forskjellige. Dette betyr at en potensiell testprosedyre vil variere fra prosjekt til prosjekt. Det er derfor viktig å definere en testprosedyre som fanger de viktigste test aspektene. Hvordan man kan definere et testomfang og identifisere kravene til et WLAN vil derfor være nødvendig. Metodene og verktøyene som har blitt brukt i dette prosjektet prøver å dekke de viktigste aspektene av kvalitet og sikkerhetstesting av WLAN som testere kan møte i industrielle miljøer. Testene og teorien som dette dokumentet inneholder burde gi testere de nødvendige midlene for å oppdage feil og mangler i kunders WLAN.

Kvalitetstesting av WLAN har primært fokusert på å finne forskjellige måter å utføre "site surveys" på. Sikkerhetstesting av WLAN har fokusert på hvordan man kan finne måter å få tilgang til utstyr som er kritisk for en bedrifts sikkerhet og daglige drift. Resultatene og erfaringene som jeg har fått gjennom dette prosjektet har blitt brukt til å lage to testprosedyrer. Disse testprosedyrene er steg-for-steg rettningslinjer som kan bli brukt til å teste WLAN kvalitet og sikkerhet.

Preface

This Master's Thesis has been carried out at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. This report uses different methods and tools to find good ways of testing WLAN quality and security in industrial environments.

I would like to thank and acknowledge my supervisors Mate J. Csorba, from DNV-GL, and Danilo Gligoroski, from NTNU's Department of Telematics. They have supported me with feedback during my project, which have helped me reach the final results.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

The use of communication technology has become a necessity in most industrial environments. It's used for everything from controlling critical industrial equipment to giving employees the possibility for leisure activities. Although communication technology gives industry a lot of advantages, it can also lead to a lot of difficulties. There are many things that can go wrong. The network can be badly designed, poorly implemented, the equipment may be outdated, or there can be a general lack of network maintenance. Many industrial environments may have very specific challenges, that may have led to less than optimal solutions.

Industrial environments that operate expensive and potentially dangerous equipment, should have both strict security and quality requirements. These requirements should be upheld to ensure a safe and productive business environment. However, based on DNV-GLs experiences in these types of environments, this is not always the case. Some environments may for example take a little to lightly on their security requirements due to their remote locations. This may be the case for industrial environments such as ships that most of the time believe they are out of reach for potential attackers. These industrial environments may also simply lack the knowledge for how to secure their systems. What ever the reason may be, the result of vulnerable systems in these types of environments may prove fatal.

Most industrial environments have important control systems that controls crucial equipment. If we only focus on the security of these systems, then it would be ideal to lock down the systems as much as possible. However, this will affect the effectiveness and simplicity of daily operations. Industrial environment may be dependent on for example WLANs to be able to uphold their daily business. It's therefore important that there is a balance in these WLANs that both meets the business requirements for WLAN security as well as WLAN quality. Neither can be neglected as they both serve an important role in today's industry.

DNV-GL has seen that industrial environments lack the security and the quality that their networks should have. The actual testing of these environments have therefore proven to be a potential business opportunity for DNV-GL; which in terms have spurred a wish to create a procedure for testing both the quality and security of WLANs in these types of environments.

1.2 Testing Procedure

This project has focused on creating a testing procedure for industrial environments. The procedure focuses on WLANs and the effects that the WLANs may have on the rest of the environment.

Each industrial environment is unique and will therefore have different challenges when it comes to performing a testing procedure. However, this procedure should still consider general difficulties that the testers may face in these types of environments. One thing that we need to consider is the strict regulations and safety rules these environments may have. For example, DNV-GL has done other types of testing aboard ships. In these cases, DNV-GL's testers were only allowed to test in a very limited amount of time (2-3 days). This was while the ship was ashore. They also had to be accompanied by one of the crew members at all times. What DNV-GL has seen in these types of tests is that they will only be given a limited amount of time to perform their tests. There are also limitations to what the testers are allowed to do in these scenarios, i.e., only test certain parts of the network or visit specific areas. This may again be due to strict safety rules, or it may be due to the possibility of disrupting daily business, or the protection of certain company secrets. It's therefore important to have possible solutions for these types of scenarios as it could affect the test results.

The most important part of this procedure is to find good methods that tests important aspects of a WLANs security and quality. The accuracy and effectiveness of performing these methods is also an important part of this project. This is why finding good tools are important. Good tools can assist testers in obtaining accurate results in an efficient manner. However, there are an abundance of different tools to choose from. Knowing which tools the testers can use and how they can use them has therefore been an important aspect of this project. A lot of different types of tools has been researched and tested in this project to see which will serve our purpose best.

1.3 Project Goals

The main goal of this project is to create a WLAN testing procedure that tests both the quality and security of said WLANs. The two aspects of the testing procedure

will have separate goals.

- **Goal 1: WLAN quality testing**
  This project will try to find proper tools and methods to test WLAN quality. The quality testing aims to map out whether or not the WLANs can support the clients intended network use. The clients should not have to restrict what services they are able to use based on the current WLAN design.
- **Goal 2: WLAN penetration testing**
  This project will try to find proper tools and methods to test WLAN security. The penetration testing (pentesting) procedure will attempt to find security flaws related to the clients WLANs. The tests should try to figure out if the WLANs can be used as entry points for attackers that may compromise parts or the whole network.
- **Goal 3: Step-by-step procedure**
  The lessons learned in this project will be used to create a step-by-step procedure. This step-by-step procedure should be a simple and structured guide. It should tell testers which actions they need to take. Having a structured procedure should ensure good test results that creates value for the clients.

1.4 Report Overview

This project contains a total of 8 chapters. Chapter 2 describes the tools that were properly tested during this project. Chapter 3 describes related work that has influenced this project. Chapter 4 describes how to figure out the scope and requirements of the testing procedures as well as the information the testers need to get from the clients. Chapter 5 describes different methods and tools that has been used to test WLAN quality. Chapter 6 describes different methods and tools that has been used to test WLAN security. Chapter 7 contains the final testing procedures that are based on the results and experiences described in previous chapters. Finally chapter 8 contains the conclusion of this project. This document is intended to support testers with different levels of knowledge. The document therefore tries to explain different theory and methods that needs to be considered.

Chapter 2
Tools

An important aspect of this project has been to find tools that are well suited for a testing procedure. Different test methods requires different tools. Tool selection has been made based on tool features and performance. This chapter gives an overview of different tools that have been tested. It should be mentioned that all tools used for the WLAN quality testing was done using Windows 10. All tools used for pentesting were used on Ubuntu 14.04.

2.1 Wifi Pineapple

The Wifi Pineapple has been used a lot during this project[pin]. It's a wireless auditing platform with a variety of different modules suited for wifi penetration testing. This piece of hardware is developed by Hak5. I have used the Wifi Pineapple Tetra, which is currently the top model. This model supports both the 2,4 GHz and 5 GHz band. The Wifi Pineapple comes with several default modules for wireless network discovery, network configuration, filtering and setting up rogue Access Points (APs). It also has several additional modules available for installation and an API which gives you the opportunity to create your own modules. The official API is intended for PHP programming. There is however an unofficial API that enables the use of python. The Wifi Pineapple can be controlled from both the command line, and from a web interface. The command line of the Wifi Pineapple is accessed through a secure shell (ssh) connection. The web interface has been completely redesigned for the Wifi Pineapple Tetra. The Wifi Pineapple has many different modules that has been used in this project. However, the tool and most of its modules are lacking when it comes to documentation. Fortunately, the web interface makes it relatively easy to understand how the modules work. This is because it generally provides the user with a better and more logical overview of module features. This also makes the use much more efficient.

2.1.1 Recon

Recon is one of the default modules in the Wifi Pineapple. It's very simple and good for mapping APs and their clients. It has a simple GUI that allows the user to scan the 2,4GHz band and/or the 5GHz band, and will try to discover all devices in range of the Wifi Pineapple. The tool simply outputs all APs with their SSID, MAC-address, Security protocol, whether WPS is used, which channel the AP is using and how strong the signal is. All APs also have a list of connected clients and their MAC-address. The module also gives you the opportunity to deauthenticate clients from their access point. This can be used capture WPA/WPA2-PSK handshakes or get to them connect to a rouge AP.

2.1.2 Signal Strength

The Signal Strength module is not a default module, and needs to be installed on the Wifi Pineapple. It provides you with a lot of the same information that the Recon module does about nearby APs, but does not provide any information about clients and cannot deauthenticate them. It does however provide users with informational charts that represents the signal strengths of nearby APs.

2.1.3 Site Survey

The Site Survey module is not a default module on the Wife Pineapple and needs to be installed. This module collects similar data to what the Recon module does, but is a little more detailed. An example of the increased detail is that it provides a more descriptive overview of the security configurations used by the APs. It shows one Encryption field (WEP/WPA/WPA2/None), one Cipher field (CCMP, TKIP) and one Authentication field (PSK/802.1x/None). It also gives you the options of capturing handshakes for one or all APs, and deauthenticating one or all clients.

2.1.4 PineAP

PineAP is the module responsible for running the rogue AP features of the Wifi Pineapple. It has several features that allows clients to associate to the rogue AP, log probes that are sent, log the associations that is detected, whether or not the PineAP should run as a daemon in the background and whether the rogue AP should respond to beacons. It also makes it easy for you to choose which WLANs you are going to mimic. The module can choose to only pretend to be apart of one WLAN, or it can pretend to be a part of any WLAN that clients may search for. You can also choose how aggressive you wish your rogue AP should be with broadcasting its presence and answering clients.

2.1.5 Evil Portal

Evil Portal is a module that cooperates with the PineAP module. It's used to act as a starting web-portal for users that connect to the rogue AP. This web portal is the first thing that clients see in their web browser once they are connected to the rogue AP. Evil Portal is quite open as to what you can use this web-portal for. Because the portal is so open, the tester is required to do some programming; mainly designing the front-end, the fields needed, and communication between the front-end and back-end.

2.2 Password Cracking

One of the most common mistakes to do when setting up and configuring a network, is either using an insecure security protocol or setting a weak password. This is in spite of the well known weaknesses this imposes on your network.

2.2.1 Aircrack-ng

Aircrack-ng is a tool-suite that has various pentesting capabilities[Air]. It can be used for monitoring, packet injection and password cracking. It's meant for Linux distributions, but does also support different operating systems like Windows and OSX. It should be mentioned that this tool-suite does not support all chipsets on wireless cards. The Wifi Pineapple comes with aircrack-ng as one of its default modules in its terminal. In this project I used this tool-suite for cracking WEP-keys and WPA/WPA2-PSK passwords. There is not a single tool in the aircrack-ng tool-suite that can perform the entire task of cracking WEP or WPA-PSK, but it contains a combination of tools that together can complete the task.

2.2.2 Pyrit

Pyrit is a password cracker. Like aircrack-ng it can be used to perform dictionary or brute-force attacks to find WPA/WPA2-PSK passwords. Finding wifi passwords can be a time consuming task. It's therefore important to have the most efficient tools for this task, which is why multiple password crackers have been tested in this project. Pyrit can use all the cores of your CPU and GPU as opposed to aircrack-ng (which only uses the CPU). This can really lower the processing time. On the other hand, Pyrit does not have the capability of capturing the WPA/WPA2 needed to find the correct password.

2.2.3 Wifite

Wifite is a simple tool that I have used for finding WEP or WPA/WPA2-PSK handshakes. If you are going to use the aircrack-ng tool-suite, then you would have

to use three different tools to capture handshakes efficiently. Wifite combines these three tools into one simple tool, thereby simplifying the entire process. This tool also needs a decent network card to function properly.

2.2.4 John the Ripper

John the Ripper is a very popular tool for aiding in dictionary and brute-force attacks. It can be used to generate wordlists, process wordlists or generate them "on the fly". So it's a great supplement for password crackers such as Aircrack-ng and Pyrit since it can be used to better your existing wordlists, or generate words for bruteforcing the password.

2.2.5 THC Hydra

THC Hydra is a tool used to crack the credentials for different types of services. As with Aircrack-ng and Pyrit, you can use wordlists to perform dictionary attacks. However, seeing as many of the services you wish to attack use both usernames and passwords, you will need two separate wordlists for testing username and password combinations. This tool can be used to attack more than 50 different services. Some of the more known services are ssh, telnet, http and ftp servers.

2.2.6 Ncrack

Ncrack is also a tool that can be used to crack the credentials of different types of services. As with THC Hydra, this tools needs wodlists for both usernames and passwords. Ncrack supports about 10 different services. Among these are ssh, telnet, http and ftp servers.

2.3 Nmap

Nmap is one of the most popular networking tools in the world, with thousands of downloads every day [nma]. It's a network discovery and security auditing tool with a lot of different features. Being able to handle all of Nmaps features is a rather big project in itself. There are however simple features which makes it easy to use for beginners as well. Nmap can be used for host detection, port scanning, OS detection and service detection. Nmap also comes with its own script engine. This engine comes with a lot of good scripts. Some of the scans performed by Nmap are quite extensive. This means that scans can take a while, especially when you are handling big networks with dozens or hundreds of devices.

2.4 Site Survey Tools

The WLAN quality testing done in this project has focused on different ways to perform site surveys. This project has therefore tried to find proper tools to use for this purpose.

2.4.1 Ekahau

Ekahau-heatmapper is a site survey tool that displays WLAN quality using heatmaps[ekaa]. It's very simple to install and can be up and running in a matter of minutes. Ekahau-heatmapper is created for use on Windows operating systems. The Ekahau-heatmapper is one part of a larger tool for WiFi-design and site survey. There are a lot of different versions of this tool, with prices that starts at $2295[ekab]. However, the Ekahau-Heatmapper is free of charge.

Ekahau-heatmapper is simple and intuitive to use, meaning that novices can pick it up without needing any prior instructions or experience with the tool. You will need to give the tool a map of the testing area. The tool will collect signaling data from the area, which it maps to the location you are indicating on the map. As you walk around, you can simply update your current location. Ekahau will not need you to stop at any location as long as you try as often as you can to click the location you find yourself in. Once you are done, you can stop the test scenario and a heatmap, representing the signal strength, will automatically be generated.

The tool has a bar on the left hand side which provides information about the APs you have detected. This bar will contain information about every AP's MAC-address, SSID, security protocol, 802.11 protocol version, channel used and maximum possible bandwidth. It will also show which type of AP this is and if the heatmapper is able to recognize it. Each AP will be located at the tools estimated location. The tool will only place APs on the map if it feels fairly certain of their location. Figure 2.1 shows a heatmap of the WLAN quality lab using the Ekahau-heatmapper.

It should be mentioned that this free version of the Ekahau tool lacks a lot of importat features. For example, once a heatmap is created, there are basically no way to modify the heatmap to your specific needs. One of the most important features missing in the tool is the inability to automatically group together APs with the same SSID. This means that it cannot group APs that are in the same WLAN. The user cannot make this grouping manually either. The tool will only generate a heatmap of all APs in the area. This makes the Ekahau-Heatmapper unsuitable for our test procedure. This is because we need to tell which WLAN the heatmap coverage represents.

The professional versions of this tool does however seem to be one of the best

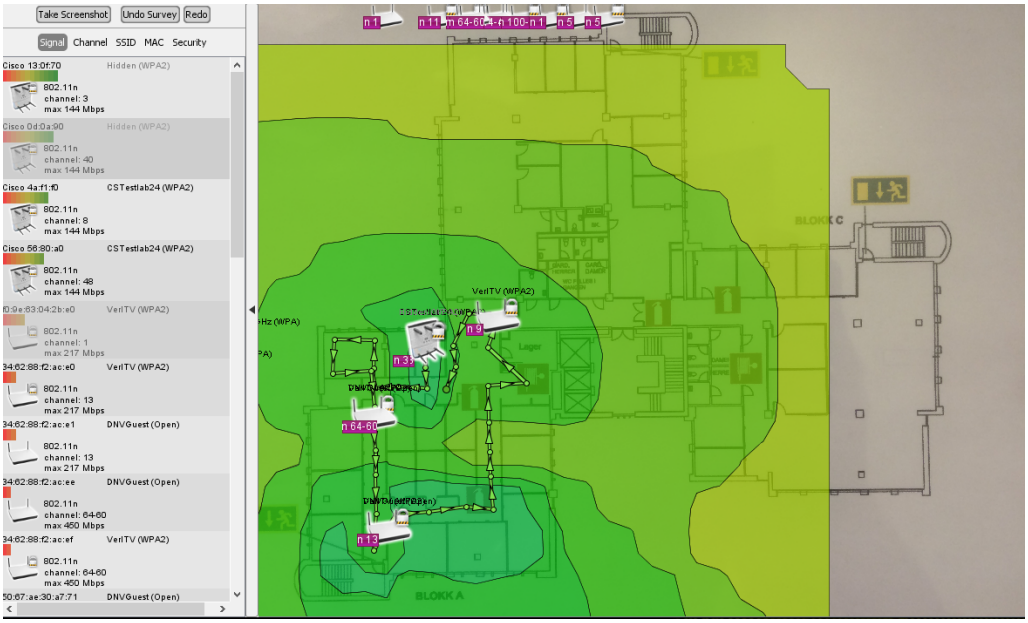

Figure 2.1: Heatmap generated using Ekahau-Heatmapper

tools on the market. This tool does enable the users to group APs belonging to the same WLAN. It has a wide range of features and is quite complex. This is why Ekahau provide their own training and certifications[ekac]. There are two positive aspects that separates the professional version of Ekahau from the other professional versions of tools I have tested. The main one is that it allows you to do predictive surveys. This means that you can simulate certain scenarios to see what might be needed of a WLAN and how you could possibly improve different scenarios[sim][1]. This would be a great feature for our specific scenario, seeing as rough simulations and calculations will make the job a lot easier for the testers. These types of services could be used to estimate the coverage and throughput in every location of the testing area. This is without you actually having to conduct the tests at the physical locations. The pro versions also have active throughput survey features.

2.4.2 Netspot

Netspot is also a site survey tool that displays WLAN quality using heatmaps[neta]. This tool is also very easy to understand and simple to install. The free version of the tool works on both Windows and OSX. However, the paid versions are currently

[1]It should be mentioned that you cannot solely rely on these capabilities seeing as they cannot account for all unknown variables that are present in the real world scenario

only available on OSX. There is one pro version of the tool that costs $149 and an enterprise version that costs $499[netb].

As with the Ekahau-heatmapper, this tool is very easy to understand and does not really require any instructions for you to be able to create heatmaps. Netspot does require you to mark two points on you map and tell the tool the approximate distance between these points. Once you have marked the real world distance between these two points, you can start to physically roam the testing area. When moving around the testing area, the tool will require you to stop at every location you want to collect signal data ("data points") from. Each measurements takes 5-10 seconds. This iterative process forces the testers to use longer time in an area than what you would do using a "continuous" process.

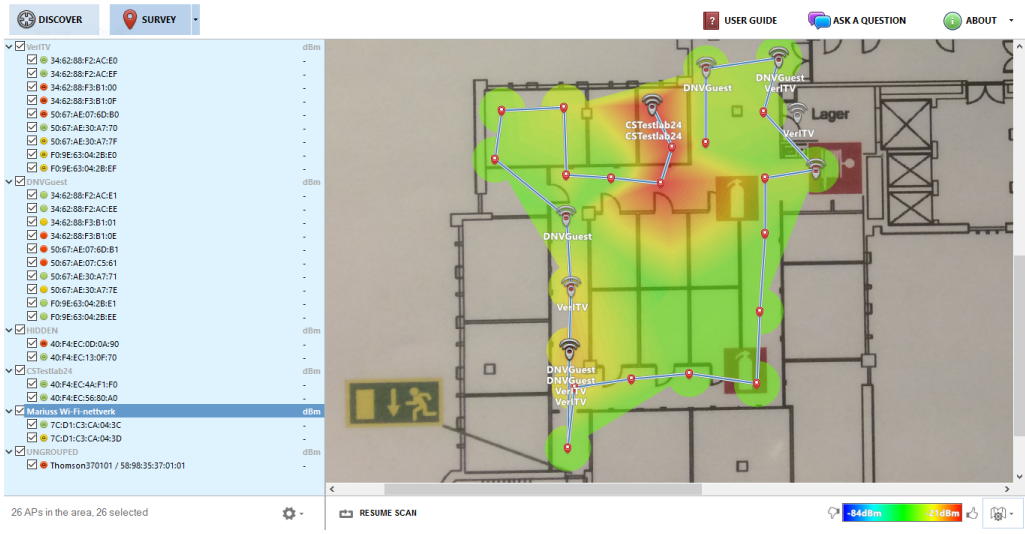

Figure 2.2: Heatmap generated using Netspot

The heatmap view has a tab where all APs are listed. They are automatically grouped together with the rest of the APs in its WLAN. This way you can easily choose which WLAN you wish to create heatmaps of. The tool stores all the data it has gathered while you walked around the test area. It can therefore render itself based on what you wish to view. This that you do not need to do several tours around the area to test every single WLAN. Neither the Ekahau-heatmapper or Netspot will automatically detect walls from the floor-plans. As you can see from figure 2.2, the tool will not estimate the signal propagation far outside the outer data points. However, it will estimate the area that exists between data points. Netspot will place APs on the heatmap based on where the signal is strongest. Other tools I

have tested try to estimate the exact location of APs, even though they might be placed far outside the outer bounds of the target area. Some of these calculations have not proven to be very reliable and with very weird results. Netspot places the APs where the signal is strongest inside the testing area. The tool also allows you to choose to only show APs on the map that has registered signals strengths stronger than X dbm. You can also choose which range of signal strengths the heatmap should represent. Figure 2.2 has quite a wide scale, where red is -21 dbm and dark blue is -84 dbm. Netspot also provides you with a so called "discover" mode. This is a separate view that contains information about all of the APs that the tool has detected since it started. This view provides you with a lot of general information about each AP, but more importantly it gives you a lot of signaling data from your current location.

The free version on Windows will render the heatmap based on signal strength. The professional versions can use several different types of data to create heatmaps. It can for example create heatmaps based on signal-to-interference levels, noise floor, how much nearby channel overlapping there is, where the different frequency bands have coverage, download speeds, upload speeds and so on. The pro version also has other additional features such as functionality for doing test over multiple floors and patching the results together. The pro version also has active throughput survey features. It's worth mentioning that the free version on OSX have a lot of limitations compared to the Windows version. The OSX version will for example only show up to 5 APs in one heatmap and only 50 data points per project. Overall the free Windows version is the best free site survey tool I have tested. This is because it gives you a lot of good features for doing a passive survey of WLAN coverage.

2.4.3 InSSIDer

The most important information that the heatmaps Netspot and Ekahau provide us with, is information about areas that do not have sufficient WLAN coverage (critical areas). However, we do not get enough information about the situation in an area by looking at a signal strength heatmap. You should get more information about critical areas. This is where InSSIDer is a useful tool.

InSSIDer is a wifi troubleshooting and optimalization tool made by Metageek[InS]. It gives you a lot of detailed information about signals from all the APs in the area. There are different versions of the tool. The prices ranges used to start at $0 with the Home edition and range to $149 for the office edition. This project used the home edition which is no longer available. However, you will now have to pay at least $19.99. Metageek also has a USB spectrum analyzer called Wi-spy. This tool gives InSSIDer the opportunity to analyze all RF-signals in the 2,4 GHz and 5 GHz band. This means that it provides you with valuable information about the amount of

activity at specific frequencies. It also makes it easier to locate sources of interference that does not originate from wifi equipment. You have two different versions of the Wi-spy. The mini version only allows you to analyze the 2,4 GHz band and adds $100 to the price of InSSIDer Office. The DBx version of the tool can analyze both the 2,4 GHz and 5 GHz band and adds $500 to the price of InSSIDer Office.

The Home version used in this project is a very good tool, although it does not give you all the same features as the Office version. The tool has one main window as you can see in figure 2.3. This window has four different parts. In the top left you will get general information about all the APs in the area (SSID, signal strength, channel, MAC-address and 802.11 version). Selecting different APs in this area will highlight different information in the rest of the tool. The top right shows you the signal strength over time of the selected APs. It also shows you how many co-channel[2] and adjacent channel[3] APs that the selected AP has. This part also shows a so-called link score. This is the tools way of indicating how good a possible connection to this AP would be. The closer this link score is to 100 the better it is. The bottom left shows a graph that illustrates the channels the APs are using in the 2,4 GHz band. It also shows the signal strength they have in this area. The bottom right shows the same information for the 5 GHz band. On the top of the tool you also have a bar for filtering different APs. If you select an AP, the tool might also display a pop-up bar, that suggests which channel this AP should use to get a better link score. Having this specific information is great for understanding the situation in a specific location.

[2]Co-channel APs are APs that use the same channel frequencies to communicate

[3]Adjacent channel APs are APs that do not use the same channels to communicate, but still use channels with overlapping frequencies

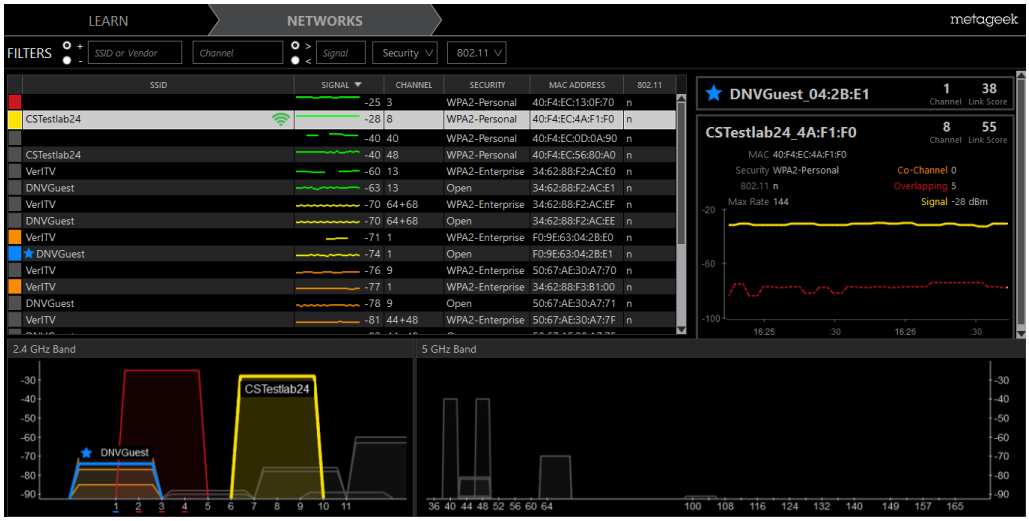

Figure 2.3: Tool layout for InSSIDer Home

Chapter 3
Related Work

This chapter contains a description of work that has been done within the fields of network quality and security. This work has affected the the results in this project.

3.1 Network Development and Site Surveys

Cisco is a major actor when it comes to network design and deployment. They therefore have a lot of experience when it comes to site surveys. Site Survey Guidelines for WLAN Deployment[Cis13] contains some basic guidelines for how to do different types of site surveys and suggestions for tools that may be used. It also has simple checklists for the most important things to remember when doing a site survey. However, it does not go very deep into each aspect of site surveys. Wireless LAN Design Guide for High Density Client Environments in Higher Education[Floa] is a more specific article. As the title suggests, it describes how to overcome the challenges of designing and deploying high density networks. This article goes through the different stages of network planning and implementation. It especially goes into details on how to increase the overall throughput in high density WLANs. Both of these articles has inspired the WLAN quality testing procedure in this project.

Other good sources of information and guidelines for how to perform site surveys can be gathered from the companies that actually create site survey tools. I have primarily used the informational pages from the companies that create InSSIDer[met], Netspot[neta], Ekahau[ekaa] and Tamographs[tam]. These companies have a lot of great guidelines for how to use their tools and how to perform site surveys in general.

3.2 PCI Data Security Standard (PCI DSS)

The Penetration Test Guidance Special Interest Group PCI Security Standards Council has created an information supplement to their Data security standard. This information supplement is a penetration testing guidance[Cou15a]. This document is,

among other, intended for companies that specializes in offering penetration testing services. The document focuses on four different things.

1. Penetration testing components
2. Qualifications of a Penetration tester
3. Penetration testing methodologies
4. Penetration testing report guidelines

The two parts that has played the biggest part during this project is the penetration testing components and penetration testing methodologies.

When it comes to the penetration testing components, the document describes all the different parts that goes into a complete penetration test. Testing scope is the first component. According to PCI a complete penetration test should check the security of the people, processes, and technology that store, process, or transmit sensitive data. The entities that fit this description are said to be inside the Cardholder Data Environment (CDE). A penetration test should test all surfaces that may affect a CDE, whether these are attack surfaces that are available to the public (external perimeter) or only available inside of the internal network (internal perimeter). PCI says that the engagement part of a pentest consist of four different components.

1. Application-layer testing
2. Network-layer testing
3. Segmentation testing
4. Social engineering

The two components that this project focuses on are the network-layer testing and segmentation testing. The network-layer testing should reveal bad configurations or old software. Segmentation testing checks that all segmentation controls are functioning properly and does not allow any entities outside of the testing scope to access the CDEs.

The methodology that PCI defines has also been of great use to this project. PCI divides the pentesting methodology into three different parts. These parts describe what steps a complete pentesting procedure should contain. First we have the pre-engagements work. This part of the process is where you among other things define the scope, success criteria and rules of engagement. The second stage

of a pentest is the engagement. This is the part that performs application-layer, network-layer, segmentation and possibly social engineering testing. The final part of a penetration test is the post-engagement. This is used to summarize the results and check for possible ways to fix the vulnerabilities. The post-engagement could also include possible retesting of vulnerabilities once the organization has had a chance to implement countermeasures. This part should also include a clean-up procedure to remove any malware or fix any damage that may have been caused.

3.3 Siemens Simatic s7-300

This project has used Siemens s7-300 Programmable Logical Controllers (PLCs). These PLCs have been an important component for simulating realistic industrial environments. These types of PLCs are widely spread throughout the industry and are used to control critical systems and machinery. In this project, they are used to control the most critical parts of our test setup: a miniature drilling-rig. Exploiting these PLCs have therefore been the end goal of all attacks. This project utilizes known vulnerabilities about this specific type of PLCs.

There are several well known vulnerabilities associated with these PLCs. Exploiting Siemens Simatic S7 PLCs[Ber11], is an article that was prepared for Black Hat USA 2011. The attacks used in this paper utilize the fact that data is transferred unencrypted in many Siemens systems. The paper argues that replay-attacks are especially dangerous for these types of systems. This is due to the fact that attackers can listen and register actual messages sent between PLCs. These messages can then be used to insert instructions to the PLCs. Having a large enough arsenal of messages means that you can basically control the PLCs (and the equipment they control) as you want. There has also been done work at NTNU that has led to the discovery of vulnerabilities in s7-300 PLCs. Finding vulnerabilities in offshore networked control systems[Sol15] is one of these articles that was written by Amund Bauck Sole in 2015. This project used fuzzing to find a specific DOS attack. The attack uses a specific set of messages in the PLCs payload to crash all involved PLCs. This exploit has proved very successful in this project. Testing communication robustness in networked control systems[Ohm16] is another project at NTNU that discovered a vulnerability with the s7-300 PLCs. The vulnerability was part of a previous research project I have conducted. The previous research focused on creating a tool for finding vulnerabilities in different control systems. This tool discovered a DOS attack that exploits a vulnerability in the s7-300s transport layer. This attack leaves PLCs unable to continue messaging each other. This exploit has also been used during this project.

Chapter 4
WLAN Scope & Requirements

This chapter describes how you can set the scope for the testing procedure. It also has guidelines for figuring out what requirements client have or should have for their WLANs. Defining the WLAN testing scope and requirements is something that needs to take place before any testing actually occurs. The chapter first talks about how to define the scope and requirements for WLAN quality testing before it moves over to the security aspect.

4.1 WLAN Quality Scope

There are two main attributes that we wish to focus on when it comes to WLAN quality:

1. **Throughput**
   Each AP in a WLAN will have a maximum possible throughput. This is often referred to as the APs bandwidth. This is the data rate that an AP can send/receive under optimal circumstances. The actual throughput describes the data rate that an AP will actually manage to deliver successfully. Knowing the actual throughput a WLAN has in different parts of the testing area will tell us if the users can actually send/receive enough data to support their intended network use.

2. **Coverage**
   A WLANs coverage is the area where devices can be located whilst using said WLAN. Good or bad coverage quality can affect a devices ability to communicate properly with an AP, without having to much corrupted or lost data. Really bad coverage will lead to devices being unable to connect to the WLAN.

The amount of throughput a WLAN should provide, and which areas the WLANs should cover, depends on how the clients intend to use their WLAN. The clients will therefore need to be interviewed to find out what they actually need.

4.1.1 Throughput

The first thing that interviewers need to ask the clients is what kind of services and applications they want their network to support. This might be everything from allowing the crew to stream videos, to sending simple signal data. Testers therefore need to know how much throughput different applications require to run properly. Specific applications will have their own requirements. This means that testers need to check each applications recommendations to know what they actually require. Netflix recommends at least 0,5 Mbit/s to simply connect to the site. It recommends 5 Mbit/s for HD quality and even a total of 25 Mbit/s for 4K video.

If testers want a more general idea of what different types of service/applications require then they can look at the throughput estimations Cisco has made. These estimations are based on their work in high density environments. Cisco is one of the leading companies in the area of complex WLANs. They have listed nominal throughput requirements for different types of services in their WLAN Design guide for High density client environment[Flo13], as can be seen in table 4.1. Note that these are only nominal values and may fluctuate a lot. They can however be used as an indication when trying to estimate the throughput needed in certain areas.

Table 4.1: Bandwidth requirements per Application

Application by Use Case	**Nominal throughput**
Web-casual	0.5 - 1 Mbit/s
Audio	0.1 - 1 Mbit/s
On-demand or streaming video	1 - 4 Mbit/s
Printing	1 Mbit/s
File sharing	1 - 8 Mbit/s
Online Testing	2 - 4 Mbit/s
Device backups	10 - 50 Mbit/s

It should be noted that accurate throughput estimations will require gathering data about the network usage over a longer period of time. The information interviewers get by simply asking the clients will only be rough estimations in comparison. If the clients does not have this kind of data available, then you could recommend that they invest in a network monitoring tool. You could also invest in more advanced site survey tools that has predictive survey capabilities. These types of tools will

most likely be able to use the information you get from the clients more accurately than what you can calculate yourself (see section 5.2). However, these throughput estimations will also be based on the clients estimations. They will therefore be less accurate than having gathered data about the actual network usage. Also, you will need to figure out if you think these tools are worth the investment, seeing as they are quite pricey (see section 2.4.1). You could also try to find the throughput in different areas by testing it yourself. However, these results may vary a lot based on different variables. Different hours and days of the week will have different throughput requirements. Testers will only have a limited amount of time to test throughput. This limited amount of time might therefore be a very bad representation of the overall situation.

If you do not have accurate network usage data, or more sophisticated tools, then you can consider estimating the throughput requirements yourself. You will need to do this by combining the number of people in an area and what type of services/applications the clients want to support. The throughput that is needed will vary from place to place. Different areas will need to support a different amount of people. Different areas may also use more or less of certain types of services. The actual throughput a user gets in an area is a product of bandwidth, signal strength, noise/interference and channel competition. A "critical area" is therefore (in this context) defined as an area where the sum of these factors might not provide the necessary throughput. These areas are usually areas with high user density, such as meeting or conference rooms.

4.1.2 Coverage

Coverage will also affect the quality of a WLAN. Coverage is a product of both signal strength and interference. Different types of network usage will need to provide different levels of coverage quality. You will therefore need to know which specific areas that should be covered by which WLANs and what coverage quality these areas should have.

Table 4.2 show some signal strengths that might be worth noting. These values are taken from Metageek, which is the company behind the InSSIDer tool (see section 2.4.3). These values represent a good guideline for what signal strengths you should have to be able to support certain services. However, as with throughput, you will need to check specific application recommendations for more detailed information.

4.1.3 Interference

A factor that can degrade both your throughput and coverage is interference. Interference comes in many forms, ranging from thermal background noise, to signals coming from other wifi devices. For example, a high level of interference may lead

Table 4.2: Acceptable signal strengths[Acc]

Signal Strength	**Signal quality**	**Description**	**Required for**
-30 dbm	Amazing	Max achievable signal strength. The client can only be a few feet from the AP to achieve this. Not typical or desirable in the real world.	N/A
-67 dbm	Very Good	Minimum signal strength for applications that require very reliable, timely delivery of data packets.	VoIP/VoWifi, streaming video
-70 dbm	Okay	Minimum signal strength for reliable packet delivery.	Email, web
-80 dbm	Not Good	Minimum signal strength for basic connectivity. Packet delivery may be unreliable.	N/A
-90 dbm	Unusable	Approaching or drowning in the noise floor. Any functionality is highly unlikely.	N/A

to an area having bad coverage even though you may have a good signal strength. Interference can lead to users being unable to connect to a WLAN, it can lead to lost/corrupted data or it can simply lead to inefficient network usage.

You should therefore have an idea of how the services the clients wish to use may be affected by interference. One way you could try to check how much the interference affects the communication is by using the Shannon-Hartley-theorem (equation 4.1 and 4.2[DMR13]). In equation 4.1 C represents the channel throughput, B is the bandwidth of the transmission channel, P_{signal} is the signal power of your AP and P_{noise} is the combined power of signal interference and background noise for this channel. These calculations could be used to do spot checks at different locations. However, representing the different types of interference that may affect 802.11 networks, in a proper way, may prove difficult[1].

$$C = Bln(1 + \frac{P_{signal}}{P_{noise}}) \tag{4.1}$$

$$Pnoise = P_{interference} + P_{background\_noise} \tag{4.2}$$

[1]You also have other things that needs to be considered in an 802.11 network

A more general guide for how good a SNR value is can be seen in table 4.3. However, if available, you should check the recommended values for the specific applications your client wants as different types of applications require different SNR to function properly. Cisco recommends a SNR of at least 25 dB and a signal strength of -67 dBm for their VoIP to function properly[Cis13].

Table 4.3: Quality of SNR in Wifi[Geib]

SNR	**Description**
40 dB	Excellent signal; always associated; lightening fast
25dB to 40dB	Very good signal; always associated; very fast.
15dB to 25dB	Low signal; always associated; usually fast.
10dB - 15dB	Very low signal; mostly associated; mostly slow.
5dB to 10dB	No signal; not associated; no go.

4.2 WLAN Security Scope

When it comes to WLAN security, we wish our test procedure to expose any vulnerabilities that the current system has and how these vulnerabilities can be exploited. All businesses have certain critical information and/or systems they want to protect from the outside world. In industrial environments there are often systems that operate critical and possibly dangerous equipment. These types of systems are often what clients would want to protect the most. If we use the PCI Data Security Standard (PCI DSS), then these types of systems would be inside what we call the Cardholder Data Environment (CDE) (see section 3.2). Enteties inside a CDE will in our case be devices that store, process, or transmit sensitive data. The scope of our pentesting should therefore be to try and compromise devices inside of the CDEs. The CDEs will need to be defined in the interview process. As with the PCI DSS, we want this procedure to test the perimeter of our CEDs[2]. This is in order to find possible ways of getting closer to the CDEs. The focus of this project is WLANs, which is why this is defined as the outer bounds of the CDEs perimeters. We will therefore need to find ways to compromise the CDEs by using the WLANs as the entry point to the rest of the system.

This pentesting procedure will mostly consist of network-layer and segmentation testing[Cou15a]. Network-layer testing will check that both the WLANs and internal network are configured in a secure way that does not leave devices open for exploitation. If devices are possible to exploit, then we want to test whether these exploits will lead to further access. Hopefully this may lead to us being able to

[2]In our pentesting procedure we define the perimeter as any device/user that may get us closer to accessing devices in the CDEs

compromise/exploit the CDEs. The segmentation testing will ensure that WLANs does not provide users/devices unnecessary access. The clients WLANs will most likely have different levels of access to the internal network. This means that each WLAN will have different security requirements. You should therefore ask the clients about which WLANs that should be allowed to access different parts of the internal network.

You will also need to set "rules of engagement" for the pentesting procedure[Cou15a]. This is an important aspect of any pentest, seeing as you want to have clear guidelines for what is legal and what is not. You will need to set how far the testers should actually go. The tests could restrict you to only scanning devices in the CDEs, but not exploiting them. Exploiting different devices may cause irreversible damage. Exploiting devices in the CDEs may therefore not always be an option. If you would still like to document exploits of crucial devices or equipment, then you can create a separate controlled environment. Documenting the effects such exploits may have can be important for the clients to see what an attack actually may do to their system. This is what we have done in chapter 6. You will also need to set special consideration for the types of environment we are working with. Industrial environments may have strict safety rules. This may mean that testers are only allowed to perform tests at certain times of the day.

Another relevant aspect that the PCI DSS recommends doing before actually testing, is to review previous findings. This may give you valuable insight into what you can expect and what type of previous work that has been done to secure the testing environment.

Chapter 5
WLAN Quality

In this chapter we will look at the testing environment used in the project, addressing different tools and how they can be used in the procedure, how the tools can be used to see how check quality requirements, and a description of different ways to test possible improvements. The chapter ends with a discussion of the experiences and results.

5.1 WLAN Quality Lab

All testing in this project has been conducted at DNV-GL's Trondheim office. Because of the many WLANs operating at this office, the environment located here is comparable to actual industrial environments. Realistically, industrial environments can consist of multiple WLANs operating in the same area. This testing environment has been used to test various tools that may be useful. It also played an important role in assessing how different changes may improve WLAN quality.

The interview process should have given you the information needed to figure out throughput and coverage requirements. Once you have set these requirements, then you will be ready to start testing. Testing WLAN quality will be done by performing a site survey. There are three main types of site surveys[Cis13]: passive, predictive and active survey. Passive site survey means that the testers will operate in a "listen-only" mode. A passive survey is good for testing signal coverage, checking down-link capabilities and finding rogue APs. We will mainly use this type of survey to check signal coverage (section 5.2). We can also do a predictive survey in this procedure[1]. Predictive surveys are good for testing AP placements/simulations based on the current environment and requirements. Tools that have these capabilities can calculate the best way to add or move APs, and calculate how the environment will react to these changes. Active surveys are cases where you connect to the network

[1]However, this depends on which site survey tool you choose (see section 5.2.1)

and actually insert data to see how the network performs. These types of tests could be used to test throughput.

All site surveys are done under different circumstances. Therefore, each site survey should be customized to fit the specific situation. Our site survey especially needs to consider the very limited time frame, and the fact that these WLANs are located in industrial environments.

5.2 Site Survey Tools

There are a lot of different tools for performing site surveys. One of the most useful features that site survey tools have are heatmaps. Heatmaps are basically a way of visualizing data using colors. The reason why heatmaps are well suited for testing WLAN quality, is that it takes a lot of raw measurement data and transforms it into simple and understandable information. This information is also fairly easy for clients to understand, even though they might not be too knowledgeable about computer networking. Even though they are simple, they also make great use of all the raw data to intelligently provide you with detailed information as well. Heatmaps has become a vital part of any project involving site surveys and wireless network design. This is evident in the Cisco Guidelines for WLAN Site Survey[Cis13] and Network Design[Floa].

The heatmaps that site survey tools generate are used to map signaling data to a physical location. The tester will therefore need to indicate the current location while the heatmap-tool gathers signaling data. The process of gathering data could be automated by using GPS-signals and mapping out a WLAN using wardriving. Because a lot of the testing have to be done indoors, the GPS-coverage will not be as good. You will therefore need some sort of map to indicating the testers location. This could be a blueprint provided by the clients, or it could be a simple photo of the floor-plan. I have used a floor-plan photo during my tests, which I have found to work quite well. Now, if you have a map of the target area, you can use this for your heatmaps.

The heatmap tools that has been used the most during this project are Netspot and Ekahau (See section 2.4.2 and 2.4.1). What is important to notice about the tools that were used, is that they are all free versions of the tools. This means that the paid versions have additional features. The conclusions made about the paid versions are therefore based on research. The pricey tools will have more features than the cheaper/free ones. This is where you will have to evaluate if the trade-off in features and price is worth the investment.

5.2.1 Ekahau vs Netspot

Netspot is definitely the best free version of any heatmapping tool I have seen. This is because this is the free version tool with the most features. It definitely beats the Ekahau-Heatmapper. The tool provides a whole lot of important features, and can be used to give good results when it comes to testing the WLAN coverage. The results would however improve dramatically if you invested in one of the pro versions. These versions are not very expensive compared to the competitors. The main advantage of the pro versions is that you can use a total of 15 different heatmap visualizations, while the free version only has 1 (signal strength)[2]. These extra visualizations will make it easier for you to assess the WLAN quality. The pro versions of Ekahau provides even more features. Most importantly it gives you predictive survey capabilities. This makes it easier to perform capacity planning and AP simulations. However, Ekahau is a lot more expensive than Netspot. I do however suggest that you at least invest in the Netspot pro version that costs $149, seeing as this investment will give you a lot for the money. The heatmap illustrations in this chapter are done using the free Windows version of Netspot.

5.2.2 InSSIDer

Another good tool that you should definitely consider using is InSSIDer (see section 2.4.3). This site survey tool will not provide you with any heatmaps. It's therefore not great at displaying WLAN quality over an entire area at once. However, it's great for gathering more specific signaling information about your current location. It's very good to use when you want to identify the root of a problem. You can for examaple use it to figure out a optimal channel selection for your APs(see section 5.5).

5.3 Performing Site Surveys

Performing a site survey should try to reveal areas that does not meet the clients requirements. As mentioned, heatmaps are a great way of representing the different WLANs situation. However, there are certain things you should remember in order to create a proper heatmap. First of all, you will have to get a proper map that is made to scale. Having incorrect scales will make the tools unable to properly calculate signal propagation. You should try and add as many details to the map (floor number, walls, source of interference etc.), as this will make your results more accurate. This is provided that the tool you are using can handle this type of information[3].

[2]Note that Netspot is rather new. The tool does therefore get a lot of updates. The 15 visualizations are therefore what is available at the time of writing

[3]Free versions of Netspot and Ekahau does not have these capabilities

5.3.1 Gathering data for a heatmap

Data gathered for our heatmaps are only done as a passive survey. This is due to the limitations of the free Windows version of Netspot. Figure 5.1 shows the layout of the DNVGuest WLAN. As you can see, the building is divided into Block A, Block B and Block C. Block A has one AP (AP 1). Block B has two APs (AP 2 and AP 3). Block C has one AP (AP 4). It's worth noting that not all rooms were accessible during this project. Data has therefore not been gathered from all areas of the building. All rooms that do not have "data points" inside them are therefore outside of the target area[4]. There are four important things to remember when gathering data for a heatmap:

1. First, you should try and gather as much data as possible. If you are using Netspot, this means that you should stop as many times as possible to create a data point. Measured data is more accurate than calculated data. The site survey tools will try to calculate the signal levels inside the entire testing area. These calculations will take into account all the information you have given it. However, some information will always be unaccounted for or misrepresented. These calculations will therefore always be less accurate than actual raw data collected from different locations.

2. You should always try to cover as much of the testing area as possible. This is to avoid having larger areas of a heatmap, where the signal strength is only based on the tools calculations. The bigger the distance is between each datapoint, the more inaccurate the calculations will become. As you can see from figure 5.1, the data points are spread out evenly throughout the area.

3. You should always take measurements from the outer bounds of the testing area (especially if you are using Netspot). As you can see from figure 5.1, some of the rooms at the bottom of the figure are not included in the heatmap. Netspot will only try to calculate signal propagation between data points. This means that all areas, that are outside the outer data points, will not be taken into account.

4. You should always try and measure data inside each room in the target area. This is especially important if the tool does not know where the walls are located. Walls are a source of signal degradation. If the tool does not know where the walls are located, then it will not be able to do an accurate calculation of signal propagation. Some of the rooms with restricted access does not have data points inside of them. This means that the signal information will be less than optimal in these locations.

[4]This may also happen in a real world scenario, seeing as certain areas may be off limits

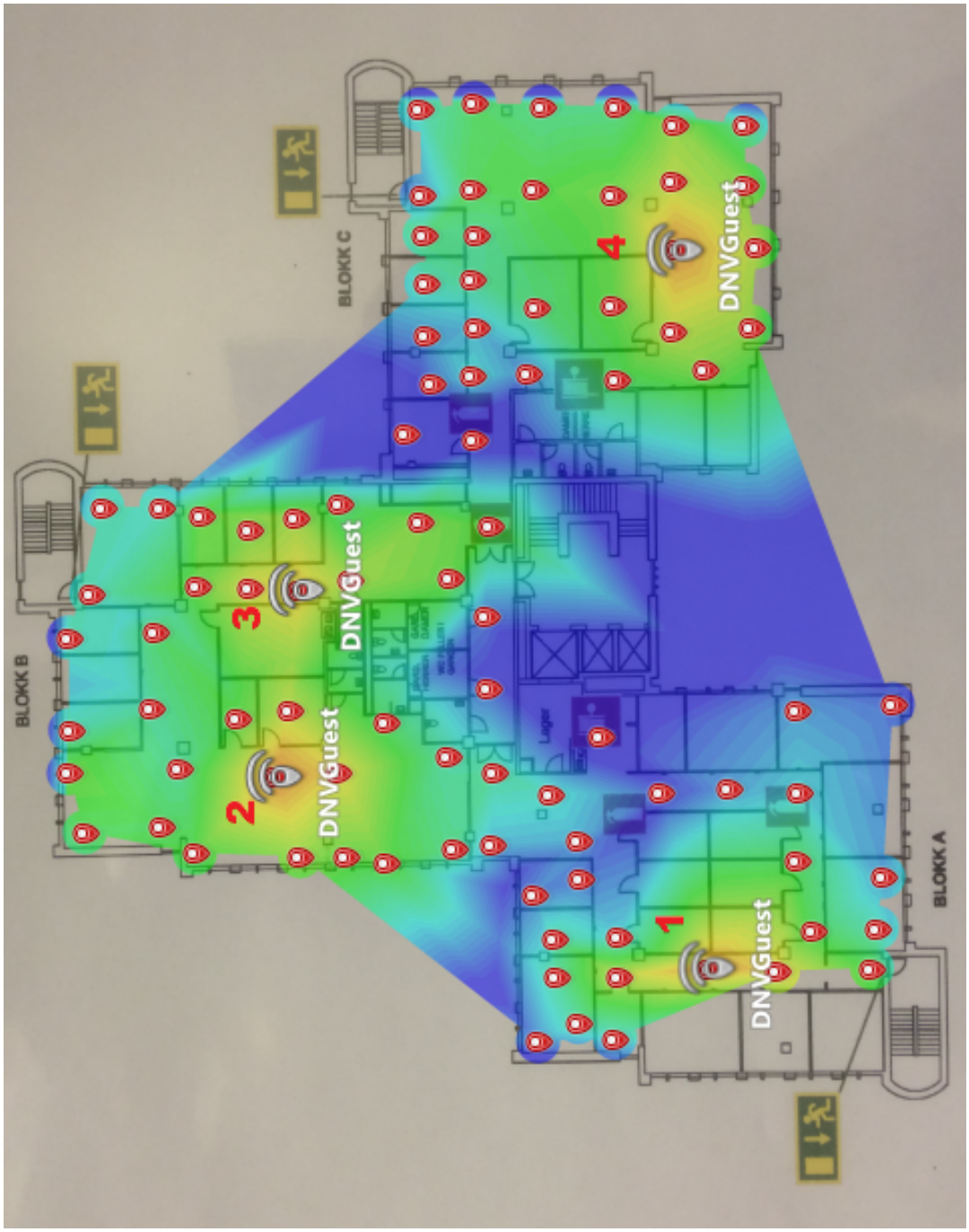

Figure 5.1: Heatmap Of entire WLAN quality lab. Dark blue indicates signal strengths lower than -67 dbm

5.3.2 Checking signal coverage and throughput

Once you have gathered enough data from the testing area, you will need to edit the heatmap. Editing the heatmap correctly will help you find out where the WLANs does not meet the requirements. Figure 5.1 is configured to show any areas that cannot support services like streaming video and VoIP based on signal strength. First off all I have only selected the APs for the WLAN that we are testing. All APs in our target network has DNVGuest as their SSID. As you can see from table 4.2, you should at least have a signal strength of -67 dbm for these types of services to function properly. I have therefore set the lower end of the color scale to -68 dbm. This means that all of the dark blue areas do not provide sufficient signal coverage for this WLAN. The areas that does not meet the signal requirements can be marked as critical areas. The heatmap Netspot creates is interactive. When the cursor hovers over one spot on the heatmap, all the APs will automatically show their signal strength at this location. These values will automatically update as you move your cursor. You can also click at specific areas on the map which will bring out a small window. This window contains a list of all the APs that can be reached in this area. They are all sorted from the strongest to the weakest signal. These features can provide you with more exact data than the colors on the heatmap will. It's important to know that test results may vary from time to time. Different tests done in the same environment will often show slightly different results. This is because we are dealing with a number of variables that may change slightly from time to time. These slight variations should be expected, which means that signal values should be given a slight margin of error. You should therefore do spot checks using InSSIDer to double check the situation in the critical areas.

After checking the signal strength on the heatmap, you should try to find out if there are any areas that have a lot of interference. Interference could be anything from thermal background noise to interfering wifi signals.The signal strength only tells you how strong the signal from your AP is. It does not tell you how strong it's compared to other signals. An area with decent signal strength can still have insufficient coverage if there is a lot of interference. A capacity that you have calculated to provide sufficient throughput, may prove to be insufficient due to interference. If you have a pro version of Netspot or Ekahau, then you can create heatmaps that visualizes how your WLAN is affected by such quantities. For example, one of the predictive capabilities of Ekahau is called capacity planning[cap]. This feature can tell you the estimated throughput in an area based on parameters that you have provided the tool. It can also use data that the tool has gathered itself. I highly recommend that you check out these pro versions. They will most likely make your throughput calculations more accurate. Doing these calculations by hand is extremely demanding and time consuming for an entire testing area. This is time and manpower you don't necessarily have in this testing environment and the pro

versions alleviates this excess cost. You will have to consider AP bandwidth, AP configurations, co-channel interference, adjacent channel interference, background noise, signal strength, overlay and so on. You might therefore be better off by simply checking that the capacity in an area is well within the throughput requirements and check if there are any major sources of interference (in which case you should try to remove or minimize this interference). The capacity can be found by checking the total amount of bandwidth that is provided in an area. You will therefore need to check the configurations of the APs that actually covers an area. It's not enough to simply check the amount of bandwidth a certain AP can provide. This will only tell you what data rates the AP can support. The configuration on the other hand, will tell you which data rates that are actually enabled.

5.4 Non-Wifi Interference

By creating heatmaps, you will get a good grasp of the situation in the different areas. However, you may still be unaware of how interference may affect a WLAN. You should therefore do a more detailed investigation. More detailed research may show that areas are better or worse then you initially thought. You can also find out how the clients could possibly improve their WLANs. You should keep in mind that you cannot treat throughput and coverage as a single problem. Coverage, throughput and interference are all part of the same problem.

The first thing you should do is to fire up InSSIDer (preferably with the Wi-Spy). You should then do a quick tour around the places you have marked down as critical areas. The main thing you want to check out is if there are any non-wifi equipment that may cause a lot of interference for the WLANs. This type of noise can affect both the coverage and throughput in an area. This is because it may cause data loss/corruption and connectivity issues. A Wi-Spy will be able to show you any active sources of interference. You will also have to keep an eye out for sources of interference that may be inactive. Typical sources of interference are cordless phones, audio systems, wireless video camera and microwaves. In this type of environment there may also be some industrial equipment that creates interference. If you expect that something might cause interference, then check it out. When you find a source of interference, then you should check if the clients actually need this equipment. If not, then they should remove it. Other equipment may be moved to more strategic locations. Areas where there is high channel utilization will be more affected by sources of interference. You may therefore consider moving the equipment responsible for the interference to an area that will not suffer greatly from some loss of data. Interference from non-wifi equipment was not a problem at DNV-GLs offices. The only source of interference was a microwave in the kitchen, which was rarely used. The overall background noise was generally considered insignificant for both the 2,4 GHz and 5 GHz band. It was not strong enough to cause any significant packet

loss/corruption and was not strong enough for any devices to interpret any channels as busy. Vendors will usually not consider a channel as busy unless non-wifi interference reaches energies between -65 to -72 dbm. This depends on the 802.11 version being used (see section 5.5.2 for more info on Clear Channel Assessment)[Kru][Uni][Flob].

5.5 Wifi Interference

Wifi devices may cause two types of interference. They can create co-channel interference. This comes from two or more APs using the same channel. There may also be adjacent channel interference. This comes from APs using overlapping channels. As with interference from non-wifi equipment, these types of interference will also affect your coverage and throughput in a negative manner.

5.5.1 Adjacent channel interference vs. Co-channel interference

Data being sent on adjacent channels use overlapping frequencies for data transmissions. This leads to a lot of colliding transmissions which again leads to data loss/corruption. This can severely damage an APs throughput and coverage; and it is the main reason that adjacent channel interference is worse than co-channel interference. Even though co-channel interference is better than adjacent channel interference, it is not preferable either[adj]. This is because 802.11 networks operate in a polite manner with other devices using the same channel. All devices using the same channel will cooperate and share the transport medium. We therefore avoid having devices screaming over each other trying to get through. The devices will instead take turns talking over the network. This means that we get less data loss/corruption. You should therefore always select non-overlapping channels. Both types of wifi interference are usually quite easy to avoid in the 5 GHz band. This is because the 5 GHz band contains a lot more channels than what the 2,4 GHz band has. There are a total of 14 channels defined for use in the 2,4 GHz band[Poo]. However, some parts of the world do not allow all 14 channels to be used by the public. North-America only uses 11 channels while most of Europe use 13. This is a fact that you need to be aware of when working with industrial environments such as ships. These types of environments may move between areas with different laws. I have therefore used 11 channels in this project. The channels used here are not illegal in other parts of the world where a ship may dock. Although there are 11 channels in the 2,4 GHz band, there are only 3 non-overlapping channels. These are channel 1, 6 and 11. You should only use these 3 channels in the 2,4 GHz band, seeing as we always want to avoid adjacent channel interference[5]. This is often a problem that clients may not be aware of when deploying their network on the 2,4 GHz band. This problem is a lot smaller when using the 5 GHz band because of the

[5]You can modify the procedure to use channels 1, 5, 9 and 13, if the clients wishes to use 13 channels

increased number of non-overlapping channels. For example, North-America has a total of 24 non-overlapping channels[wif]. The channel selection strategies used in this project primarily focus on the 2,4 GHz band. However, the similar strategies will apply to the 5 GHz band. Many APs are deployed using the default settings. The channel selection at DNV-GLs offices are a great example of this. The test WLAN used, did not seem to have been set up with any regards to channel selection. Figure 5.2 is taken from one of the meeting rooms at DNV-GLs Trondheim offices. This shows how different APs are overlapping with each other. Figure 5.3 shows the channel selections taken from a location at the NTNU campus. This shows that all of the school administered APs are operating on either channel 1, 6 or 11[6]. As you can see from figure 5.2 this is an area that is already struggling with low signal strength. The situation then becomes quite bad when you also consider the data loss/corruption that comes with adjacent channel interference. Devices may therefore have difficulties operating properly in this area.

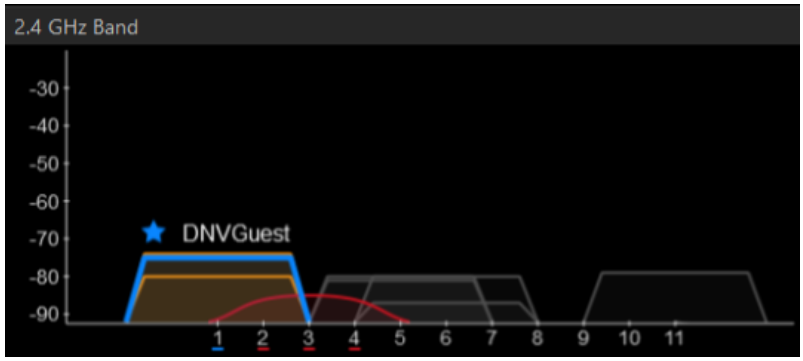

Figure 5.2: 2.4 GHz band at DNV-GLs Trondheim office.

Removing all adjacent channel interference will most likely lead to an increase in co-channel interference. This is because we limit the number of channels that APs allow to use, thereby making them share channels. You should therefore also know how to decrease and possibly remove this type of interference.

5.5.2 Channel selection

Which APs that gets which non-overlapping channels is not always obvious in complex WLANs. If the channels are not selected strategically, then this may lead to a lot of unnecessary co-channel interference. The idea is that you want to keep APs using the same channel as far away from each other as possible, without creating coverage issues. In an optimal solution you will have cells that create a honeycomb diagram.

[6]There is one AP that is not part of any of NTNUs official networks which operates on both channel 2 and 6

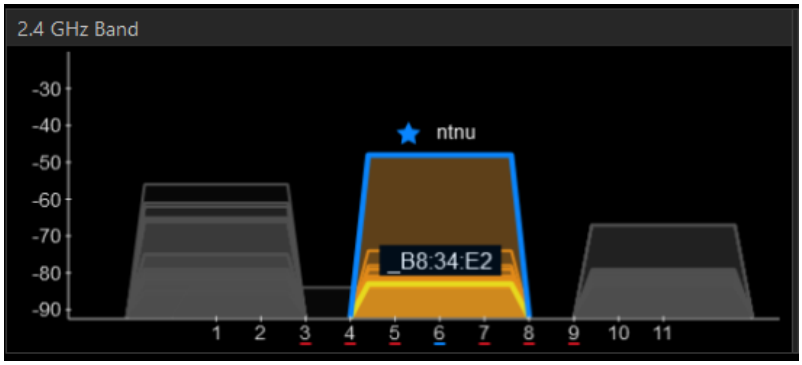

Figure 5.3: 2.4 GHz band at the NTNU campus

Each of the cells should partially overlap with its neighbors to provide good coverage. All overlapping cells should be using different channels. A good example of this can be seen in figure 5.4[netc]. Note that you can also design a network like this where you mix in 5 GHz cells. If the clients do not require all areas to have 2,4 GHz coverage, then the task could be a lot easier.

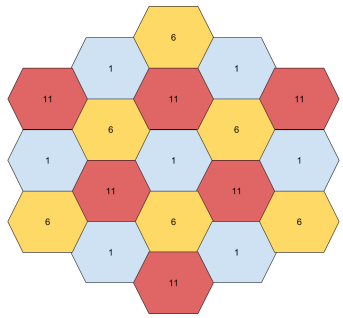

Figure 5.4: Optimal channel selection and AP placement

Creating a network that looks like this might be difficult in real world scenarios. There are many factors about each environment that you cannot control. A simple thing like having a WLAN over several different floors can make this task a lot more complex. However, if you were able to create a similar scenario, then you could avoid both adjacent channel interference and co-channel interference. The APs would no longer need to share the channel capacity, which means we will get a lot closer to using the APs maximum possible throughput. In a real life scenario the cells will not be so clearly defined as they are in figure 5.4. As you can see when using any of your site survey tools, the signal strength will gradually dissipate. How far same channel APs should be placed away from one another to avoid co-channel

interference depends on the Clear Channel Assessment (CCA). CCA is one of two wifi functions for carrier sensing. CCA is responsible for detecting energy and wifi signals on the radio interface. If the CCA is triggered in a wifi device, then this device will mark the channel as busy and will wait its turn. CCA can be triggered in two ways. The first way is through carrier sensing. Here the device is able to detect and decode a wifi signal on the used channel. The other way is if the device detects a certain amount of energy on the channel[Uni]. Both of these triggers have certain thresholds for how weak a signal/energy should be to actually mark the channel as busy. Devices may vary a bit depending on vendor. The standards for 802.11 a/g/n/ac require all equipment to at least be able to decode signals that has a strength of -82 dbm[Kru]. Other 802.11 versions requires equipment to be able to decode signals at -80 dbm[Uni]. The energy of a signal that a device is not able to decode will have to be a lot stronger than this. Most modern devices can decode wifi signals that are weaker than these suggested thresholds. Some go all the way down to -95 dbm. All devices that can decode a wifi signal will not transmit themselves. This means that the effective cell size around all devices using a channel is quite large. It can therefore be quite difficult to create separate "bubbles" around each AP that does not affect any devices using another AP on the same channel[7]. Moving or minimizing the cells of APs using the same channel may diminish the co-channel interference for devices that are located in overlapping cells. On the other hand this might impact the coverage in a negative way. Keep in mind that higher data rates are more difficult to decode than slower data rates. This means that disabling slower data rates will effectively decrease the cell size. However, older equipment may not be able to use these higher data rates and may therefore be unable to use the WLAN.

Be aware that you should not try to assign channels, to different APs, before you have figured out if you are going to add new APs. Adding new APs may change the situation completely and will therefore make any channel selections you have done pointless. It's however really important to keep this in mind when you are improving coverage and throughput.

If we for a second say that the capacity and signal strength in figure 5.1 does provide sufficient coverage and capacity, but is suffering due to poor channel selection. Then we do not have to move or add any APs to the environment. In this case we only want to optimize the channel selection of the current situation. To do this we first need to figure out what channels the different APs were using. All the APs were in this case using the 2,4 GHz band. AP 1 was using channel 11, AP 2 was using channel 9, AP 3 was using channel 1 and AP 4 was using channel 1. AP 3 and AP 4 are close enough together to create co-channel interference. Seeing as Norway allows the use of 13 channels in the 2,4 GHz band, we could give each AP their

[7]Note that each device will create this type of a bubble to. However, we cannot control where these devices are located. Besides, there are usually a lot more traffic down-link than up-link

own non-overlapping channel. However, for demonstration purposes we wish to use channel 1, 6 and 11. Therefore, two of the APs will have to use the same channel. AP 1 and AP 4 are the ones with the greatest distance between each other. They are therefore least likely to create co-channel interference. Channels with higher frequencies are the ones that are hardest to decode and are therefore less likely to trigger the CCA. We therefore set AP 1 and AP 4 to channel 11[8]. We can then give AP 2 channel 1 and AP 3 channel 6. The signal strengths will then need to be checked around the area to see if we have we managed to eliminate/reduce co-channel interference. This was done by walking around with InSSIDer, checking if there were areas where both of the APs could trigger the CCA of our computer. This could happen in the hallway connecting the different blocks. The signal strengths here of both AP 1 and AP 4 ranged from -80dbm to -90dbm. However, if there were devices using the WLAN from this area, then they would most likely be connected to either AP 2 or AP 3. This is because they had signal strengths from -55 dbm to -72 dbm in this area.

5.6 Improving Coverage

At this stage, you should be aware of the current situation the different WLANs are in. You may now consider improving the coverage in places where this might be needed. You should note that changes done to improve your WLANs coverage may also affect the throughput. Adjacent channel interference and non-wifi interference will affect your effective coverage area. You should therefore try to remove these sources of interference as much as possible (see section 5.4 and section 5.5). Keeping this in mind, you will have to visit all the critical coverage areas and assess the situation.

5.6.1 Optimizing AP placement

You should first see if you can better the coverage without adding new APs. There is no definitive answer for how you should do this. However, you should see if the current AP placement has objects blocking signal propagation. Objects like concrete, brick walls, refrigerators and metal objects can cause bad coverage in different areas.

If we look at Block C, we can see that certain meeting rooms do not have good enough coverage to support services such as VoIP. Some of these meeting rooms have signal strengths as low as -76dbm, when they should be at least -67dbm. This area should therefore be considered as a critical area. The position of AP 4 means that it is only able to cover parts of Block C. Netspot did in this case not place the AP accurately. The AP placement missed by approximately 2 meters. The AP is actually placed right next to a large concrete column. This concrete column is

[8]Note that this in reality would just make the coverage of these areas even worse than they already are

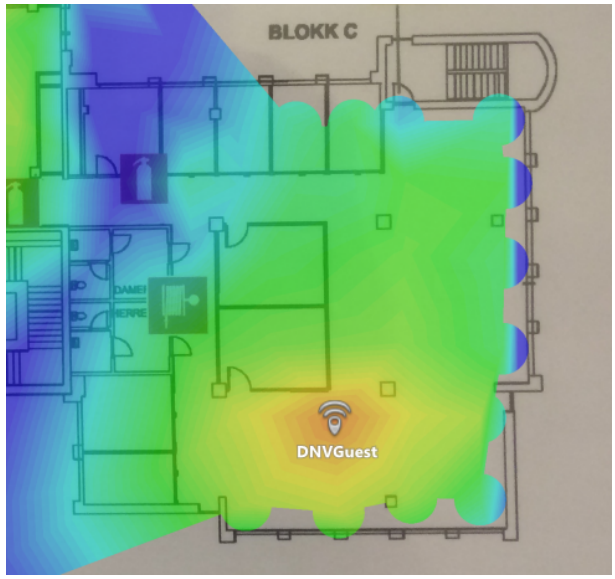

Figure 5.5: Initial coverage Block C. Dark blue indicates signal strengths weaker than -67 dbm

blocking a lot of the AP's propagation path. The propagation path is also blocked by a couple of meeting rooms. By moving it to a more central location in Block C, that is not close to any concrete, we can get the coverage shown in figure 5.6.

Moving AP 4 to this location will not make any significant changes to co-channel interference. We can still use the same channel selection as we chose in section 5.5.2, without any significant changes.

5.6.2 Transmission power

Another way you can improve the coverage area is by increasing the transmitting power. APs might not be using the maximum transmission power. You will therefore need to check their configuration. One thing that you should be aware of when increasing the transmitting power, is that it will increase the cell size for down-link transmissions. That does not necessarily mean that devices have sufficient transmission power to transmit data up-link. All APs in this project are by default set to transmit at maximum power. Increasing the transmission power is therefore not an option in this case.

Figure 5.6: Improved Coverage Block C. Dark blue indicates signal strengths weaker than -67 dbm

5.6.3 Adding APs

Adding new APs is another way of improving WLAN coverage. This is probably the best way of improving coverage if there are big "dead zones". Other improvements may be better if you want smaller changes. If you are not careful when adding new APs, you may end up doing more harm than good. Adding APs to a high density area, can make the adjacent channel and/or co-channel interference a lot worse for nearby APs. This is why it's important to first figure out if you need additional APs, before changing channels. Adding new APs will also mean that the clients have to invest in new equipment. You may want to consider some of the smaller changes before adding new APs.

If we again focus on Block C, then we can try to better the critical areas by adding a new AP. If we only considered the signal strength, then the best location for the new AP would be in the heart of the critical area. If we do this, our coverage will become quite a lot better (see figure 5.7).

The problem with this solution is that the new AP will create interference no matter which channel it selects. We therefore either have to move it to a more strategic location or reduce the transmission power (see section 5.7.4).

Figure 5.7: Additional AP placed in Block C. Dark blue indicates signal strengths weaker than -67 dbm

5.7 Improving Throughput

Improving the throughput of a WLAN will (as the coverage) depend on changes done to improve coverage and reducing interference. You will therefore also need to consider sections 5.4, 5.5 and 5.6 while you try to increase a WLANs capacity. You should by now have a grasp of which areas that might not meet the throughput requirements.

5.7.1 Increasing bandwidth

As with the coverage, there are a couple of methods to increase the throughput without adding new APs to a WLAN. One way to do this is by increasing the frequency band that an AP is using. You will have to check if the APs actually supports using multiple channels simultaneously and how much this actually will increase your capacity. Using multiple channels per AP would however not be recommended in most cases, especially not in the 2,4 GHz band. This is because it will most likely lead to co-channel interference and/or adjacent channel interference. Channel selection can then become quite difficult. However, you may consider increasing the frequency band for APs using the 5 GHz band. This is due to the fact that the 5 GHz band has a lot more non-overlapping channels. A lot of modern APs have the possibility of using both the 2,4 GHz and 5 GHz band simultaneously. Using both bands will also increase the total capacity in the area.

5.7.2 Disabling lower data rates

Another way of improving the throughput is by disabling lower data rates in the AP configurations. APs will use the lowest enabled data rate to transmit management frames[cis15]. This means that the management transmission will take up more of an AP's capacity than it would at higher data rates. In other cases, APs may not use its full potential since the highest data rates may be disabled. As mentioned in section 5.5, disabling lower data rates will reduce the effective cell range of an AP since higher data rates are more difficult to decode. This means that you should be vary of the coverage around the APs you wish to configure.

In this project, one of the APs that were used for testing was a Cisco Aironet 1142N. In this APs 2,4 GHz band[9], there are a total of 12 different data rates that can be used. These data rates range from 1 to 54 Mbit/s. However, the default settings on this type of AP only uses 1, 2, 5,5 or 11 Mbit/s to transmit regular traffic. The AP will try to use the best possible data rate that is enabled. Enabling the higher data rates can therefore boost the overall throughput. We could then disable the lower data rates. However, excluding lower data rates may exclude certain devices. Some equipment may only support lower data rates. Other equipment may not be able to use these data rates due to insufficient SNR at their location.

5.7.3 Equipment upgrades

A third way of increasing a WLANs capacity is by upgrading APs and/or equipment using the WLANs. WLANs may be using old or low quality equipment. Investing in newer and better APs gives you the possibility of using higher data rates. You could for example easily check if any of the APs are only using older 802.11 versions. They should optimally support both 2,4 and 5 GHz bands.

5.7.4 Add more APs

If you are unable to meet the throughput requirements with the network resources you have at hand, then you can try to add new APs. These APs will however become new sources of interference which means they might affect other APs throughput and coverage.

When we introduce a new AP into the environment we still want it to cause little adjacent- and co-channel interference. If we still assume that we only want to use channel 1, 6 and 11, then the AP's cell (area which can trigger CCA) should only overlap with two other APs. These two APs should have non-overlapping channels. The AP added in figure 5.7 will introduce co-channel interference. If we still want to have the additional AP, to get a higher total capacity for Block C, then we can try

[9]The same rules applies to the 5 GHz band

moving it. We want to move it far away from AP 2 but still provide coverage for all devices in Block C. If we place the new AP at the edge of Block C, then this AP will get the cell size shown in figure 5.8. Note that this heatmap is edited to show the area where it will trigger CCA. I have used -80 dbm as a general trigger point for CCA. The dark blue areas are therefore areas with signal strength lower than -80 dbm. The coverage in the meeting rooms are still quite good. They have signal strengths no lower than -64 dbm. If we then add the cell of AP 2 to the heatmap

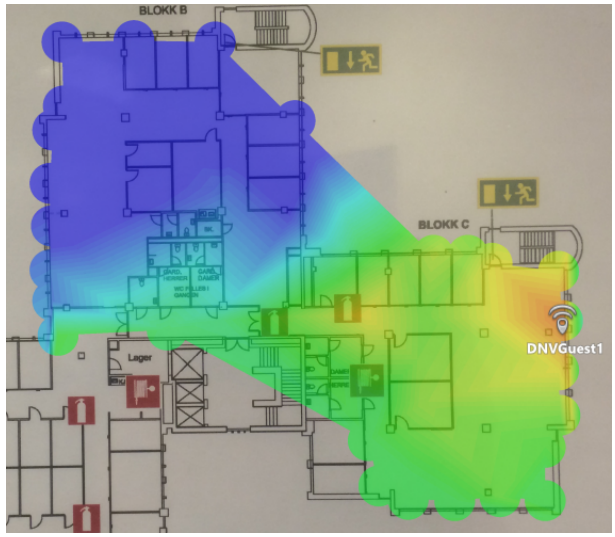

Figure 5.8: Cell size of new AP. Dark blue indicates signal strengts weaker than -80 dbm

(figure 5.9), we can see that the cells are partially overlapping. However, devices that are located in the overlapping areas will in most cases be handed over to AP 3. We will therefore in most cases avoid co-channel interference. The new AP is currently transmitting at full power. The overlap could therefore be reduced even further by reducing the transmission power.

5.8 Discussion of Results and Experiences

Finally, this chapter will summarize and discuss the results and experiences that were discovered during WLAN quality testing and research.

5.8.1 Things that can be left out of the procedure

What I have experienced when doing site surveys is that this is most definitely a time consuming task. By splitting the procedure, you can choose to offer the clients

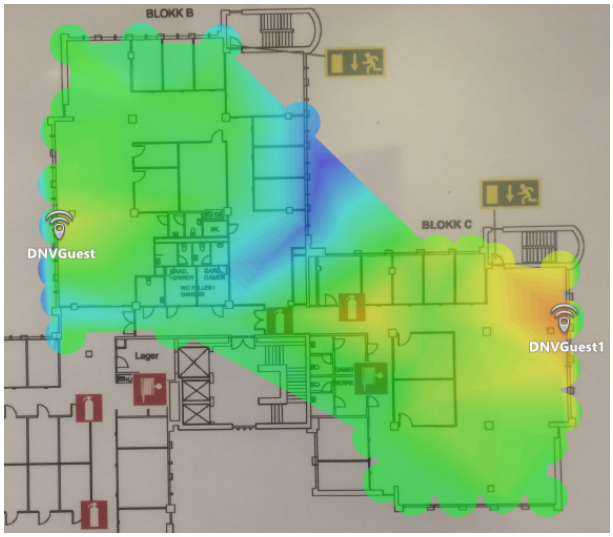

Figure 5.9: Partially overlapping cells. Dark blue indicates signal strengths weaker than -80 dbm

two separate packages. The first part could test if the current WLANs meet the clients requirements. The second part could then try to figure out how to improve the WLAN quality. If the clients wants you to test possible improvements, then you would have to ensure that you actually get enough time to do this. If not, then the recommended improvements may be to inaccurate for them to actually guarantee meeting the quality requirements. The results may therefore end up having no or little value for the clients. The procedure I have created in chapter 7 does contain both parts of the WLAN testing procedure. However, it also indicates where you should stop if you only wish to perform part 1.

5.8.2 Inaccurate results

As mentioned in section 4.1.1, an accurate throughput requirement will require network usage data over a longer period of time. This means that the clients will have to use some sort of monitoring tool. If you are going to estimate these requirements yourself, then you should consider investing in tools such as Ekahau. This is because this tool has predictive survey capabilities. These capabilities will do a lot better job of estimating accurate throughput requirements than you will be able to do by hand. Additionally, the throughput estimations you make are based on information you get from the clients. This information will also be inaccurate. The estimations you do will therefore have an inaccurate starting point. You will therefore have to recommend the clients to install some sort of monitoring tool to be

able to have an accurate starting point.

If you still want to calculate the throughput yourself, then you will have to base your calculations on the information the clients give you. You will then have to use the services/applications and the people/devices an area should support to create a throughput requirement. This will then have to be compared to the total capacity the APs in the area provides. The actual capacity should have quite a good margin compared to the actual throughput requirements. You will also have to try and adjust this margin based on the amount of interference in the area. You may want to consider being a bit generous when calculating how much throughput that is needed. Being generous should ensure that you do not underestimate the amount of throughput the clients actually need. This way you should at least avoid saying that certain areas are meeting the clients requirements when they actually do not. You can still improve the throughput even though you only have an inaccurate throughput requirement. The different methods described in section 5.7 can still be used. Basically you want to increase the capacity without introducing to much interference.

The coverage requirements should be easier to figure out. This is not something that needs to be measured over a longer period of time. You will simply have to check if the signal quality in an area meets the requirements of the clients services/applications.

5.8.3 Using results from the pentesting procedure

There are some results from the pentesting procedure that can be beneficial to share with the quality testing. The different site survey tools do not have features for finding the SSIDs of APs that are set to hide their presence. These will only appear as a seperate MAC-address on your heatmap. These MAC-addresses will not be placed inside of any WLAN, even though they might belong to one. The pentesting procedure will be able to figure out the actual SSIDs of such APs. If this AP does belong to one of the WLANs we are testing, then this will have an affect on our results.

5.8.4 Tools you should use

The selection of tools to be chosen for the testing is based on how much you are willing to spend on software licenses. If you want predictive survey capabilities, then I would recommend buying Ekahau. If not, then I would choose Netspot. This is because of the general good experiences I had with Netspot and the fact that you get a lot of good features for a relatively small price. Additionally, you should definitely consider investing in one of the pro versions of InSSIDer that supports Wi-spy. This is because it will really help with identifying and removing interference.

Chapter 6
WLAN Security

This chapter describes methods and tools that can be used to test WLAN security. The chapter first describes the testing environment that was used for pentesting during this project. It then goes on to describe different methods you could use to gain WLAN access. The chapter then goes on to show different ways you could exploit a system, based on the access you gain from a WLAN. Finally the chapter discusses the experience and results that we have obtained.

6.1 Pentest Lab

This project tried to use a realistic industrial environment to perform tests on. This was done in order to create results and experiences comparable to a real world scenario. No industrial environment is going to have the exact same network or devices. This means that there are a lot of unknown factors. Trying to cover every possible scenario of networks, equipment and their configurations would be impossible. The test setup used in this project, was therefore intended to represent the important aspects of what you may find in an industrial environment. DNV-GL fortunately has good experience with these types of environments. The test setup and assumptions made about real test environments, are therefore based on their experiences. The test setup in our pentest lab is depicted in figure 6.1. This illustrates a simplified version of an industrial environment. This test setup was created at DNV-GLs cyber-security lab.

6.1.1 Internet access

Internet access could in real world scenarios be provided in different ways. Ships for example mostly get their internet access through satellite communication. Most industrial environments will have a firewall protecting networks in the industrial environment. In this test setup, we get the internet access through an ethernet connection, which goes through a Cisco ASA 5505 firewall.

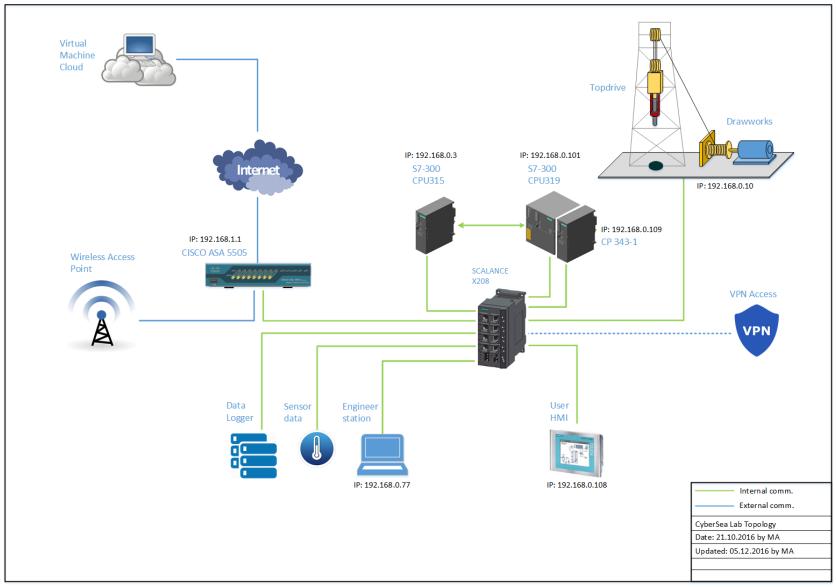

Figure 6.1: Main test setup at DNV-GLs pentest lab

6.1.2 Wireless LAN

The wireless part of the network usually consists of more than one WLAN that serves different purposes. Most industrial environments provides employees with internet access for personal use. There are also often other WLANs that require special clearance. There could for example be WLANs that are restricted to the captain of a ship and a few other crew members. These WLANs may be used for administrative work, and may provide access to different kinds of sensitive information/devices. There could be x number of routers, switches or firewalls, in the network, which limits the access to different devices. This means that there may be different scenarios that testers may face. Testing has therefore been done with multiple variations of the test setup in figure 6.1. It should be mentioned that attacking the systems in industrial environments does not need to start with the WLAN. However, the WLANs will be the only attack point considered when trying to get access to the network[1]. The wireless components used in this project consisted of two Cisco Aironet 1142N APs, as well as one D-link GO-RT-N150 router.

6.1.3 Wired network

The wired network in the pentesting lab is controlled by different firewalls/routers/switches. The main setup (seen in figure 6.1) consists of the Cisco ASA 5505

[1] As mentioned in chapter 4 the WLANs are the outer perimeter of our CDEs

firewall, which is connected to different APs, as well as a Siemens Scalance X208 switch.

The devices connected to the wired network are all used in the offshore industry. First of all, there are two Programmable Logical Contollers (PLCs). A PLC is a specialized computer that can be used in any setting where you need to control some type of equipment. PLCs are used in industrial settings, seeing as they provide a flexible way to control and communicate with industrial equipment[plc]. The PLCs in the test setup are one Siemens Simatic S7-300 CPU315 and one Siemens Simatic S7-300 CPU319. The wired network also has a Human Machine Interface (HMI). The HMI makes it possible to interact with the PLCs. The whole system controls a miniature version of a drilling-rig. The drilling-rig is controlled by the PLCs that gets its input from the HMI.

6.1.4 Pentest lab variations

I have used four different variations of the test setup shown in figure 6.1. This was done to illustrate and test the different levels of access, that you may obtain, from different WLANs. Although the WLANs are given different levels of access, the ultimate goal is to be able to exploit the PLCs. This is because exploiting these PLCs may prove fatal for the drilling-rig. Attackers that are able to manipulate the PLCs and the drilling-rig is the worst case scenario for this system.

The different setup variations are based on different firewall and router configurations. The Cisco ASA 5505 firewall differentiates traffic based on which Virtual LAN (VLAN) it originates from[cis14][cis]. The basic license that I had for this firewall, only allowed the use of two different VLANs. These VLANs were called "inside" and "outside". The inside VLAN is by default set to security level 100, while the outside security level is set to 0. The main idea of this firewall is that devices can contact/connect to devices that are on the same or lower level of security. This means that devices on the inside VLAN can contact/connect to devices on the outside VLAN, but not the other way around. I have also used a D-Link GO-RT-N150 router on the inside VLAN. This router has its own firewall capabilities. This creates a third security level, which protects the control systems. The level of access that a user/device can get, depends which VLAN the APs are placed in. Different users/devices can also be given special privileges to access higher security VLANs. Each of the variations I have used gives user different levels of access.

In WLAN 1 (figure 6.2), the Cisco AP is placed in the outside VLAN. The AP is in the 192.168.168.0/24 subnetwork with the Cisco firewall as the default gateway. The Cisco firewall is connected to the Siemens switch on the 192.168.1.0/24 subnetwork. The switch is then connected to the target PLCs on the 192.168.0.0/24 subnetwork.

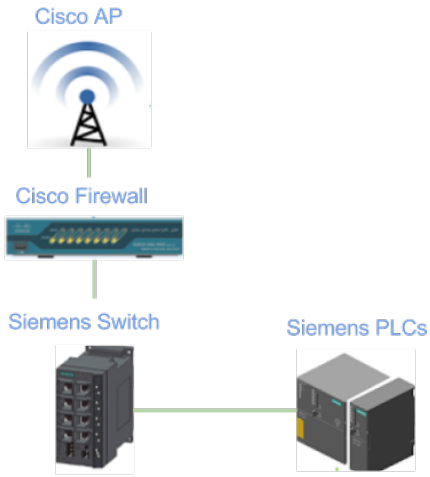

Figure 6.2: WLAN 1 is located in the Cisco firewalls outside VLAN

In WLAN 2 (figure 6.3), the AP is placed in the inside VLAN. The AP is on the 192.168.1.0/24 subnetwork with the Cisco firewall as the default gateway. It's important to note that this is the default configuration of the firewall. This is because the ethernet port, providing internet access, is by default the only port set to the outside VLAN. The firewall is directly connected to the Siemens switch on the same subnetwork. The switch is then connected to the target PLCs on the 192.168.0.0/24 subnetwork.

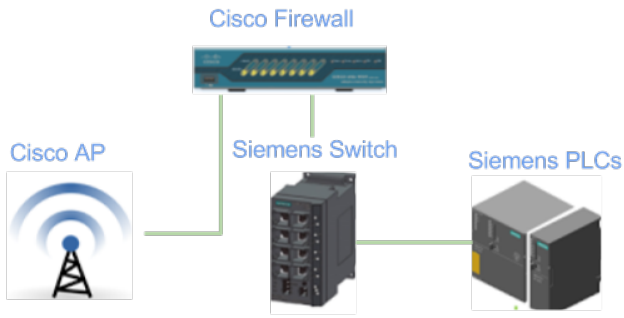

Figure 6.3: WLAN 2 is located in the Cisco firewalls inside VLAN

In WLAN 3 (figure 6.4), the AP is placed on the inside VLAN. The AP is on the 192.168.1.0/24 subnetwork with the Cisco firewall as the default gateway. The Cisco

firewall is now connected to the D-Link router on the same subnetwork. The D-Link router has its own "inside" VLAN. The Siemens switch is located on the D-Link routers inside VLAN on subnetwork 192.168.2.0/24. The Siemens switch is finally connected to the PLCs on the 192.168.0/24 subnetwork.

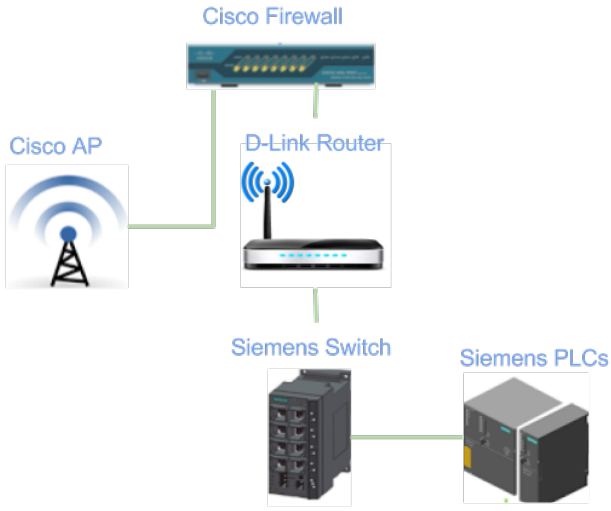

Figure 6.4: WLAN 3 is located in the Cisco firewalls inside VLAN but on the D-Link routers outside VLAN

In WLAN 4 (figure 6.5), the user is connected directly to the D-Link routers inside VLAN. The AP is now on subnetwork 192.168.2.0/24 with the D-Link router as the default gateway. The D-Link router is now directly connected to the PLCs on subnetwork 192.168.0.0/24.

6.2 Wireless Network Discovery

Seeing as this test procedure will test what access you are able to obtain from a WLAN, it's only natural to first try to gain access to a WLAN. The first thing you should do in any pentest is reconnaissance. Different WLAN configurations will give you different attack vectors. These vectors are what determines how you should proceed with the pentesting procedure.

We want to detect different APs in the area, which WLAN they belong to and what type of security protocol the APs are using (Open/WEP/WPS/WPA-PSK/WPA2-PSK/WPA-enterprise). You could use the Wifi Pineapple for this task. Both the Recon module and the Site Survey module are made to perform

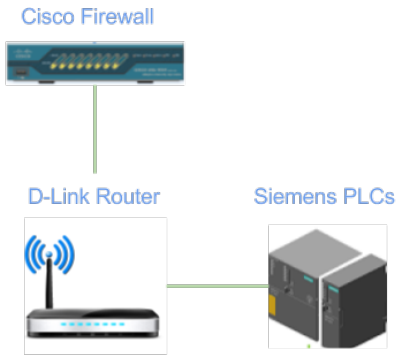

Figure 6.5: WLAN 4 is on the D-Link router which is on the cisco firewalls inside VLAN

wireless network discovery. Both modules will provide you with information about the security protocol, SSID and Connected clients. Both of them also have the ability to deauthenticate clients. However, the Site Survey module is the only one that can capture authentication handshakes[2]. You should try scanning the network from different locations, seeing as you will most likely not be able to detect all devices from one location. If you do not have a Wifi Pineapple, then you could use the aircrack-ng tool-suite. This tool-suite works similarly to both the Recon and Site Survey modules. This is because these modules use aircrack-ng for much of the underlying functionality. The Wifi Pineapple modules are however easier to use. The web-interface also provides better overview and logging capabilities.

Another thing that you should check is whether or not there are rogue APs in the area. The clients should have provided you with a list of MAC-addresses that represents their APs. Any APs you detect that does not have a MAC-address on this list, should be located and investigated. You could locate these APs by using the results you got in the WLAN quality testing procedure. The heatmaps you generated here should make it easy for you to locate different APs. Alternatively, you can try to locate this AP based on the signal strengths you get from the Recon or Site Survey modules. Once you have located said AP, then you will have to confer with the clients. They should be able to tell you if this actually is a rogue AP or just an AP they forgot to mention.

Once you have scanned the wireless network, then you should use this information, to gain access to the different WLANs.

[2]Deauthenticating clients are important when you want to crack WEP, WPA-/WPA2-PSK or obtain the SSID of "hidden" APs

6.3 Accessing WLANs using WEP, WPS or No Authentication

Certain WLAN configurations will make it quite easy for you to get network access. WLANs that does not use any form of authentication are obviously easiest. In most cases where you see open WLANs, it's because the clients intentionally wants this WLAN to be easily accessible. However, it may also be a misconfiguration, or in a worst case scenario a rouge AP. You should also check if these WLANs are using some other form of authentication. Networks often use web-portals to authenticate users on the application-layer rather than the link-layer. To check if a WLAN is using a web-portal, you simply need to connect to the WLAN, open a browser and try to access some web page. If there is a web-portal, then this should ask you for your credentials, before granting you further access.

APs that operate with WEP as the security protocol are also easy to access. Securing a WLAN using WEP is futile, seeing as it can easily be "cracked". If testers find APs using WEP, then "alarms" should be going of. The clients are obviously trying to secure their network, from unauthorized access, but are failing miserably. Accessing a network using WEP is a very well known exploit. The most well known tool for cracking WEP is aircrack-ng. The process can be split into two parts. The first part is to obtain lots of initialization vectors (IVs)[3]. Capturing these IVs is the part that takes the most time. You can use a combination of airmon-ng, airodump-ng and aireplay-ng to obatin these IVs. These tools are all part of the aircrack-ng tool-suite and will automatically be installed with aircrack-ng. Note that not all network cards are capable of letting you use monitor mode properly on your computer. This means that you will not be able to capture IVs. Wifi Pineapples does support monitor mode and can therefore be used if you have one available. Airmon-ng enables monitor mode (lets you listen to all traffic in the air). Aireplay-ng lets you do fake deauthentications, which means that you can collect IVs a lot quicker. Airodump-ng is the tool that actually captures the the IVs. You will make sure that any traffic captured by airodump-ng is logged to a pcap file. The second part of the exploit is to find the WEP-key. This is done by providing aircrack-ng with the captured pcap file. Finding the correct WEP-key should not take more than a minute. When testing this exploit, I set up one of my Cisco APs with WEP. The tests that were done with different passwords all found the WEP-key within minutes. You can also use a tool called Wifite to simplify the cracking process. With Wifite you can do the whole process with just one tool[4].

[3]It's worth mentioning that capturing a handshake will also reveal the SSIDs of APs that are trying to hide their presence

[4]Wifite also needs a proper network card to function properly, since it simply runs the aircrack-ng tool

Another security configuration that should raise some "alarms" is APs that has Wifi Protected Setup (WPS) enabled. WPS is a network security standard that was created to make it easy for users with little wireless security knowledge to easily setup their network with WPA/WPA2 as their security protocol. WPS has multiple documented vulnerabilities. The first weakness found, utilizes the weak PIN structure which enables attackers to brute-force the PIN in 4-10 hours[wps]. Some APs also have a separate physical WPS button. Pushing this button will temporarily open the AP for new connections. APs with this feature could therefore be exploited if the AP is physically available. A popular WPS exploitation tool is Reaver. This tool can be used to crack the PIN by exploiting the weak PIN structure.

The common factor for WLANs that has open APs, APs using WEP or APs using WPS, is that you know an attacker can get access to these WLANs. APs using WPS should be disabled. APs using WEP should be configured to use WPA-/WPA2-PSK or WPA-/WPA2-Enterprise. Equipment that only supports WEP should be exchanged, seeing as this technology is deprecated. Knowing what to do when you find open APs is not as straight forward. You should first of all have information from your clients whether or not this configuration is intentional. If so, then you should do proper segmentation testing. This is to ensure that you are not able to reach or use devices or services in the internal network (see section 6.7). These types of WLANs should be completely separated from the rest of the clients internal network and should only provide users with internet access.

6.4 Accessing WLANs using Web-Portals

As mentioned in section 6.3, WLANs listed as open may prove to be using a web-portal for user authentication.

6.4.1 Web-page vulnerabilities

Most web-portals are very simple web pages, with limited functionality. They have a limited amount of input fields and a simple graphical layout. This makes the actual website rather simple to secure. However, it could be a good idea to do a quick scan using tools like Metasploit or OWASP ZAP. These tools are quite extensive and has a lot of different features. However, they also make it quite easy to perform quick security scans. Scanning the web-portals may suffice. However, you could of course do more extensive testing, if you have the time.

6.4.2 Phishing for credentials

You could also try to gain access to these types of WLANs by phishing for credentials. This could be done by using a combination of Wifi Pineapple modules. The first

step of the attack is to have the Wifi Pineapple act as a rogue AP. The second part is to provide the connected clients with a fake web-portal, which they enter their credentials into. To make this attack work, your Wifi Pineapple will need to be in proximity to the actual WLAN users. Nearby clients, and the APs they are connected to, can be detected using the Recon or Site Survey modules (see section 6.2). These modules can also be used to deauthenticate the victims from their AP. Deauthenticated victims may decide to connect to your rouge AP. Whether they connect to the rogue AP or the actual AP depends on the signal quality that a user receives. The Wifi Pineapple tetra uses a higher transmission power than most APs. The signal quality from the rogue AP will therefore seem better than other APs. PineAP is the Wifi Pineapple module that manages the rouge AP features. The PineAP can act as a rogue AP in multiple WLANs simultaneously. It can pretend to be a member of any network it has detected in the vicinity. It can also pretend to be a part of WLANs that are actually nowhere near its current location. However, it could be a good idea to limit your phishing attack. PineAP should only associate with clients asking for the SSID of the WLANs using a web-portal. You may even wish to limit your attack, to certain MAC-addresses, that you know you are allowed to target. This can be configured in Wifi Pineapples filtering module. This module allows you to set which users you are allowed to or not allowed to target.

The second part of the attack is to provide the connected users with a fake web-portal. The Evil Portal module can be used for this purpose. When Evil Portal is enabled, then all users will be "blacklisted". This means they have not been authorized to get any further network access. Your fake web-portal will appear when blacklisted clients try to use a browser. Evil Portal requires you to design the front-end yourself. You could possibly clone the front-end of the web-portal you are trying to mimic. The credentials will also need to be posted correctly to the back-end. The Evil Portal handles most of the authentication/rejection procedure. You can use some of Evil Portals helping functions to either deny or approve users. You only have to write the code that extracts the credentials, writes the credentials to a log and decides whether the posted credentials should allow the users to become "whitelisted". Whitelisted users will be allowed to use the Wifi Pineapples network connection, whatever this connection may be[5]. The back-end code is written in PHP. This attack was tested out in a commercial pentesting project, that I participated in, with DNV-GL. See section 6.11 for details.

6.4.3 Things you need to consider when testing a web-portal

Using the Wifi Pineapple as a rogue AP is not always as straight forward as it may seem. The little documentation that there is, only covers the basics. There has

[5]The Wifi Pineapple can give users internet access by using a hotspot, ethernet connection or some other internet connection in the area

been a lot of cases where there were no documentation for solving my problems. When this happens, you have to fumble around for countless of hours before you hopefully solve the problem. Several of the Wifi Pineapple modules act sporadically, without there being any explanation as to why. One of the main issues I have had, is how "selective" the Wifi Pineapples DHCP server is. The DHCP server is supposed to provide devices with IP-addresses. This only works sporadically when clients attempt to connect to the rogue AP. Devices can sometimes be provided with an IP-address and sometimes not (without it having anything to do with the filtering configurations). Some types of devices were more problematic than others[6]. However, I was unable to find out exactly what causes these problems. It should however be mentioned that the Wifi Pineapple Tetra is quite new. There will therefore most likely be updates to both firmware and documentation.

It could be a good idea to set up a web-portal before actually entering the test environment. This could save you a lot of time on the test site. Cloning the web-portal and making it work properly with Evil portal may take some time. Evil Portal is not one of the official Wifi Pineapple modules and does therefore have very limited documentation. This means that you may find yourself in a tricky situation, that will require some time to find a solution to.

A final thing that you should be well aware of, when using a Wifi Pineapple as a rogue AP, is the "rules of engagement". By this we mean which clients and APs you are actually allowed to exploit. It's not a given that you can steal the credentials from any clients on a WLAN. If the filters are not set correctly, then the Wifi Pineapple might overreach and attack users that should be left alone. There were several times in the pentest lab where this happened. It's easy to forget checking the filter configurations before initializing the PineAP module. This led to several cases where random users connected to the rogue AP. You should be aware that every time you edit the Wifi Pineapples network configurations, then you will automatically reset all filter configurations.

6.5 Accessing WLANs using WPA-/WPA2-PSK

WLANs using WPA-/WPA2-PSK (from now on referred to as WPA-PSK) are usually safer than WLANs using WEP, WPS, web-portals or no authentication. However, the security of WPA-PSK depends on how it's used.

6.5.1 Cracking WPA-/WPA2-PSK passwords

WLANs that use WPA-PSK with weak passwords, are susceptible to being cracked. Cracking a WPA-PSK password has two phases. The first phase is to capture a

[6]This sporadic behavior was not connected to any rogue AP countermeasures

WPA-PSK handshake. The process of capturing a WPA-PSK handshake is similar as capturing WEP-IVs (see section 6.3). You can also use the exact same tools as you do when cracking WEP. You can also use the Wifi Pineapples Site Survey module to capture handshakes. This module may be the simplest way to capture a handshake. The second part of the attack is to try and find the correct password. You will here try to find the correct password by creating a hash, that is equal to the one that the client responds with, during the four-way-handshake. The hashes you generate are based on the SSIDs and MAC-addresses involved in the handshake, as well as the password you wish to test. The password search can be done by using a brute-force attack or a dictionary attack. A brute-force attack will systematically generate all possible passwords. This is not the best way to find a WPA-PSK password as it usual takes way to long. The better option would be to use a dictionary attack. This type of attack will iterate through a list of possible passwords, until you get a match. The success of this attack depends on the password entropy and size/quality of the "wordlist". Skull Security has a wide range of different types of password lists[Seca]. These lists contain everything from the 500 most used passwords, to lists that contain terabytes of actual stolen passwords. The bigger the wordlist, the more likely you are to find the right password. However, trying all the passwords in a big list will take a lot of time. It could take several hours or even days. Your computers processing power will therefore have a large impact on how long the process will take. You will have to consider how big a wordlist you should use. A popular tool, that is often used to create or edit wordlists, is John the Ripper. This tool has some good functionality that lets you create your own wordlists. You can use it to create permutations of the wordlists you already have. John the Ripper comes with its own set of permutation-rules. These rules try to capture the patterns people use when creating passwords. If I use a basic set of permutation rules on a wordlist that has 8,5MB of passwords, then these rules creates a wordlist containing 194,5MB of permuted passwords. KoreLogic Security has yearly password cracking competitions. Here people show of different approaches to cracking passwords. They have, based on the experiences of these competitions, created a set of John the Ripper rules. These rules have been created to give better password cracking results for corporate environments[Secb].

Once you have got your wordlist, then you need to use a password cracking tool to actually check the different passwords. The key to a good password cracker is speed. You could use tools such as aircrack-ng or Pyrit. The main difference between aircrack-ng and Pyrit, is that aircrack-ng only utilizes the CPU-cores on your computer. Pyrit can also utilize the processing power of your GPU. Aircrack-ng is however a viable option, seeing as the tool-suite already has tools for capturing a handshake (pyrit does not).

6.5.2 Password cracking test results

Note that using a proper GPU may improve your processing power a lot. How much faster the the cracking process becomes, depends on the quality of your GPU. My graphics card did not support WPA-PSK password cracking. I therefore only cracked passwords using my CPU. My CPU is a Intel Core i7-6500U 2.50GHz CPU with 4 cores[7]. One of the tests that were done, used a wordlist containing 1.000.000 passwords[8].

Aircrack-ng will use the captured handshake and the SSID/BSSID of the target AP to test every password in the wordlist from the start to the end. If the tool finds the right password, then the process will be terminated. If the password actually is in the wordlist, then the processing time will be dependent on which index the correct password has in the wordlist. So if the correct password is listed first in the wordlist, then aircrack-ng will find it instantly. The tests were done with the correct password at different indexes in the wordlist. Whats important to notice, is that the cracking tools automatically recognize which of the passwords, in the wordlist, that do not fit the WPA-PSK specifications. These passwords will be eliminated immediately. The cracking tools will not bother to test them at all. This means that they reduce the actual amount of passwords they will test. In our case, the amount of passwords was reduced to 488.135. Table 6.1, shows how long time it took to find the passwords when it was placed at different indexes in the 1.000.000 password list. However the processing speeds do not fluctuate as much as the times indicate. They only fluctuate this much since a lot of passwords, from different places in the wordlist, has been removed(over half of the passwords are removed immediately). The actual processing speed on my computer only fluctuates between 1600-1800 passwords per second.

Table 6.1: Pre-processing and cracking WPA-/WPA2-PSK passwords using Aircrack-ng

Index of password in wordlist	Time used to find password
20	0,31s
50.000	1m 37,41s
200.000	2m 1,91s
1.000.000	4m 55,93s

Pyrit works differently from what aircrack-ng does. The great thing about Pyrit, is that you can do most of the processing, before actually capturing a handshake. As

[7]The speed of the cracking process will also depend on the amount of processing power other processes on your computer are using.

[8]You would usually need a bigger wordlist in a real life scenario

long as you have the SSID and a list of passwords, then you can do most of the time consuming work, before actually entering the testing environment. The password cracking is extremely fast if you have done pre-processing ahead of time. The way it works is that Pyrit makes "prepared-hashes". These prepared-hashes are generated at approximately the same speed as aircrack-ng crack a passwords[9]. In other words, it creates 1600-1800 of these prepared-hashes per second. If you get the SSIDs of the WLANs you wish to test beforehand, then you could make as many prepared-hashes as you want before you actually start testing. You will still need to capture the correct handshake once you are on site. Then you can use the prepared-hashes to see if they have the correct password. If you look at table 6.2, then you can see how long Pyrit uses to create prepared-hashes and how long it actually takes to crack the password[10].

Table 6.2: Pre-processing and cracking WPA-/WPA2-PSK passwords using Pyrit

Process	**Time**
Pre-processing 1.000.000 passwords	4m 48,12s
Cracking password	0,89 s

6.5.3 Things you need to consider when cracking a password

So, Pyrit will be a lot faster if you can create prepared-hashes ahead of time. This way we could ask the clients to give us their SSIDs before we start testing. However, aircrack-ng can stop the password cracking early, if it finds the correct password. You can therefore be lucky and find the password with very little processing. If you use Pyrit, then you cannot stop midways. This is because it has not been given a handshake yet. The best cracking procedure will depend on the situation. You should therefore consider the different options you have:

1. You could choose to crack the password yourself on-site. However, if you do this, then you should not choose a massive wordlist and waste a lot of time.
2. You could bring a separate computer for cracking passwords. This computer could continuously work on finding passwords.
3. You could send the captured handshake to someone that are off-site. They could then crack the password for you.
4. You could get the SSIDs of the networks you are going to test from your clients beforehand. That way you could pre-process passwords.

[9]Note that this is without me using the GPU
[10]This process used the same wordlist that was used with aricrack-ng

5. The last option is that you could "cheat" and ask the clients for the actual password. You could then evaluate the password strength without cracking it. This may be the quickest way of checking the password. However, this will be less realistic.

6.6 WPA-/WPA2-Enterprise

WPA-/WPA2-Enterprise (from now on WPA-Enterprise) is the most secure wifi security configuration. However, there are still ways to gain access to these networks.

You could still try to perform a dictionary attack. As apposed to WPA-PSK, this configuration uses personal passwords for each user. The authentication is also handled by a separate Remote Authentication Dial-In User Service (RADIUS) server. This makes the password cracking procedure a lot more complex than it is for WPA-PSK password cracking. To capture a WPA-Enterprise challenge and response requires you to simulate the target WLAN. This means that you will have to set up a rogue AP and a RADIUS server[Geia]. The rogue AP would have to use the target WLAN SSID. The RADIUS server would have to support all types of Extensible Authentication Protocol (EAP). If your server does not have the correct type of EAP, then you would not be able to perform a fake authentication. You could use a FreeRADIUS server for this purpose. This server could be set to accept all login attempts. Clients would then provide you with the correct challenge response. This challenge response could then be used in a dictionary attack.

Unfortunatly, I did not get to test this type of attack. This attack would therefore need further investigation. You will have to find out details on how to complete the attack and which tools you could use.

6.7 Wired Network Discovery

Trying to gain access to different WLANs, may yield different results. You may be able to gain access to all of them or none at all. You should still proceed as if you were able to access all of them. The fact that you were not able to exploit the different WLANs, does not mean that others were not as well. An attacker could have more time to perform their attack, be more skilled than you, or even be an employee that already has a certain amount of access. You should therefore check what you are able to do once you have got access to the different WLANs. This means that the clients should grant you access to WLANs you were unable to access.

At this point, the topology of the wired network becomes highly relevant. When you have just accessed a WLAN, you will most likely be unaware of the actual structure of the network. Detecting the different devices on the wired network is a

more complicated process than it is on a wireless network. This is because wired networks do not share a transport-medium as the wireless networks does. However, there are still ways to perform wired network discovery.

In the pentest lab, the ultimate goal was to exploit the PLCs. We therefore wish to locate these PLCs. You may not be able to detect them at first. You will therefore try to see if you could get past, or use other devices, to get "closer" to the PLCs. It's therefore important to gather as much information as possible about devices. The more information you can get about the network, the more likely you are to find vulnerabilities. Nmap is a tool that is well suited for this task. This tool is a security scanner with a lot of possibilities. Nmap is a good tool for beginners as well as experts. It has different features that ranges from discovering devices on your network to detecting what types of services a device is running. The time you invest in learning how Nmap works is truly worth the time, as it will help you immensely when doing wired network discovery.

6.7.1 Ping sweeps of one subnetwork

In a pentesting procedure, you will often be unaware of the network structure. You should therefore try to figure out which devices you are able to reach. One of the pitfalls of using Nmap, is that most of the scans it uses, can take a lot of time. You therefore have to be selective, when choosing how to scan the network. A good way to start of, is by doing a ping-sweep. A ping-sweep is done to check a range of IP-addresses for live hosts. Nmap has several different ways of doing ping-sweeps. The most common one is to use ICMP packets. The other alternatives are used when you try to trick firewalls/routers/switches, which often try to block ping-probes. Doing a ping-sweep of your own subnet is quite easy and only requires a single Nmap command. In this command, you will need to specify what type of ping-sweep you wish to do, as well as the subnetwork you are on. The output you get from the different devices in a ping-sweep, looks something like this:

```
Nmap scan report for 192.168.0.1
Host is up (0.0023s latency).
MAC Address: 78:54:2E:50:2D:C6 (D-Link International)

Nmap scan report for 192.168.0.3
Host is up (0.0039s latency).
MAC Address: 00:0E:8C:F9:15:4A (Siemens AG A&D ET)
```

As you can see from these results, you will already get good hints about what type of devices you are dealing with. In these example outputs, from WLAN 4, the first device is the D-Link router. The second device is one of the Siemens PLCs.

Note that the devices MAC-address, and the vendors these addresses belong to, only appears if you run Nmap with root privileges[11].

When doing this type of ping-sweep, you will only be able to detect devices that are located on your own subnetwork. The different WLANs in section 6.1.4 will detect different devices. However, the only WLAN being able to see the PLCs at this point, is WLAN 4. Note that in a realistic scenario, you will only have a limited amount of information about the devices that you have detected at this point. To be able to figure out what types of devices you have actually detected requires more comprehensive scans (see section 6.8).

6.7.2 Ping-sweeps across multiple subnetworks

To get more information about which devices that are active on the wired network requires you to do ping-sweeps across multiple subnetworks. Doing these types of ping-sweeps are a bit more complex. First of all, you most likely do not know exactly which ranges of IP-addresses that are in use. You will therefore not know how many subnetworks there actually are. Ping-sweeps of a subnetwork, with a total of 254 addresses, usually takes 2-11 seconds[12]. This means that a ping-sweep, of all possible network addresses, could take quite a while. This will depend on the network type the clients are using. You should use the information you have available, to limit the amount of addresses you have to check.[sub] The most common types of networks are A, B and C. Class A has a default subnet mask of 255.0.0.0. Class B has a default subnet mask of 255.255.0.0. Class C has a default subnet mask of 255.255.255.0. It should be mentioned that these are only default values and can therefore be modified by network administrators. It's therefore difficult to know which IP-addresses that may be in use. However, some address-ranges are more common than others in private networks[com]. The most common IP-ranges for class A networks are 10.0.0.0 to 10.255.255.255. The most common IP ranges for class B networks are 172.16.0.0 to 172.31.255.255. The most common IP ranges for class C networks are 192.168.0.0 to 192.168.255.255. This should help you narrow down the number of addresses you need to scan. However, you may still be left with quite a lot of possible IP-addresses. Also, there is no guarantee that all devices are within these ranges. Asking the clients which subnetworks that are actually active, may therefore be a good idea. This can save you a lot of time.

In the pentest lab, all devices had IP-addresses between 192.168.0.0 to 192.168.255.255. The different subnetworks could not be scanned, by simply setting a new subnetwork, in the Nmap ping-sweep command. This is because we would

[11] Certain Nmap features requires root privileges to function optimally

[12] Although there are a total of 256 possible addresses, the addresses ending with 0 and 255 are used to indicate network ID and broadcast messages

need to have a valid IP-address within the subnets we are checking. This is because the different APs we are connected to will only provide us with one IP-address. We could set a static IP-address inside the 192.168.0.0/24 subnetwork using ifconfig:

```
$ sudo ifconfig wlan0:0 192.168.0.250 netmask 255.255.255.0 up
```

Doing this manually, for each subnetwork, can be quite time consuming. This is why I used a python script to automate the process. This script assumes that all subnetworks have the last eight bits, of the IP-address, reserved for hosts. This gives us a possibility of 256 subnetworks with 254 hosts each. The script takes a list, or range of target subnetworks. It then iterates through each of these subnetworks. For each iteration the script:

- Sets up a static IP-address in said subnetwork
- Does a ping-sweep in this subnetwork (using Nmap)
- Logs the output
- Removes the static IP-address

This script was used for wired network discovery in the pentest lab. It was also used in a commercial pentest project (see section 6.11). The script was tested on all 256 possible subnetworks. This took 44 minutes and 42.66 seconds. The subnetworks that were within reach, and had live devices, only used about 2.5 seconds. The subnetworks that were inactive used approximately 10.5. This is due to the timeout Nmap uses before giving up on a subnetwork. This can be fixed by using Nmap's timeout options. The timeout options can tell Nmap to give up on a subnet after a certain amount of time. We can then set this timeout to 1 second. Nmap is now still able to detect all devices in the subnetworks. Scanning all 256 subnetworks now only takes 4 minutes and 59,49 seconds. Another good option you have, when doing ping-sweeps, is to use Nmap's traceroute feature. This will provide you with information about the different "hops", your packets take, on your way to the target IP-address.

Using the python script made it possible to detect the PLCs from WLAN 2, as well as WLAN 4. However, the information we have at this point, is not enough to know that these actually are PLCs.

6.7.3 Discovering devices behind firewalls

At this point we are unable to detect or contact the PLCs from WLAN 1 and WLAN 3. This is due to the fact that in these scenarios, the ping-probes are being stopped

by firewalls. Fortunately Nmap has different ways of detecting devices that hides behind firewalls. You can first of all test out all the different types of ping-sweeps. Different ping-sweeps may not be stopped by a firewall. Nmap's help page also provides you with alternative options. One of the sections, in the Nmap manual, is called Firewall/IDS evasion and spoofing. These scans give you the opportunity to do things like MAC- and IP-address spoofing, fragmenting packets or sending decoy packets. You should definitely consider using these features. They may help you discover what lies behind a firewall. Nmap also has its own script engine. This comes with a large amount of scripts. I have tested firewalk[Dor] and firwall-bypass[Ben] during this project. The firewalk script tries to find out what type of traffic rules the firewall has. The firewall-bypass script tries to find vulnerabilities in the firewall, and open up ports to devices behind it. It should be mentioned that detecting devices behind a firewall does not necessarily mean that you will be able to connect to it. It will however give you a better overview of the network. This should then help you figure out where to focus your attacks.

Both the Cisco firewall and D-Link router proved difficult to exploit when they were using default firewall rules. Nmap was unable to detect any of the devices the firewalls protected from WLAN 1 and WLAN 3. None of the firewall evasion features, nor firewall scripts, were able to detect any protected devices. This showed that the the firewalls used in this project were quite good. The only result you get from these scans is that our probes are being blocked. However, as you will see in section 6.9, the firewalls were later configured to provide certain devices/users extra privileges. These devices/users were here allowed to bypass the firewalls.

Other firewalls may be easier to get past. This may be due to clients wanting less restrictions on their network, bad configurations or outdated equipment. Firewalls should definitely be tested properly seeing as they are an important aspect of a networks defense.

6.8 Service and OS Detection

Once you have done a network discovery, you will only have a limited amount of information about the devices that you have detected. You would for example not be able to identify PLCs at this point. You will need to do more detailed scans of different devices. Different types of devices or services may have known vulnerabilities. More information may therefore reveal possible exploits. Nmap offers a lot of great features for this purpose as well. Note that you should from now on only use IP-addresses of active hosts. This is because ping-sweeps are a lot faster than more comprehensive scans. Comprehensive scans may take several minutes for a single IP-address.

Nmap has separate features for doing OS-fingerprinting. This means that Nmap

uses certain known traits, about different operating systems. Nmap tries to match known traits with the devices you are scanning. Here you can see default OS scan results of one of the Cisco APs, as well as one of the PLCs:

```
Nmap scan report for 192.168.1.6
Host is up (0.0017s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
80/tcp open  http
MAC Address: 1C:DF:0F:95:D8:CA (Cisco Systems)
OS details: Cisco 836, 890, 1751, 1841, 2800, or 2900 router
(IOS 12.4 - 15.1), Cisco Aironet 1141N (IOS 12.4) or 3602I
(IOS 15.3) WAP, Cisco Aironet 2600-series WAP (IOS 15.2(2))

Nmap scan report for 192.168.0.3
Host is up (0.0020s latency).
All 1000 scanned ports on 192.168.0.3 are closed
MAC Address: 00:0E:8C:F9:15:4A (Siemens AG A&D ET)
Too many fingerprints match this host to give specific OS details
```

As you can see from the results, the tool had some problems identifying what type of device the PLC was. You could be able to understand exactly what type of devices these are. However, you would most likely need some prior knowledge about the network, or these specific devices. The results you get can now be used to search for possible exploits. If you were able to identify these devices, then you could use this information to search in exploit archives. I found different exploits of the s7-300 PLCs at Exploits Database[exp] and ICS-CERT[hom].

Nmap can also be used to scan for running services. Devices can be running services like ssh-, telnet-, ftp- or http-servers. Some services may prove vulnerable (see section 6.9). Whether or not a service is vulnerable often depends on the service version. This means that you should include version detection in your Nmap scans.

A running services will show you which port they are running on. However, it's important to note that different service and OS scans will not necessarily detect all open ports. It's therefore important to scan additional ports if you suspect a device of using less common ports. This is the case for the s7-300 PLCs. They use port 102. If you do a regular OS scan, then this will only check the 1000 most common ports. Port 102 is not among these ports. If you know or suspect what kind of device you are dealing with, then you could scan specific ports. If you are unsure, but think that a device may be vulnerable, then you could do a more extensive scan. You could

possibly scan all 65535 ports. If we suspect that a devices is a Siemens PLC, then we could combine service, OS and port scan on port 102. In this case we get:

```
Nmap scan report for 192.168.0.3
Host is up (0.0023s latency).
PORT     STATE SERVICE  VERSION
102/tcp open  iso-tsap Siemens S7 PLC
| s7-info:
|   Module: 6ES7 315-2EH14-0AB0
|   Basic Hardware: 6ES7 315-2EH14-0AB0
|   Version: 3.2.11
|   System Name: 315_HPU_copy
|   Module Type: CPU 315-2 PN/DP
|   Serial Number: S C-B5W071052011
|_  Copyright: Original Siemens Equipment
MAC Address: 00:0E:8C:F9:15:4A (Siemens AG A&D ET)
```

This information should definitely be enough to identify the device. At this point, we are therefore able to identify the PLCs from WLAN 2 and WLAN 4[13]. This means that we could possibly exploit the PLCs from these WLANs.

6.9 Exploiting Devices on the Network

Once you have completed the more detailed scans, then you might have found some vulnerable equipment. You could use tools like Metasploit to possibly simplify the exploit process. This is a tool for developing, testing and using exploit code. It comes with hundreds of pre-built exploit modules. I have not used this tool myself for any of the exploits that were used during this project. But being in control of the environment, I could choose the types of vulnerabilities that were present. In a more realistic scenario, you will most likely see vulnerabilities that you do not know how to exploit. It could therefore be useful to use tools such as Metasploit, which has a large database of exploits for various situations.

As you can see from section 6.7.3, we were unable to find any vulnerabilities in the default firewall configurations. This means that we are unable to detect the PLCs from WLAN 1 and WLAN 3. In a lot of cases the amount of access you have depends on the specific user and/or device you are using. These users/devices could be allowed past certain firewalls. Attackers could therefore use these users/devices to get further access. After scanning different devices, you may find that these devices have certain vulnerabilities. These vulnerabilities could be used to create a rogue

[13]The ability to identify certain equipment depends a lot on the testers experience as well as the test results themselves

agent or a backdoor. Taking over a device that does not have special privileges is also an important result. These devices still give you a valuable asset inside the network that could be used at any time.

6.9.1 Default credentials on open services

The first thing you could test out, with open services, is default credentials. The clients may have forgot to set new credentials when they install new equipment, or they may be unaware that this is something you need to do. The default credentials of a device can be found by "googling" it. For example, the default credentials for the Cisco APs are username: Cisco, password: Cisco. You could also check default passwords on the firewalls you are trying to bypass. In the pentest lab the D-Link router is open on port 80. You can therefore connect to it through a web-browser. The Cisco firewall requires you to run special software called asdm. If you get access to an AP's or a firewall's configurations, then you basically own the part of the network that these devices control. The firewall's configuration allows you to edit WLAN 1 and WLAN 3. They can now be set to the inside VLAN of their firewall, rather then the outside VLAN. I reconfigured the Cisco firewall this way and allowed WLAN 1 to be on the inside VLAN[14]. This way WLAN 1 has the same amount of access as WLAN 2. Doing ping-sweeps across multiple subnetworks will therefore allow WLAN 1 to detect the PLCs.

6.9.2 Cracking open services

You could also use tools such as Hydra or Ncrack to crack service-credentials. These tools can be used to perform dictionary attacks. The pentest lab was at this point set up with devices that had special privileges. This was done using MAC-filtering on the D-Link router. A Cisco AP and an HP computer (running Windows 7) was allowed access to the D-Links inside VLAN. The Cisco APs have ports 22 (ssh), 23 (telnet) and 80 (http) open by default. The HP-computer was set up to have a ssh- and telnet-server. Both of these devices were within reach of WLAN 3. Hydra was then used to try and crack the different service-credentials. Cracking WPA-PSK passwords and WEP-keys can be done offline. This means that you do not need to be in range of the entity you are trying to crack. When you are doing the tests offline, you are only dependent on the cracking speed of the computer and tool that you are using. Cracking ssh, telnet and http is a bit different. First of all you have to crack them online. This is because you rely on confirmation or rejection from each of your login attempts. Each of your attempts therefore needs to wait for the target device to respond. The speed therefore mostly rely on the target device. In some cases a device may use very long time to reply. There will also be a limit to how many parallel login attempts you can make. Service cracking is therefore slower

[14]Which actually is the default settings for this firewall

than WPA-PSK cracking. You will also need to use separate wordlists when cracking these services. One for usernames and one for passwords. This combination can be a lot more complex than a single password. The process can therefore consume a lot of time. You should therefore consider the different password cracking approaches mentioned in section 6.5.3[15].

Hydra had some difficulties cracking telnet on the different devices. The tool kept giving false positives. Ncrack on the other hand had problems with ssh. These results were kind of strange since the process of cracking ssh and telnet is quite straight forward for both tools. Cracking devices using http-login can in some cases be a bit more complex. This requires you to open a web-browser and investigate how the login works. The Cisco APs had a pop-up box. This appeared when trying to access the device. This pop-up box was very simple and made it quite easy to insert the credentials correctly. The D-Link router had a more complex front-end. This made the process more difficult, since you need to investigate the page properly. However, the correct form and Hydra command was found after some research.

We can compare the cracking speeds of these services with Aircrack-ng. This tool tested tested 1600-1800 passwords per second. The slowest service tested in this project was ssh on the Cisco AP. Hydra could only do 1,60 login attempts per second. The fastest service was the Cisco APs http-login. Hydra could here perform 8.35 login attempts per second. Trying to crack credentials that are moderately complex, would therefore take ages. You should therefore think twice before executing this type of attack. If you do wish to perform these types of attack, then you should at least restrict the size of your wordlists.

6.10 Exploiting the PLCs

As mentioned in section 6.1.3, the PLCs in the pentest lab are controlling a drilling-rig. The drilling-rig is the most critical piece of equipment in this setup. The s7-300 PLCs often control critical equipment, but still use unencrypted traffic because they are meant to operate in closed environments. These environments are supposed to be unreachable for unauthorized personnel.

There are two different exploits used on the PLCs in this project. Both of these attacks has been discovered during specialization projects at NTNU. Exploit 1 is a DOS-attack. This was discovered by Amund Bauck Sole in 2015[Sol15]. This attack is performed by connecting to a specific PLC and sending a special sequence of data. This vulnerability has been patched in newer firmware updates. This means

[15]Note that in this case you cannot use other people off-site to crack the credentials for you. Also, you do not have the option of pre-processing credentials

that it has to be performed on older versions[16]. The PLCs in this project used an older version for this specific purpose. Exploit 2 is a DOS-attack I discovered in 2016[Ohm16]. The attack manipulates specific bytes, in the transport-layer, between two communicating s7-300 PLCs. This attack requires the attacker to be placed as a MITM. This attack is not patched in the latest firmware. The effect that the different attacks may have on a system depends on the setup and what the PLCs are controlling. It was therefore unclear how these attacks would affect the drilling-rig.

6.10.1 Attacking from WLAN 4

Attacking from WLAN 4 is the easiest option. With this WLAN we can do all types of different Nmap scans from the same subnet. We can therefore identify that these are Siemens PLCs and that port 102 is open. This means that we can connect to this port and use Exploit 1. We then send the specified data to one of the PLCs. This is done using a python script that Amund Bauck Sole created. This makes all of the PLCs crash and start flashing red lights. This leads to quite a fatal failure in the drilling-rig. If the drilling-rig was on its way down, then it would keep going downwards. You would not be able to stop it. When the drilling-rig was going upwards it would not stop going upwards. The only way to stop the drilling-rig was to cut all power to the system. All PLCs needed a cold reboot to start working again. Exploit 1 had the same consequences independent of where the attack was performed from.

Exploit 2 was also possible to execute from WLAN 4. This is because we could arp-poison two of the PLCs. This was done using the arpspoof tool. By doing this we could manipulate certain bytes. This was done using a tool I have created in a previous project. Once the the manipulation was done, we can end the arp-poisoning. This sets the arp-caches back to normal. None of the lights start blinking when this attack is performed. This is because the PLCs think that everything is normal. Once this attack is executed, you could no longer operate the drilling-rig from the HMI. However, the way the drilling-rig was set up was so that it would move up and down within a certain range. This motion was not disrupted with Exploit 2 as it was with Exploit 1. You still needed to do a cold reboot of the PLCs to use the HMI again. WLAN 4 is the only WLAN we can execute Exploit 2 from.

6.10.2 Attacking from WLAN 2

With this WLAN we can do all types of different Nmap scans, as long as we have an IP-address inside the correct subnet. We could therefore gather the same information as with WLAN 4. So we are still able to identify the PLCs and perform Exploit 1. This gave the same results as it did with WLAN 4.

[16]This is something that could be discovered with Nmap scans

6.10.3 Attacking from WLAN 1 and WLAN 3

WLAN 1 or WLAN 3 is a bit more tricky. Here we are unable to detect the PLCs directly. However, WLAN 3 could reach devices with special privileges. We can therefore crack the ssh, telnet and http credentials of the Cisco APs. However, the Cisco AP has no way of running the python script used in Exploit 1. We therefore need to crack the ssh and telnet credentials of the HP-computer. Telnet does not allow file transfers. We therefore have to use Secure Copy (scp) to transfer the files to port 22. From here, we can connect to the PLCs. This way we can run the python script for Exploit 1[17]. The result of this attack were the same as it was with WLAN 2 and WLAN 4.

6.11 Testing a Real World Environment

Some of the methods and tools mentioned in this chapter has also been used in a commercial pentesting project. This is a project that my supervisor Mate Csorba was working on, which I got to participate on. Due to the confidentiality of this work, the results from these tests will not be shared in this paper.

6.11.1 Mapping out WLAN coverage

In this project we used Netspot and InSSIDer to map out coverage. We mapped out the coverage that the clients network had in two nearby restaurants. Netspot was used to generate heatmaps of these restaurants. This showed where an attacker could possibly try to attack their WLANs from. Screen-shots from InSSIDer were also taken from different locations. These were used to add additional signal information to the heatmaps.

6.11.2 Using a fake web-portal

The client used a web-portal to authenticate users on one of their WLANs. We therefore used the Wifi Pineapple Recon module to find APs and their clients. The OWASP ZAP scan of the web-portal, revealed little information of value. We then set up our phishing attack using the Wifi Pineapple. The Evil Portal module had already been prepared. The front-end was cloned beforehand in order to save time. However, the PineAP configurations were done on site. This is because it's a lot easier to configure the rogue AP filters correctly if the Wifi Pineapple actually knows which devices that are present. All of the tools and tests worked well. These tests gave a lot of valuable information for our testing procedure.

[17] I skipped the process of scanning the network again from this server. This can be done if you are able to install Nmap or write your own scanner

6.11.3 Scans of the wired network

We also got to to check the amount of access we got from the target WLAN. The clients had provided us with a list of active subnetworks. This way we could eliminate a lot of subnetworks from our ping-sweeps. We used the python-script I created for doing ping-sweeps across multiple subnetworks. Knowing which exact subnetworks that were active saved a lot of time. It also ensured that we did not scan anything we were not supposed to. I only got to participate at this project for half a work day. The amount of detailed scans we were able to perform were therefore limited. Again, the results from these tests proved valuable for the testing procedure.

6.11.4 Setting up a rogue AP/agent

The last thing we did, was to install the Wifi Pinapple on the inside of the clients WLAN. This was first of all done to see if the system could detect rogue entities. We could also use the Wifi Pineapple as our own backdoor. We could have tried to exploit one of the devices on the network to create a backdoor. Due to the limited time frame, creating a backdoor through exploitation was not prioritized.

6.12 Discussion of Results and Experiences

Finally in this chapter, I wish to summarize and discuss the results and experiences that were discovered during testing and research.

6.12.1 Things that can be left out

Certain aspects of the pentesting procedure could be left out if you find them to time consuming. PCI DSS recommends that an organization should do a full pentest once a year. In this pentest you should try to exploit the vulnerabilities that you have found. A vulnerability scan on the other hand should be done every quarter [Cou15b]. Therefore, you could argue that you should only check for vulnerabilities. The exploits that have recently been pentested can be omitted from this quarterly vulnerability scan.[18]. If you do choose to pentest, then you may still want to leave out the most time consuming aspects. All the processes that relates to finding passwords can be skipped. You can choose to do a white-box approach. This way clients can simply give you the passwords. Then you can evaluate them without any password cracking. The cracking processes are good for demonstrating to the clients why their web-portals are unsafe or why their passwords are to weak. However, you could also determine if a password or a web portal is weak by simply evaluating it. However, they may not take the problem as seriously if you do not actually crack them.

[18]The recommendations from PCI DSS may be a bit unrealistic for certain clients

6.12.2 Testing approach

You should also consider how much information you want to provide the testers with, for instance whether the pentesting procedure should be white-box, black-box or grey-box. The advantage of using a black-box approach is that this is the most realistic type of pentesting. Black-box testing are therefore more likely to convince clients to take exploits seriously. However, a black-box approach will take a lot more time. You will need to find information that you could have gotten straight away with a white-box approach. I had full control over the details in the pentesting lab. However, a lot of testing has been done as if I was using a black-box approach. Based on this experience, I would say that a grey-box approach could be a good solution. This way you could be selective about the information you choose to provide the testers with. This way they can save valuable time while still keeping most of the realism. You could for example provide the testers with more superficial information about the networks. This was the approach taken when I participated in the real pentest project (see section 6.11.3). We had the general information about which subnetworks that had live hosts. This saved us a lot of time. Not sharing information like live subnetworks may just be a waste of time.

It may be an idea to use different testers for the two different procedures. This is first of all since they may be given different amounts of information. Secondly, this also means that the testers can specialize in their respective area. Thirdly, you may save time if these procedures are done in parallel.

6.12.3 Using results from the quality testing procedure

Sharing test results between the two testing procedure may be beneficial. The pentesting procedure could benefit from getting coverage results. Based on these results, the pentest procedure may recommend restricting a certain WLAN to a smaller area. This way possible attackers are less likely to reach it. This is one of the things that we tested in the real pentesting project. The pentesting could also benefit from knowing the exact locations of the APs. This way they can more easily target specific APs.

Chapter 7
Testing Procedures

This chapter contains the final WLAN quality testing procedure and WLAN pentesting procedure. These procedures are a product of the results and experiences that have been gathered during this project. Every step in the procedures contain a short description, as well as a reference to the sections of this document where you can find additional documentation.

7.1 WLAN Quality Testing Procedure

1. **Training**
   Testers will have to learn all the theory behind the WLAN quality testing procedure. They should also have full control over the tools they are going to use. Little knowledge about your tools could make an expensive investment go to waste. See chapter 2 for tool information and chapter 5 for WLAN quality testing details.

2. **Interview**
   The interview should be used to define the scope for the test procedure. Specific boundaries should be set to define the testing area. The clients should provide you with one or several scaled maps of the target area. They also need to provide you with information about intended network usage. This means that they should tell you how many users they want to support in the different areas. Clients should also specify which services/applications the WLAN should support. Any further details about the environment would also help. This means anything from the placement of concrete walls to known sources of interference. See section 4.1 and 5.8 for more details.

3. **Define coverage requirements**
   The testers will now need to figure out the coverage quality needed in different areas. These requirements will depend on which services/applications they wish to use. See section 4.1.2 and 4.1.3 for more details.

4. **Define throughput requirements**
   How you define the throughput requirements depends on what kind of tool you have.

   a) **Estimating throughput requirements using a predictive-survey tool**
      This option requires you to use one of the more expensive tools. Ekahau is a good choice for using predictive-survey capabilities. This tool requires you to upload a scaled map of the area. It also requires you to provide information you have from your clients. This means the amount of people/devices an area should support and what type of services/applications they should be able to run. A predictive-survey tool should then be able to calculate the total throughput needed. See section 2.4.1 and 4.1.1 for more details.

   b) **Estimate throughput requirements without predictive-survey tool**
      In this case you will have to make a rough estimation. This estimation will also be based on the information provided by a client. This means the number of people an area should support and the services/applications they should be able to use. This type of estimation should be avoided if you can. This is because your results will be inaccurate and time consuming. See section 4.1.1 for more details.

5. **Gather signaling data for heatmaps**
   The first thing you will have to do, when you enter the testing environment, is to collect data. You will have to collect data from the entire testing area. This can be done by using either Netspot or Ekahau. Make sure to collect data from as many locations as possible. This will make the results more accurate. You should remember to keep a small distance between each data sample, collect data from the outer bounds of the testing area and get data samples from every room. You should also try to get samples near any sources of interference. See section 5.3.1 for more details.

6. **Edit heatmaps to show areas that do not meet the requirements**
   The colors of the heatmaps should clearly illustrate areas that does not meet the quality requirements. You should at least edit the signal strengths heatmap. This should also be done to signal-to-noise ratio and interference heatmaps if you have a paid version of Netspot/Ekahau. See section 4.1.2, 4.1.3 and 5.3.2 for more details.

7. **Examine heatmaps to find critical coverage areas**
   The areas that does not meet the coverage requirements should be marked for further investigation. See section 4.1.2, 4.1.3 and 5.3.2 for more details.

8. **Check AP specifications to find the capacity in different areas**
   The specifications of different APs should tell you the total capacity (maximum possible throughput) in different areas. You will have to do a rough estimation of how much throughput the area has based on the capacity and how much throughput that will be lost due to interference. Any area that does not meet the clients requirements should be marked as a critical area. These calculations could be more precise if you have a tool with predictive-survey capabilities. See section 5.3.2 for more details.

9. **Do spot checks to identify the root of coverage and throughput issues**
   Use InSSIDer at the "critical areas" to check if the signal information on the heatmap is accurate. You can then analyze the information to find the reason for coverage or throughput issues. Coverage may be bad due to low signal strength and/or due to interference. Throughput can be bad if there is a low total capacity within reach, low signal strength and/or to much interference. See section 5.4 and 5.5 for more details.

10. **Document why certain areas does not meet the quality requirements**
    Document which areas that do not meet the quality requirements and why they do not meet them. This is a good place to stop testing if you do not intend to test for WLAN quality improvements.

11. **Evaluate and test for improvements in areas that does not meet the requirements**
    Throughput, coverage and interference will all be affected by changes you make to the environment. Improving throughput may worsen the coverage. Improving coverage may worsen the throughput. You should therefore combine the information from section 5.4, 5.5, 5.6 and 5.7 to find a suitable solution. The main improvement aspects are:

    a) **Evaluate and test if the WLANs can meet the requirements with the current APs**
       Using the current APs can be a good solution if you only need to make minor adjustments. You could try moving, re-configure and redo channel selection of APs. You could also try to remove/relocate sources of interference.

    b) **Evaluate and test how you can meet the requirements with new APs**
       Adding or upgrading APs is often a good solution if you need to make bigger changes. However, you should check that the new APs does not create to much interference.

    c) **Evaluate if you have to do a complete redesign of the WLANs**
       If there are two many critical areas which makes the situation extremely

complex to solve, then you might consider telling your clients to redesign their WLANs from scratch.

12. **Document network improvements and which areas that has been improved**
    You should deliver the data you have collected to the clients. There should be data and documentation that shows the situation they have as well as the improvements you have tested. You should focus their attention to the critical areas and the improvements you have tested.

7.2 WLAN Pentesting Procedure

1. **Training**
   Testers will have to learn all the theory behind the pentesting procedure. They should also have full control over how the different tools work. See chapter 2 for tool information and chapter 6 for pentesting details.

2. **Interview**
   The interview should be used to define the scope for the pentesting procedure. It should also define what the security requirements for the WLANs should be. The "rules of engagement" should also be defined here. This should set strict limits for what you can and cannot test. These limitations should affect whether or not you set up a separate testing environment for executing exploits. You may also want to get information about details that may simplify the pentesting procedure. See section 4.2 and 6.12 for more details.

3. **Preparations**
   Certain aspects of the pentesting procedure will be better if you have done preparations ahead of time. These preparations are dependent on the information you get from the interview process.

   a) **Setting up a fake web-portal**
   If the clients are using a web portal to authenticate users, then you could set up a fake web-portal ahead of time. The tester will then need to know the design of the web-portal. Use the Evil Portal module to set up a fake web-portal. This tool will need you to clone the web-portals design. You will also need to write a simple backend using PHP. This code should log all credentials and accept/decline users further network access. See section 6.4.2 for more details.

   b) **Prepare WPA-/WPA2-PSK dictionary attacks**
   If the clients are using WPA-/WPA2-PSK, then you could do most of the processing work before you enter the testing area. You can get a good

lists of possible passwords. Skull Security has several different wordlists of different sizes containing previously cracked passwords. These lists can then be improved by using password-permutation-rules from John the Ripper. The interviewers will need to get information on which SSIDs that are using WPA-/WPA2-PSK. Pyrit can then use a list of possible passwords and the SSIDs to do most of the processing ahead of time. The processing you will have to do on site, once you capture a WPA-/WPA2-PSK handshake, will then be reduced drastically. See section 6.5 for more details.

4. **Wireless network discovery**
   This step should map out all the existing APs in the environment, which WLANs they belong to and what type of security configurations they have. You should also figure out which APs that have active hosts connected. Most of the WLAN exploits require active hosts. Optimally this should be done using a Wifi Pineapple with the Site Survey module or the Recon module. Optional tools are airodump-ng and Netspot/Ekahau. See section 6.2 for details.

5. **Exploit wireless configurations**
   The different security configurations will determine which steps to take next.

   a) **Open**
   In this case you can simply connect to the WLAN. You should also check that further access is not blocked by a web-portal.

   b) **WEP**
   This configuration will guarantee you access to the WLAN. You could use the airmon-ng, aireplay-ng and airodump-ng to capture WEP-IVs. The WEP-key can then be cracked using aircrack-ng. You could optionally use Wifite for the entire process. See section 6.3 for more details.

   c) **WPS**
   This configuration will also guarantee you access to the WLAN. However it will take more time than cracking WEP. Use Reaver to crack the WPS-PIN. See section 6.3 for more details.

   d) **Web-Portal**
   Set up the Wifi Pineapple as a rogue AP using the PineAP module. This module should act as the target WLAN. It should allow users from this WLAN to connect to the rogue AP. Use the Recon or Site Survey module to deuathenticate users from their current APs. You should then use the Evil Portal module that you have already prepared. Alternatively you can now set up a fake web-portal from scratch. See section 6.4.2 for more details.

e) **WPA-/WPA2-PSK**
The first thing you need to do is to capture a WPA-/WPA2-PSK handshake using the Wifi Pineapples Site Survey module. You can then use the pre-processed dictionary attack you have from Pyrit. An alternative is to use the aircrack-ng tool-suite to capture the handshake and crunch the wordlists you have. See section 6.5 for more details.

f) **WPA-/WPA2-Enterprise**
You could also try to perform a dictionary attack on WPA-/WPA2-Enterprise. This will require you to set up a rogue AP and a separate RADIUS server to capture a challenge response. If you wish to perform this attack, then you will have to find suitable tools and exact method yourself. This is because this attack was never tested during this project. See section 6.6 for more details.

6. **Check for rogue APs**
The interviewers should have a list of all APs that should be in the different WLANs. You should use the test results you get from Netspot or Ekahau in the WLAN quality procedure. You can match the registered APs in each WLAN against the list of APs you got from the clients. You will then have to investigate if there are any APs that should not be present.

7. **Get access to any WLANs you were unable to exploit**
You should get access to all the WLANs once you are done checking the wireless configurations. This is to check the level of access you can obtain from each WLAN. See section 6.7 for more details.

8. **Wired network discovery**
Wired network discovery can be done using ping-sweeps. This is to get IP-addresses and basic information about active devices you can reach from your WLAN. Ping-sweeps on one subnetwork can be done using Nmap. You can use a script that automates the process of doing ping-sweeps across multiple subnetworks. The interviewers could obtain a list of active subnetworks. This is so that you do not have to scan a lot of inactive subnetworks. The results may give you information about devices that are critical for the network security (such as firewalls). It can also indicate whether there are devices that are critical for the clients safety and daily business. See section 6.7.1 and 6.7.2 for more details.

9. **Try getting past possible firewalls**
Suspected firewalls can be scanned using the Nmap firewall options. These have different techniques used to get past firewalls. You should also use the firewalk and firewall-buypass scripts. These are Nmap scripts that have different functionality for figuring out how a firewall is configured. They will try to

figure out the rules a firewall is using and how you can bypass them. If you are able to to find a way past the firewalls, then you should return to point 8 in the procedure. Redo point 8 with alternative ping sweeps that lets you scan behind the firewall. See section 6.7.3 for more details.

10. **Perform more detailed scans of promising targets**
    You should now perform more detailed scans of the live hosts you have found. Start with devices that are likely to be critical for the clients safety and daily business. Use the different scanning techniques Nmap has. You should use scans for figuring out which operating system, services and versions the devices are running. You should also perform extended port scanning if you suspect a host of using unusual ports. See section 6.8 for more details.

11. **Test open services for default credentials**
    Different types of devices may be using default credentials. You should be able to "google" these credentials. These credentials can then be used to see if you are allowed access to the devices open services. However, the scans you have done will have to reveal enough information for you to identify what type of device you are dealing with. See section 6.9.1 for more details.

12. **Research device information to find possible vulnerabilities**
    Use the information you have gathered about the different devices to search in a vulnerability databases. You could use the web-pages of "Exploits Database" or ICS-CERT to find vulnerabilities related to your information. See section 6.8 for more details.

13. **Determine which devices you are allowed to exploit**
    You may now have a good grasp of what the different devices in the network actually are. At this point, you should confer with the interviewers, or possibly the clients. You should ensure that the devices you want to exploit actually are devices that you are allowed to exploit. This is so that you do not accidentally damage the clients safety and business. This should be clarified in the rules of engagement.

14. **Exploit vulnerable devices**
    You can now exploit the vulnerable devices that you are allowed to exploit. You can search for previously created exploits online. You can also check the exploit database in tools such as Metasploit. Alternatively you could write your own exploits. See section 6.9 and 6.10 for more details.

15. **Dictionary attack on open services (Optional)**
    If you have time, then you could try to perform dictionary attacks on the different devices open services. This could be done using separate lists of passwords and usernames. You can use Hydra or Ncrack to perform the

dictionary attacks on the different services. This is an optional part of the procedure seeing as these dictionary attacks are usually very slow. This makes it less likely for you to find the correct credentials and will eat up a lot of valuable time.

16. **Use rogue agents to gain further access**
    If you have been able to create rogue agents with your exploits, then you can try to use these devices/users privileges. These privileges may give you access to new parts of the network. You should then go back to point 8 and repeat. Note that you may not be able to use the same tools in the rogue agents. You may therefore have to write your own scripts, to perform some of the tasks, that you were able to use tools for previously.

17. **Clean up after yourself**
    If you have added any malware somewhere, or changed any configurations, then this should be changed back at this point.

18. **Perform further exploits in a controlled environment**
    You could now perform exploits in a controlled environment. You should test out any exploits that you were unable to perform in the industrial environment. This may be exploits you were not allowed to perform. It may also be to test exploits you did not have time for. It might be a good idea to to use a white-box approach at this time. This is because you may have overlooked some network details in your previous investigation. Excluding details in your test setup will make the exploit results less realistic. You will of course need to have the resources to create this type of test setup. See chapter 6 (especially section 6.1) for more details.

19. **Document your findings**
    Finally you will have to document your findings. This has not been a major focus during this project. However, every result should be documented. Most importantly, you should document if an attacker could reach any devices that are critical for the clients safety and daily business. See section 4.2 for more details.

Chapter 8
Conclusion

This chapter contains the conclusion of this project. The chapter will summarize what has been done, whether or not the project goals were reached and what possible future work there is.

8.1 What Has Been Done?

This project has quite a broad topic that covers different fields. It has therefore been necessary to test and research many different tools and methods. The main goal of this project was to create a testing procedure. To do this we first of all needed to define exactly what we were testing. Chapter 4 therefore defines the scope for our testing procedures and how to define requirements. The scope for the WLAN quality mainly focuses on WLAN throughput and coverage. The WLAN security focuses on network configurations and network segmentation.

WLAN quality testing has focused on site survey tools and methods. The tools gather raw signaling data and transform it into useful and understandable information. Most testing has been done at DNV-GLs Trondheim office outside of office hours. This was necessary to avoid disturbing daily business. Different tools have been evaluated based on what type of features they offer and how they can be used to benefit our testing procedure. They have been used for two different purposes. First, to identify areas that does not meet the clients requirements. Secondly, to try and improve problematic areas. It's important to note that the methods used to improve WLAN quality, will only make changes to the current situation. They are not intended to completely redesign networks. Research of what affects WLAN quality has been a major factor in this project. The most important theory for assessing and improving WLAN quality has been documented. This theory serves as assistance for any testers that are going to use the WLAN quality testing procedure.

Finding a good WLAN pentesting procedure has also been done through testing and research. Proper test results required a realistic industrial environment. A

lot of time has therefore been used to set up and configure such an environment. We have combined various components such as APs, firewalls, routers, switches, Programmable Logical Controllers, Human Machine Interface, miniature drilling-rig, Wifi Pineapple and test computers. Different configurations of these components has therefore been used to create realistic scenarios. The different tools have been evaluated based their features and performance. Testing has been separated into two parts. The first part tested different ways to gain access to WLANs. The second part tested what could be access and exploit once you were connected to a WLAN. Some of the methods that were used required the use of my own scripts and tools. These were used when other tools did not have the necessary features. Some of the methods, tools and scripts that were used in this project has also been used in a real pentesting project. This has been a great assistance. The results and experiences from this project served as good guidelines for the testing procedures.

8.2 Evaluation of Project Goals

There were three original project goals.

8.2.1 Goal 1

Goal 1 was to find proper tools and methods for testing WLAN quality. The tests that are used should reveal if WLANs actually meet the clients quality requirements. This goal has been accomplished by documenting different site survey tools and how they should be used. The tools and methods can first of all tell you the initial state of the WLANs, but will also be able to tell you if improvements are working. However, there are a certain amount of inaccuracy related to these results. I have experienced that it's difficult to accurately estimate if WLANs actually have the needed amount of throughput. It does not help that the information you get from clients may be inaccurate estimations. Accurate estimations will require long time traffic monitoring. The clients would therefore need to invest in network monitoring tools for accurate results.

8.2.2 Goal 2

Goal 2 was to find proper tools and methods to test WLAN security. The tests that are used should reveal if the WLANs actually meets the security requirements. This has been done by documenting tools and methods for accessing WLANs. These tests have considered different kinds of security configurations. We have then documented methods and tools for accessing and exploiting critical devices on a wired network. The exploits of wired networks illustrate what an attacker could accomplish from a WLAN connection. There are of course security issues we did not get to test.

8.2.3 Goal 3

Goal 3 was to create a step-by-step procedure that gathers the results and experiences from this project. The procedure was going to be simple and structured. This project has created two separate procedures. One for WLAN quality testing and one for WLAN security testing. The procedures have been shortened down to only contain the most important information. This way the procedures are simple and structured. Details that the readers might need are referred to at the end of a step. Including to much details in the procedure would only make it confusing and tiring.

8.3 What Value Has This Project Produced?

The value of this work does not lie in new qualitative research. The value lies in the testing procedures. These procedures are a product of results and experience using different tools and methods. People can use these procedures to create value for possible clients. The procedures are also supported by documentation. This documentation shows how to perform different parts of the procedure. Some of this documentation may be a bit detailed. This is so people, with different levels of experiences, will understand what is going on.

8.4 Future Work

There is definitely work that can be done to improve the testing procedures. First of all, the procedures need more testing in real industrial environments. This should help to refine the procedures. Experiences gained from these types of projects may give the procedures extra steps, fewer steps or it may change the content of different steps. One of the areas that needs the most work is the interview process. The information you should get from the clients are not based on real interviews. They are based on the information that was needed/beneficial to the procedures. Experience from real industrial environments should also reveal new questions that the interviewers should ask the clients.

The different professional versions of the site survey tools should also be tested properly. These versions were not tested in this project. They may prove to be very beneficial for quality testing. It would be very interesting to see how you could benefit from predictive-survey capabilities. You may also consider testing network monitoring tools, seeing as they should improve throughput calculations.

Finally, this project could benefit from further investigation and testing of WPA-Enterprise networks. Section 6.6 describes an attack that could be investigated further. If this attack is found to be a viable option, then it could be added to

the pentesting procedure. There may also be alternative methods to gain access to WPA-Enterprise networks.

References

[Acc] Acceptable signal strengths. http://www.metageek.com/training/resources/understanding-rssi.html. Accessed: 2016-11-24.

[adj] Adjacent and co-channel congestion. http://www.metageek.com/training/resources/adjacent-channel-congestion.html. Accessed: 2016-12-03.

[Air] Aircrack-ng home page. https://www.aircrack-ng.org/. Accessed: 2016-10-19.

[Ben] Hani Benhabiles. Nmap nse firewall-bypass. https://nmap.org/nsedoc/scripts/firewall-bypass.html. Accessed: 2016-12-02.

[Ber11] Dillon Beresford. Exploiting siemens simatic s7 plcs. *Black Hat USA*, 2011.

[cap] Wifi capacity planning. http://www.ekahau.com/wifidesign/blog/tag/wifi-capacity-planning/. Accessed: 2016-11-26.

[cis] Cisco asa series cli configuration guide, 9.0: Chapter: Starting interface configuration (asa 5505). http://www.cisco.com/c/en/us/td/docs/security/asa/asa90/configuration/guide/asa_90_cli_config/interface_start_5505.html. Accessed: 2016-11-14.

[Cis13] Young Kim Cisco, Patrick Croak. Site survey guidelines for wlan deployment. http://www.cisco.com/c/en/us/support/docs/wireless/5500-series-wireless-controllers/116057-site-survey-guidelines-wlan-00.html, 2013. Accessed: 2016-11-14.

[cis14] Cisco asa series cli configuration guide, 9.0: Getting started. http://www.cisco.com/c/en/us/td/docs/security/asa/asa90/configuration/guide/asa_90_cli_config/intro_start.html#96174, 2014. Accessed: 2016-11-14.

[cis15] Cisco wireless lan controller configuration best practices. http://www.cisco.com/c/en/us/td/docs/wireless/technology/wlc/82463-wlc-config-best-practice.html#pgfId-380239, 2015. Accessed: 2016-12-04.

[com] What is network address translation? http://whatismyipaddress.com/nat. Accessed: 2017-01-07.

[Cou15a] Penetration Test Guidance Special Interest Group PCI Security Standards Council. Information supplement: Penetration testing guidance. PCI Security Standards, 2015.

[Cou15b] Penetration Test Guidance Special Interest Group PCI Security Standards Council. Information supplement: Penetration testing guidance. pages 3–4. PCI Security Standards, 2015.

[DMR13] Iwona Dolinska, Antoni Masiukiewicz, and Grzegorz Rzadkowski. The mathematical model for interference simulation and optimization in 802.11 n networks. In *CS&P*, pages 99–110. Citeseer, 2013.

[Dor] Henri Doreau. Nmap nse firewalk. https://nmap.org/nsedoc/scripts/firewalk.html. Accessed: 2016-12-02.

[ekaa] Ekahau-heatmapper. http://www.ekahau.com/wifidesign/ekahau-heatmapper#!overview-0. Accessed: 2016-11-26.

[ekab] Ekahau price list. http://www.kernelsoftware.com/products/catalog/ekahau.html. Accessed: 2016-11-26.

[ekac] Ekahau price list. http://www.ekahau.com/wifidesign/training-webinars. Accessed: 2016-11-26.

[exp]

[Floa] Amrod Woodhams Florwick, Whiteaker. Wireless lan design guide for high density client environments in higher education. In *Design Guides*.

[Flob] Jim Florwick. Bitrateselection: Integration into jist/swans.

[Flo13] Amrod Woodhams Florwick, Whiteaker. Wireless lan design guide for high density client environments in higher education. In *Design Guides*, page 8. Cisco, 2013.

[Geia] Eric Geier. Security vulnerabilities of enterprise (802.1x) wi-fi security. http://www.windowsnetworking.com/articles-tutorials/wireless-networking/Security-Vulnerabilities-Enterprise-8021X-Wi-Fi-Security.html. Accessed: 2017-01-05.

[Geib] Jim Geier. How to: Define minimum snr values for signal coverage. http://www.wireless-nets.com/resources/tutorials/define_SNR_values.html. Accessed: 2016-12-01.

[hom] Ics-cert s7-300 search. https://ics-cert.us-cert.gov/advisories/ICSA-16-348-05. Accessed: 2016-12-03.

[InS] Inssider product information. http://www.metageek.com/products/inssider/. Accessed: 2016-11-29.

[Kru] Rob Krumm. Wi-fi csma/ca – going deep. https://robrobstation.com/2016/04/. Accessed: 2016-12-04.

[met] Metageek home page. http://www.metageek.com/. Accessed: 2016-11-24.

[neta] Netspot home page. https://www.netspotapp.com/. Accessed: 2016-11-24.

[netb] Netspot versions. https://www.netspotapp.com/netspotpro.html. Accessed: 2016-11-27.

[netc] Troubleshooting low snr. https://www.netspotapp.com/help/troubleshooting-snr/. Accessed: 2016-12-01.

[nma] Nmap home page. https://nmap.org/. Accessed: 2016-10-20.

[Ohm16] Magnus A. Ohm. Testing communication robustness in networked control systems. NTNU, 2016.

[pin] Wifi pineapple tetra home page. https://www.wifipineapple.com/pages/tetra. Accessed: 2017-01-10.

[plc] Plc definition. http://www.plcdev.com/definition_of_a_plc. Accessed: 2016-11-10.

[Poo] Ian Poole. Wi-fi/wlan channels, frequencies, bands & bandwidths. http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php. Accessed: 2016-12-03.

[Seca] Skull Securituy. Password lists. https://wiki.skullsecurity.org/index.php?title=Passwords. Accessed: 2016-10-20.

[Secb] KoreLogic Security. Korelogic custom rules. http://contest-2010.korelogic.com/rules.html. Accessed: 2017-01-05.

[sim] Creating the network plan manually using simulated access points. https://docs.ekahau.com/index.php/Designing_a_Wi-Fi_Network#Creating_the_Network_Plan_Manually_Using_Simulated_Access_Points. Accessed: 2016-11-26.

[Sol15] Amund Bauck Sole. Finding vulnerabilities in offshore networked control systems. NTNU, 2015.

[sub] Understanding tcp/ip addressing and subnetting basics. https://support.microsoft.com/en-us/kb/164015. Accessed: 2017-01-07.

[tam] Tamosoft home page. http://www.tamos.com/products/wifi-site-survey/. Accessed: 2016-11-24.

[Uni] Humbolt Univarsitat:Informatikk. Bitrateselection: Integration into jist/swans. https://sarwiki.informatik.hu-berlin.de/User:Ofriedri. Accessed: 2016-12-04.

[wif] Designing a dual-band wireless network. http://www.metageek.com/training/resources/design-dual-band-wifi.html. Accessed: 2016-12-03.

[wps] Wi-fi protected setup (wps) vulnerable to brute-force attack. https://www.us-cert.gov/ncas/alerts/TA12-006A. Accessed: 2016-12-11.