Home network security

Helling, S.

Award date:
2015

Department of Mathematics and Computer Science
Systems Architecture and Networks Research Group

Home network security

Master Thesis

Stef Helling

Supervisors:
dr. D.S (Dmitri) Jarnikov PDEng
dr. J.I. (Jerry) den Hartog
prof.dr. J.J (Johan) Lukkien

Eindhoven
Publishing date: August 17th, 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today almost anyone has multiple Internet connected devices at home that are able to cooperate with each other. The communication between two hosts in the network require an infrastructure that takes care of the data transfer. Since the average amount of devices within home networks is growing also the complexity of the infrastructure increases. Even by sophisticated householders it is difficult to understand an manage their home network [PCGE08]. One of the reasons for this difficulty is the effective invisibility of the network, meaning that the configurations of individual machines, parameters needed for communication with the network, and patterns of traffic flow are all hidden unless one explicitly looks for them.

1.1 Current home networks

A home network consists of a set of devices that are connected with each other within a home. Many types of devices can be connected to a home network, the type of device is not restricted any more to the 'traditional' devices such as PCs and laptops. Already devices such as phones, tablets, IP cameras, Smart TVs, game consoles, Network-attached Storage (NAS) etc. have made their introduction into our homes. All these home network devices rely on an infrastructure that takes care of the communication between the devices. An example of such an infrastructure is shown in Figure 1.1.

The home network infrastructure enables devices to connect to the Internet via a connection that is provided by a Internet Service Provider (ISP). The ISP can provide such a connection via several mediums such as: telephone line, coax cable, (A)DSL, UTP cable, and glass fiber etc. In most cases the medium is connected to a gateway device which is referred to as a modem. The modem transforms the ISP connection into an Ethernet connection and forwards it to the central router of the home network. The router distributes the network traffic and manages some administrative functions such as addressing within the home network. To connect more endpoint devices to the home network it is possible to extend the network by adding additional routers, switches, hubs, etc. These devices only extend the wired part of the home network.

Home networks are often extended with a wireless IEEE 802.11 network (also known as Wi-Fi network). This is done by deploying one or more access points into the network, as has been shown in Figure 1.1. A Wi-Fi compatible device is then able to connect to one of the access points. Wireless networks are convenient because devices are able to connect to it without user interaction, which will give the user the impression that the network connection "is just there". Another advantage of wireless networks over wired networks is that the amount of devices that can be connected is not restricted to the amount of physical available sockets.

From a security perspective wireless connections are fundamentally more insecure than wired connections. This is because wireless communication is broadcast through the air where anyone who is within the perimeter of the signal is able to receive it. Therefore there are countermeasures required to ensure the confidentiality of the communication. Fortunately for wireless communica-

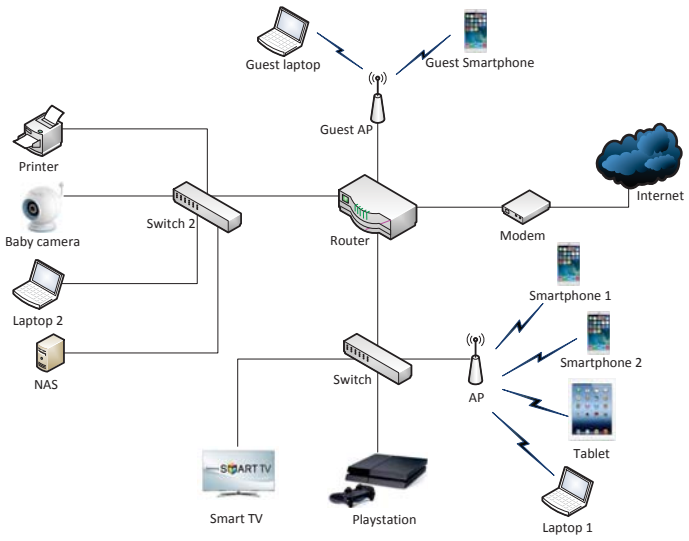


Figure 1.1: Example of a home network

tion there are already multiple techniques available.

Ideally the home network infrastructure should function fast, reliable and secure. On the one hand people want user-friendly systems which increase productivity or that entertain them. People want systems that they are easy to access and do not restrict them into what they want to do. On the other hand people want systems including its data to be safe such that the data cannot be retrieved, modified or destroyed by a malevolent person. Often solutions that secure a system set restrictions to the usage of the system. For example, to implement authentication into a system a password mechanism can be used, only when someone knows the password he can use the system. This solution already reduces the usability of the system since the user has to perform an additional action to perform its task. In general it is a challenge to combine usability with security.

1.2 Domain analysis

1.2.1 Computer security concepts

In this thesis we will refer to several computer security concepts. This section will explain and give definitions about these concepts.

Attack: An attack is an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system [Shi00].

- Active vs. passive: An "active attack" attempts to alter system resources or affect their operation. A "passive attack" attempts to learn or make use of information from the system but does not affect system resources.
- Insider vs. outsider: An "inside attack" is an attack initiated by an entity inside the security perimeter (an "insider"), i.e., an entity that is authorized to access system resources but uses them in a way not approved by those who granted the authorization. An "outside attack" is initiated from outside the perimeter, by an unauthorized

or illegitimate user of the system (an "outsider"). In the Internet, potential outside attackers range from amateur pranksters to organized criminals, international terrorists, and hostile governments.

Threat: A threat is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm [Shi00].

Vulnerability: A vulnerability is flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy [Shi00].

Systems can be vulnerable for a type of attack. Examples of what a system can be vulnerable for are: buffer overflow, SQL injection, brute force password cracking.

To be able to refer to a vulnerability the Common Vulnerabilities and Exposures (CVE) dictionary is introduced which identifies publicly known vulnerabilities within systems. Each vulnerability within this dictionary gets a CVE-ID which is an unique identifier. This identifier has the format `'CVE-YYYY-NNNNNN'`, where `YYYY` is the year that the vulnerability is discovered and `NNNNNN` is the id.

Exploit: An exploit is a sequence of actions that takes advantage of a vulnerability in a system. Exploits can be implemented as an executable binary or as an input file for a program with the aim to perform unintended behavior with a system. Typical things an exploit could be used for is gaining additional privileges within a system (i.e. privilege escalation), acting as another user (i.e. impersonation) or to just make the system unavailable with a Denial of Service (DoS) attack.

Asset: An asset can be anything that has a value for a person or organization Within computer security this can be tangible things like machines, software or data but can also be intangible like services and reputation. From a security viewpoint it is important to determine what the assets are within a system in order to know what has to be protected.

Intruder: An intruder is a person who tries to get access to a system while he is unauthorized to do so. In the news that kind of person is often referred to as a *hacker*. Between the public and the IT world there is a different interpretation of what a hacker is actually is.

Some may refer to it as a criminal person who breaks in into your computer. But there are also interpretations where a hacker is a person who is a very skilled technical person who tries to find ways to use devices for other purposes for that it is meant for. RFC 2828 [Shi00] describes a hacker as the following:

> Hacker: Someone with a strong interest in computers, who enjoys learning about them and experimenting with them. The recommended definition is the original meaning of the term (circa 1960), which then had a neutral or positive connotation of "someone who figures things out and makes something cool happen". Today, the term is frequently misused, especially by journalists, to have the pejorative meaning of cracker.

This quotation already shows that there are many interpretations of what a hacker is. They may be divided into groups upon their knowledge, on their goals or even on their ethics.

1.2.2 Existing solutions

Many solutions to, at least partially, secure a computer network are already available. They attempt to prevent and/or detect an intrusion. Prevention solutions take steps to make it more difficult for an intruder to perform an attack. Such solutions often restrict the connectivity between devices by using network firewalls deployed on user devices themselves or on an infrastructure device in the network. Also, prevention solutions focus on improving a network configuration

by properly setting up existing security measures (e.g. security protocols like WPA2 for wireless network segment). Not all intrusions can be prevented, but it may be possible that they can be detected. Intrusion detection solutions observe the behavior of systems in real-time and raise an alarm in case of a malicious event.

In the following subsections we describe several solutions that already exist.

1.2.2.1 Network vulnerability scanner

To scan if devices within a network are vulnerable for some known exploit a *network vulnerability scanner* can be used. These scanners attempt to identify a device, detect which software runs on it, retrieve the configuration of the software, and based on this information check if the system is vulnerable. In the end the vulnerability scanner will come up with an overview of the found vulnerabilities on each device and possibly provides advice about how to mitigate the vulnerabilities. In this sense a vulnerability scanner is a tool for intrusion prevention.

Vulnerability scanners are especially convenient because they can provide an overview of the security state of multiple hosts within a network. With all the devices that are connected to the home network it can be very obscure to determine if all the devices are properly secured.

In Chapter 2 we will discuss Network vulnerability scanners in more detail in Section 2.4.1.

1.2.2.2 Intrusion Detection System (IDS)

Intrusion Detection Systems (IDS), try to detect malicious behavior on a network by observing the network traffic. An IDS can be seen as a burglar alarm for a computer network, when it detects malicious traffic it can raise an alarm. IDSs consist of one or multiple sensors that are placed within the network infrastructure. Each sensor of an IDS inspects network traffic that passes by and typically propagates information about malicious events to a central system.

IDSs are not common within home networks, deploying and maintaining these systems require expert knowledge of computer networks and security. IDS will only perform well if they are deployed on the right place within the network and the system needs to be update regularly to be able to protect for the latest threats. Also the alarms that a IDS generates in case of detection of malicious behavior are less meaningful to a non-expert user. Therefore these systems are typically only seen within enterprise networks where the network is managed by a system administrator. However if the task of deploying and managing of an IDS is done by a user-friendly application, and the IDS is able to perform countermeasures itself it may be a useful extension to increase the security of a home network.

In Chapter 2 we will get back to IDS in Section 2.4.3.

1.2.2.3 Virus scanner

A virus scanner is a tool that is used to prevent, detect and remove malicious software. Virus scanners do intrusion prevention by scanning incoming data that potentially contains malicious software. Additionally virus scanners can also do intrusion detection by monitoring the behavior of the system processes.

Virus scanners require much computational resources which means that not every device within the home network is capable of running a virus scanner. These systems are designed to function in a certain way and are not able to perform additional tasks. Virus scanners are typically not installed on these low-resource devices, also the user is usually not allowed/able to install additional software on such devices. Another reason is that the user is not always allowed by the device manufacturer to install additional software on the device. To secure these devices an external intrusion detection system is preferred which monitors the device as some black-box.

A more detailed discussion about virus scanners is given in Section 2.4.4.

1.2.2.4 Firewall

Firewalls can both be deployed on an infrastructure device or on non-infrastructure devices. These systems are used to restrict network traffic according to some policy.

The gateway router typically runs a firewall since it is the entrance into the network. A gateway router firewall in home networks typically blocks all incoming traffic unless it is instantiated from inside the network. This rule protects the internal hosts to direct attacks from external hosts (e.g. make it impossible for an host outside the home network to directly connect to a host inside the home network). Another typical place to deploy a firewall in the home network is on client devices that have sufficient resources such as PCs, laptops, phones, and tablets. These firewalls are usually more sophisticated than the firewall running on the gateway router.

In Section 2.4.2 we discuss firewalls in more detail.

1.3 Problem statement

Despite the availability of prevention and detection tools, home networks continue to be poorly secured. Existing prevention solutions typically require user interaction, thus making it difficult to use by an average consumer. There are many configuration options available to configure the home network infrastructure. The fact that these configurations are distributed over multiple devices makes it even more difficult to keep an overview. For the user it is hard to see what the implications of his/her configuration changes are on the security of the network. Suppose the user connects an additional access point to its home network to extend the range of its wireless network. Then the access point first needs to be configured identical before it is able to merge with the existing network. The user has to manually look up the configuration of the existing wireless network and set this configuration to the new access point. It is possible that the user makes an error in copying the configuration to the new access point, which creates a target to attack. Based on the difficulties we identified for the configuration of a home network we develop a system that:

1. Is able to perform an assessment of the current network configuration.
2. Give recommendations based on the assessment that has been done.
3. Instrument network infrastructure devices to apply suggested configuration changes.

Assuming the network is set up properly, it is still hard to see for the user what happens on the network. Network traffic is in a sense invisible to the user, unless the user starts manually inspecting the network traffic. It is hard to see for a consumer whether networking equipment is added or removed by one occupant without the others knowing that this change had occurred [PCGE08]. For example in case of a Denial of Service attack (DoS attack) the user would only notice that the network is less responsive as usual or does not function at all. In this case it is most likely that the user assume that the network is just broken instead of suspecting that a DoS attack is going on. This is because there is no clear indication that a DoS attack is happening. As intrusion detection solution we investigate how malicious events on the network can become better visible to the user. We want to experiment if it is possible to deploy an IDS within a home network.

For both the intrusion prevention and the intrusion detection solutions must:

1. Use existing home network infrastructure devices as much as possible.
2. Usable by a consumer with average knowledge about consumer devices.

1.4 Project contribution

This project is focused on two topics: *intrusion prevention* and *intrusion detection*. To increase the security of home network we first studied its possible entry points i.e. how would an intruder get

access to a home network. We determined three entry points: *Gateway device*, *Intermediate device* and the *Wi-Fi network*. For each entry point we studied the possible attacks and preconditions on which these attacks relied. Based on that knowledge we formulated a set of recommendations to improve security for each entry point.

We introduced a model to perform dynamic security assessments of configurations which we call assessment flowcharts. The nodes in the flowchart represent some configuration setting and the connections between the nodes represent options. To indicate importance between the configuration settings a weight is assigned to each node. Each connection (which represents a configuration option) has a fraction assigned to it that indicates how optimal the option is. The assessment is done by going through the assessment flowchart and accumulating the score. If the assessment flowchart reaches an end node the accumulated score shows the security level of the system. The idea behind the assessment flowcharts is that it is implemented by a security expert and can be executed by the system. As proof of concept an Android application is designed which is able to retrieve the configuration of a DD-WRT router and make an assessment based on multiple assessment flowcharts.

Intrusion detection is done by setting up the Intrusion Detection Systems Snort and Kismet on a custom build router. Snort is used to monitor for malicious events on the Ethernet network such as port scans. To be able to detect attacks on the Wi-Fi network we deployed the wireless intrusion detection functionality of Kismet.

1.5 Thesis outline

The remainder of this thesis consists of three chapters. In Chapter 2 we analyze where current home network security lacks. This is done by providing answers to questions such as: *what needs to be protected within a home network?* or *how are home networks attacked?*. Chapter 3 describes in detail the two proposed solutions for *intrusion prevention* and *intrusion detection*. Finally we present conclusions in Chapter 4.

Chapter 2

Home network security

We want to find ways to increase the security of home networks. To do this we first have to know: *What needs to be protected within the home network?* This question makes us think about which assets are present in a home network i.e. the elements within the home network that have a value. The idea is that if we can determine the most valuable assets within the home network, then we are able to prioritize which assets to protect first. But how can the value of an asset be determined? This is a difficult question because of the assets within home networks cannot always be expressed in a value of money. Some assets are for example personal files that may very valuable for someone but are worthless for another.

Within the field of computer security it is said that perfect security does not exist. In a system there will always exist some bug that can be exploited by some party to circumvent the security mechanisms. However, the feasibility of some attacks depends on the capabilities of the attacker. Maybe it is not necessary to protect for all attackers that are able to attack a home network. There may be no reason, or too much effort required for a skilled attacker to attack the home network. For this we want to know *Who are the attackers of home networks?*

Different attackers have different capabilities, therefore they will have different approaches to intrude a home network. Some inexperienced hackers will only use existing scripts or programs to intrude systems while others may have the capabilities to develop exploits themselves. To increase the security of home networks it is good to get an idea about *how home networks are attacked?*

While new vulnerabilities of systems are discovered and misused continuously, also countermeasures for these vulnerabilities get developed. A countermeasure can be a security patch that a software developer publishes to fix a vulnerability in its program. But also solutions are developed over the years to protect other systems, examples are: virus scanners, firewalls, Intrusion Detection Systems (IDS) etc. To find out what could be done to better secure home networks it is important to know an answer on the question: *what are current techniques to secure a home network?*

In the end we would like to conclude with an answer on the question: *Where does current home network security lack?* Now we end up with a set of questions that still have to be answered. The remainder of this chapter is divided into several sections where each discusses a question. Figure 2.1 shows an overview of which questions we answer in this chapter and in which section they can be found.

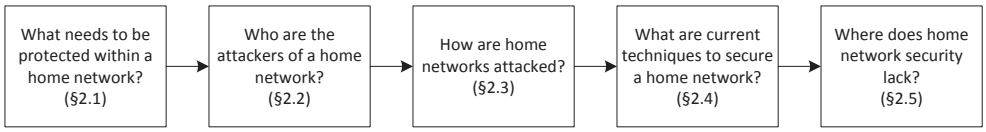


Figure 2.1: Overview of the questions that are discussed in this chapter.

2.1 What needs to be protected within a home network?

All devices that are connected to a home network can be a target for an attack. To increase the security in any system it is a good approach to make an inventory of assets of the system and identify their value. In this way becomes clear what is most important to protect within a system.

But how do we determine the value of an asset? Each asset can have its value in its own way. For the one asset it is most important that it stays secret, while for another its reliability is more important. In Section 2.1.1 we discuss so called *critical characteristics of information*, which are concepts that are used to express the value of services or information. Finally, the assets of a home network are described in Section 2.1.2.

2.1.1 Critical characteristics of information

Information has a certain value which is determined by the characteristics it possesses. For the end-user the importance of each characteristic can be different for every kind of information. To retain this value of information, these critical characteristics of information can be guarded by so-called services. A subclass of these services is the security services which only provide support for preservation of security-related critical characteristics of information. RFC 2828 defines a *security service* as a processing or communication service that is provided by a system to give a specific kind of protection to system resources [Shi00].

The next section provides a summary of popular definitions of critical characteristics of information which are related to security.

2.1.1.1 Definitions

Confidentiality: The concept of confidentiality is about keeping information secret from unauthorized actors. Multiple definitions of confidentiality are available in literature. Whitman et al. [WM11] describes confidentiality as follows:

- *Confidentiality*: Information has confidentiality when it is protected from disclosure or exposure to unauthorized individuals or systems. Confidentiality ensures that only those with the rights and privileges to access information are able to do so [WM11].
- *Confidentiality*: Stallings [Sta10] splits confidentiality into two sub-concepts:
  - *Data confidentiality*: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
  - *Privacy*: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
- *Data confidentiality*: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes [i.e., to any unauthorized system entity] [Shi00].
- *Data confidentiality service*: A security service that protects data against unauthorized disclosure [Shi00].

We see that no real consensus about the definition of confidentiality exists, mostly the scope of the concept is different. Both Whitman and RFC 2828 stress the secrecy of information for unauthorized users, where Stallings also takes the aspect of privacy into account.

Integrity: When integrity is ensured, the stored or transferred information is not unintentionally changed by an unauthorized actor or a faulty system.

- *Integrity*: Information has integrity when it is whole, complete, and uncorrupted. The integrity of information is threatened when the information is exposed to corruption, damage, destruction, or other disruption of its authentic state [WM11].
- *Data integrity*: Assures that information and programs are changed only in a specified and authorized manner [Sta10].
- *System integrity*: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system [Sta10].
- *Data integrity*: The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner [Shi00].
- *Data integrity service*: A security service that protects against unauthorized changes to data, including both intentional change or destruction and accidental change or loss, by ensuring that changes to data are detectable [Shi00].
- *System integrity*: The quality that a system has when it can perform its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation [Shi00].
- *System integrity service*: A security service that protects system resources in a verifiable manner against unauthorized or accidental change, loss, or destruction [Shi00].
- *Correctness integrity*: Accuracy and consistency of the information that data values represent, rather than of the data itself. Closely related to issues of accountability and error handling [Shi00].
- *Source integrity*: The degree of confidence that can be placed in information based on the trustworthiness of its sources [Shi00].

Stallings makes a difference between data integrity and system integrity. Both describe the lack of unauthorized or unintended modifications, but data integrity focuses on data where system integrity involve a system. The RFC 2828 does as Stallings not describe a definition of the term integrity itself, although it gives multiple more specific definitions.

Availability: The ability of an authorized user to receive the service of a system at any moment in time. There are multiple definitions of availability. The most popular of them are given below:

- *Availability*: Availability enables authorized users (persons or computer systems) to access information without interference or obstruction and to receive it in the required format [WM11].
- *Availability*: Assures that system work promptly and service is not denied to authorized users [Sta10].
- *Availability*: The property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system; i.e., a system is available if it provides services according to the system design whenever users request them [Shi00].
- *Availability service*: A security service that protects a system to ensure its availability [Shi00].

The first definition of availability given by Whitman only considers the access to information, whereas the second and third definition given by Stallings and the RFC 2828 describe availability as a system property where the system always functions as it should do.

Authentication:
- *Authentication:* The process of verifying an identity claimed by or for a system entity [Shi00].
- *Authentication service:* A security service that verifies an identity claimed by or for an entity [Shi00].

Access control: A security service that protects against a system entity using a system resource in a way not authorized by the system's security policy; in short, protection of system resources against unauthorized access [Shi00].

Non-repudiation: A security service that provide protection against false denial of involvement in a communication [Shi00].

Possession: The possession of information is the quality or state of ownership or control. Information is said to be in one's possession if one obtains it, independent of format or other characteristics [WM11].

Utility: The utility of information is the quality or state of having value for some purpose or end. Information has value when it can serve a purpose. If information is available, but is not in a format meaningful to the end user, it is not useful [WM11].

Accountability: The property of a system (including all of its system resources) that ensures that the actions of a system entity may be traced uniquely to that entity, which can be held responsible for its actions [Shi00].

Auditability: A security service that records information needed to establish accountability for system events and for the actions of system entities that cause them [Shi00].

Privacy: The right of an entity (normally a person), acting in its own behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share information about itself with others [Shi00].

Authenticity: Authenticity of information is the quality or state of being genuine or original, rather than a reproduction or fabrication. Information is authentic when it is in the same state in which it was created, placed, stored, or transferred. [WM11]

2.1.1.2 Reference models

The services given in the previous section are used in different reference models such as the CIA triad, Parkerian hexad, and RMIAS. Reference models are abstract frameworks for understanding significant relationships among the entities of some environment [CH13]. These models are focused on the security aspects of a system in order to compare the level of security in systems. Still a discussion about the completeness of each reference model is going on. Experts do not agree on which services should be present in a reference model that addresses all security aspects. The classical reference model is the CIA triad, which stands for Confidentiality, Integrity, and Availability.

Table 2.1 shows which reference model uses which aspects to assess the security of a system.

2.1.1.3 Priorities for home networking

Not all characteristics of information are equal important for home network security, and certainly not all are required to describe the level of security. We choose to use the CIA triad to describe the level of security of components in the network.

Service	CIA	Parkerian hexad	RMIAS [CH13]
Confidentiality	✓	✓	✓
Integrity	✓	✓	✓
Availability	✓	✓	✓
Non-repudiation			✓
Possession		✓	
Utility		✓	
Accountability			✓
Auditability			✓
Privacy			✓
Authenticity		✓	

Table 2.1: Overview of the services provided by various reference models.

2.1.2 Home network assets

The assets in home networks are the devices that it contains including the data they store. These devices possibly contain private data such as documents, photos, videos etc. Typically the user expects this data to remain in the state that he/she stores it (i.e. the data does not get corrupted). For all the personal data that is stored on some device in the network it is important that the confidentiality, integrity and availability remains guaranteed.

Besides that the devices store data, it is also possible that they contain sensors such as microphones and cameras. Examples are: a IP camera, smart phone, tablet, laptop etc. Once an intruder retrieves access to such a device he/she is potentially able to compromise the privacy of the user in real-time. Pages exist on the Internet that have an entire inventory of poor secured IP camera streams that are free accessible.

Besides endpoint devices in the network, also network infrastructure devices such as routers, access points, switches etc. need to be protected. Many applications exist where the user relies on secure communication between the device it's using and a server on the Internet. For instance, think about the website of a Bank or web-mail. Once an intruder gains access to a network infrastructure device, he/she can potentially create a man-in-the-middle situation where the confidentiality and integrity of the connection can be affected. For the user's experience of the home network also availability is an important aspect of the functionality of network infrastructure devices.

2.2 Who are the attackers of a home network?

To secure a system it is important to know to what kind of people we want to protect from. Who are the people that intrude a home network, and what motivations and capabilities do these people have?

People often use the term 'hacker' when they think about a person who intrudes and breaks systems. Especially in the media hackers are directly associated with criminal activities. They are portrayed as people who steal your credit card information or destroy your personal files.

In the more technical world a hacker is not always referred to as a criminal person. For example the RFC 2828: Internet Security Glossary defines a hacker as follows:

> Someone with a strong interest in computers, who enjoys learning about them and experimenting with them [Shi00].

This definition does not mention criminal activities at all. A hacker is someone who makes a system function in a way that it is not designed for.

In the following sections we describe which different different ethics, motivations, capabilities or resources hackers may have when they intrude a system.

2.2.1 Hacker ethics

Hackers can be categorized according their ethical nature. Some hackers want to improve systems by discovering vulnerabilities in systems and report these to the creators. The creators then have to opportunity to resolve the vulnerability such that it cannot be misused any more by people with malevolent intentions. Other hackers may use the vulnerabilities they found for their own good, e.g. to earn money by selling the knowledge of how to use the vulnerability.

According to ethics there are three different types of hackers:

White hat hacker: A person which might be employed to test a system for flaws [SSVE04]. It can be an employee who attacks a company's network in order to determine weaknesses, or law enforcement and intelligence agents who use their skill in the name of national security or to investigate and solve crimes. These white hat hackers do their work with the permission and knowledge of their employer.

Black hat hacker A black hat hacker is synonymous to a cracker [SSVE04]. He uses his skill in criminal and other unethical ways.

Grey hat hacker The ethical nature of a grey hat hacker is questionable, it is something between a white and black hat hacker [Pas06]. For this kind of hacker should be thought of a *vigilante* or a *hacktivist*.

2.2.2 Hacker capabilities

Different hackers have different capabilities. There is a big difference between the intrusive capabilities of your neighbor kid that is fuzzing around with freely available tools and a professional hacker. According to [Bar01] hackers can be divided into three different groups: *Script kiddies*, *Hackers*, and *Crackers*. This grouping is partly based on the knowledge level of the hacker, but also on its intentions.

Script kiddy: Someone with above average knowledge of computers but with only rudimentary knowledge about protocols used throughout the Internet. A script kiddy uses existing tools and usually gains experience and guidance from other users and may eventually belong to the *Hacker* group. The capabilities of a script kiddy are limited by the tools that are freely available. Script kiddies usually are not capable of finding new vulnerabilities and create exploits for them.

Hacker: A hacker is someone who understands the workings of the Internet and the Internet protocols themselves. He also understands the best use of some hacking tools. Originally, a hacker was thought of as a person with extreme technological talent. Nowadays the term hacker has taken on a more negative definition and is used most typically to describe a person who accesses computers and information stored on computers without first obtaining permission [Pas06].

Cracker: A cracker is someone with the capabilities of a hacker but with criminal intentions. Crackers have a goal to damage systems that are owned by other people and/or companies.

2.2.3 Hacker motivation

A hackers' high level motivation could be either to earn money, to harm the reputation of a person or company, or to satisfy own curiosity.

Barber [Bar01] describes the motivations of a hacker in the following categories: curiosity, vandalism, hacktivism, industrial espionage, extortion or fraud and information warfare. Another possible motivation of hacking a system may be that someone wants to increase its status within a hacker organization.

Curiosity: Some hackers will try to get access to a system just out of curiosity. These people do not have any intentions to cause harm to a system, but may do this because of their lack of knowledge.

Hacktivism: Is about causing damage for an ecological, political or ethical reason. A hacktivist is a digital activist who uses the Internet as a channel to reach a large audience. A typical attack a hacktivist would do is a website defacement in order to spread his ecological, political or ethical message.

Vandalism: When the hacker only has the intention to cause harm to a system without any further means we speak about vandalism.

Industrial espionage: Espionage by stealing confidential information by gaining access to an organization its systems to get commercial advantage by getting ahead of competition or to sell secrets for direct profit. The nature of the attacks used for industrial espionage is that they are stealthy. The goal is to avoid the organization from noticing that someone is breaking in their systems. For this the attacker will try to cover all its tracks. When an attacker comes in he will install back doors in order to be able to enter the system again later on.

Extortion or fraud: Organized crime syndicates want to make money by breaking into people's computers or eavesdrop communications. A hacker could obtain personal information in order to extort or hustle someone. This kind of hackers try to hide their true identities in order to impede the police organizations to trace them.

Information warfare: Hackers which are employed by government or militant wings of political parties could try to damage other countries' infrastructures or try to steal confidential information.

Increased status: A hacker could break in to a system to increase their status within a hacker community. Especially systems which are well secured or are well-known might be attacked by a hacker with this motivation.

2.2.4 Hacker resources

In the past hackers were mostly individuals which were often motivated to demonstrate their technical competence. Nowadays hackers tend to form organizations to perform criminal activities where they sell their services to companies, nations, and national government agencies. This caused that the resources and motivation available for the development of malware increased [Sta10, p. 182]

Security experts consider some malware created in the past years being too advanced for an individual to make. An example is the Stuxnet malware which was aimed to sabotage industrial installations which use Programmable Logical Controllers (PLCs).

We define three different sizes for hacker organizations:

Individual: Some hackers operate on individual basis. These hackers usually have small to no financial resources and will therefore not be able to purchase expensive equipment.

Organization: An organization of hackers can usually do more damage than a single hacker. Together they can form organized crime organizations that make money by hustling companies or individuals or form a hacktivist group. Criminal organizations want to make profit and therefore they will try to make as much money with the least effort.

Nation state: Because of information warfare government intelligence agencies are currently expanding their hacking capabilities. A nation state has compared to a hacker organization much more resources. For example, the US government National Security Agency (NSA), which is one of the world's biggest security agencies, had according to a leaked document

Profile	Ethics	Capabilities	Motivation	Resources
Curious neighbor	Grey hat, Black hat	Script kiddy	Curiosity, Vandalism	Individual
Member of a Criminal organization	Black hat	Hacker, Cracker	Extortion of fraud	Organization

Table 2.2: Hacker profiles of home network intruders.

in 2013 a budget of $10.8 billion[1]. Most governments will justify their hacking activities by arguing that it is done to protect the nation.

2.2.5 Intruders of home networks

Certainly not every hacker will be interested to intrude a home network and cause damage to it in any way. Intruding a system takes time, effort and resources, and therefore a hacker that has nothing to seek within the home network will probably not take the effort to intrude it.

In the previous sections we categorized hackers on its: ethics, capabilities, motivation and resources. But to which categories would an intruder to a home network possibly belong? If we consider the ethical categories. We consider intruding a home network in any case as unethical, therefore only black hat hackers would intrude a home network. For the aspect of capabilities all categories: Script kiddie, hacker, or cracker might have interest to intrude a home network. It is hard to say anything about the aspect of motivation to intrude a home network. This depends on the person who owns the network and what kind of data is stored within the network. From a resource perspective the most probable categories to intrude a home network would be an individual or an organization.

In Table 2.2 we describe two different profiles of hackers. We have the *curious neighbor*, which is an individual that tries to intrude the home network with free available tools. This type of hacker has few knowledge about network security and sees it as a challenge to break security mechanisms. The other type of hacker is *member of a criminal organization*. This type of hacker is a professional that has expert knowledge about network security. The main goal of this type of hacker is to earn money by for instance, committing fraud or blackmailing someone with stolen data. Since this type of hacker wants to earn as much money as possible, he/she uses preferably scalable attack.

2.3 How are home networks attacked?

To be able to attack any system an entry point has to be found by the attacker. Once the attack succeeds the number of possibilities to execute another attack is likely to increase. To attack a home network we considered the following three possible entry points:

1. Gateway router
2. Intermediary device
3. Wi-Fi network

The next sections describe for each of the entry points how the attacks take can place, and give some examples of attacks.

[1] `http://www.nytimes.com/2013/08/30/us/politics/leaked-document-outlines-us-spending-on-intelligence.html`

2.3.1 Entry point 1: Gateway router

The gateway router is the router within the network that is both connected to the Internet and the home network. In this way it is like the front door of the network. The gateway router usually has a simplistic firewall deployed that blocks all incoming traffic which does not have an active connection. A host within the home network is able to set up a connection to a host on the Internet. The firewall will open a session for this specific connection such that the remote host is able to reply on this connection. Only incoming traffic of a connection with a session will be allowed through the firewall. The firewall on the gateway router forms the first frontier for the security of the home network. However it is possible to circumvent this firewall policy by using some tricks.

Besides running a firewall and its main task of routing network traffic it also runs multiple services to support the devices that are connected to the network. Without physical access an attacker can only attack the gateway router via the vulnerabilities of the services that it runs. To support the devices within the home network the gateway router runs the following services:

DHCP To assign IP addresses within the network the gateway router runs a DHCP server. DHCP stands for Dynamic Host Configuration Protocol and serves IP addresses to DHCP clients. Typically each host within the network runs a DHCP client. When a host within the network wants to connect to a network its DHCP client sends a DHCP request to the DHCP server. After some additional negotiation messages the DHCP server will reply with a DHCP lease which assigns a certain IP address to a host for a limited amount of time.

DNS: DNS can be seen as the digital equivalent of a phone book. Instead of translating a person's name into a phone number, it translates a domain name into an IP address. DNS stands for Domain Name System. When a host wants to go to a website like `example.com` it sends a DNS lookup request to a DNS server. The DNS server will then resolve the IP address that belongs to `example.com` and reply it to the host.

HTTP/HTTPS: Consumer level routers usually run a HTTP or HTTPS server that provides a configuration web-page. HTTPS is the secured variant of HTTP, which encrypts the communication between the server and client. The router web-page enables the user to configure the router via a graphical user interface via their browser. It can be convenient to configure a router via this web-interface but it also makes it a target for intruders.

SSH: Some consumer level routers also run a SSH service. SSH stands for Secure Shell and is used to remotely control a machine via a shell where the communication is encrypted.

Telnet: Telnet is like the predecessor of SSH, it is also used to remotely control devices via a shell. However a telnet connection is not encrypted which makes it possible to eavesdrop the communication, including the login credentials.

UPnP: Universal Plug and Play (UPnP) is a set of networking protocols used to discover devices that are connected within the network.

Often in home networking there is no good reason why any service that the gateway router provides should be exposed to the Internet. It is often the case that vulnerabilities exists within any of these services. Because usually the software on the gateway router is rarely updated, these vulnerabilities remain to exist.

Once an intruder gains access to the gateway router he is able to change the configuration of the network or install a back-door to get in the future more easy access the network.

2.3.1.1 Attack 1: Multiple A record attack

The home network should not expose the configuration web-page of the gateway router to hosts on the Internet. Once an attacker gets access to this configuration web-page, he is able to lower the security of the home network or can even get full control of the device from where it is possible

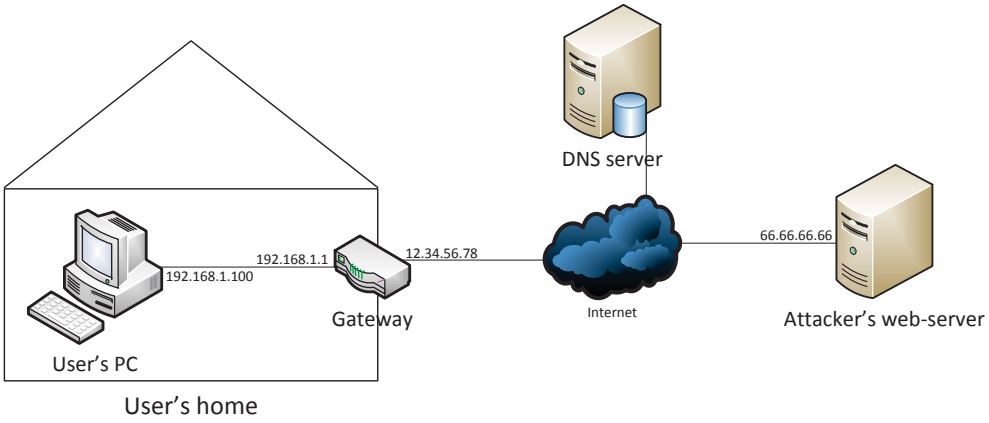


Figure 2.2: Situation of the 'Dual A record attack'.

to do even more pervasive actions. This makes the configuration web-page a target for intruders of a home network.

Suppose the goal of an intruder is to get access to the configuration page of a network but this page is only accessible from a host inside the home network. There are attacks known where an attacker still can get access to this web-page. One of these attacks is the 'Multiple A record attack' that is described by Craig Heffner in [Hef10]. It is already a fairly old attack and probably does not work with current browser which is up-to-date. However there may be devices available within the network that are still vulnerable to this attack. Think of different devices, other than a PC, that are still vulnerable for this attack since they run outdated firmware. The attack uses weaknesses in the DNS protocol and in the common implementation of the TCP/IP stack together in a cross site request forgery attack. Heffner confirmed routers manufactured by ActionTec, Asus, Belkin, Dell, Linksys, Thompson that are vulnerable for this attack, even when they run third party software such as OpenWRT, DD-WRT, or PFSense.

2.3.1.1.1 Situation

Suppose we have the situation that is shown in Figure 2.2. In the home of the user there is a home network in the `192.168.1.0/24` address range. The User's PC has only the IP 192.168.1.100. The gateway device has two IP addresses, one for the internal network: `192.168.1.1`, and one for the external network `12.34.56.78`. On the gateway device runs a web-service which serves a configuration page to configure the gateway device. The gateway device is set up such that this web-service is only reachable from within the home network.

Then the attacker has a web-server running somewhere on the Internet which serves the scripts that apply cross site request forgery. The server does not have to be owned by the attacker, it could also be some compromised (legitimate) server. At last there is some DNS server which holds the DNS record with the two 'A records'.

In a DNS record it is possible to have multiple 'A records' for the same domain name. This 'A record' contains an IPv4 address that belongs to a domain name. This functionality is used by system administrators to apply load balancing on their servers. In this attack this functionality is misused to get access to the configuration page of a home network.

2.3.1.1.2 Attack scenario

The 'A record attack' can happen as follows:

1. First the attacker has to trick the user into visiting a web-page with malicious scripts on it. Lets assume that the attacker succeeds in doing this and the user comes by the site with the malicious scripts. After the user types in the domain name of the web-page the browser will do a DNS request to retrieve the server its address. In this case the DNS server gives a DNS response with two A records, the first records contains the attacker's web-server IP: `66.66.66.66` and the second one is the external IP of the gateway device: `12.34.56.78`.

   The client's browser will then do a HTTP request to the first IP address in the DNS response and retrieves the HTML file containing the attacker its Java-script

2. Once the HTTP request is succeeded the attacker creates a new firewall rule to block further connections from that client's IP address.

3. The Java-script initiates a request to the attacker its domain via a XMLHttpRequest. This request fails because of the added firewall rule that blocks the IP address of the user. The browser of the user will receive a TCP reset packet from the Attacker its web server.

4. After receiving the TCP reset packet the browser will try to connect to the second 'A record' IP address in the DNS response. This IP address is the external IP address of the gateway device (i.e. `12.34.56.78`). Despite the destination IP of the request is the external gateway IP address, the gateway it will often still respond because the request originates from a host that is inside the home network. This is possible because of a weak implementation of the TCP/IP stack which is present in many router firmwares. The attacker its Java-script can now send requests to the router as well as viewing the responses.

2.3.1.1.3 Impact of the attack

With the *Multiple A record attack* an attacker is able to get access to the configuration page of the gateway router. If this page is not protected by an authentication mechanism the attacker will be able to change the configuration of the network. Depending on the firmware that runs on the gateway router the attacker is able to get complete control over the device. Since this gateway router has a managing task in the home network and all incoming and outgoing traffic passes through this attack is a severe security threat. In Table 2.3 is given an overview of the severity of the attack according to the CIA triad.

Although the severity of this attack is very high the scalability of the attack is very poor. The attack can only be directed to one specific network. For each to be attacked network the domain registration of the malicious server has to be modified because the second A-record has to point to the external IP address of the gateway router. Additionally the user has to be lured into visiting the attacker's web-page that serves the malicious script. Another precondition of this attack is that the user uses an old browser to visit the attacker's web-page. Current browsers are able to protect against this attack because they apply a same-domain policy on the Java-script that is executed [Hef10]. A same-domain policy enforces that the Java-script originates from the same domain, otherwise it will not be executed.

2.3.1.1.4 Recommendations and mitigations

This attack will fail in many cases because current web-browsers prevent this attack. However only one device has to be present in the home network to expose the configuration page of the gateway router. Fortunately there are some countermeasures available to mitigate this kind of attacks.

- Secure the gateway router its web-page with a strong user name and password. Even when the attacker manages to gain access to the configuration page of the router gateway a password would be required in order to make changes to the configuration. However for many router firmwares there are exploits available to circumvent this authentication.

Critical characteristic	**Severity**	**Once an attacker is able to retrieve access to the configuration page he/she is able to:**
Confidentiality	high	• If the attacker is able to get complete control over the router he/she is able to eavesdrop all incoming and outgoing traffic. • If the router contains a wireless access point it is able to retrieve the plain-text pre-shared key (password) of the Wi-Fi network. That allows the attacker to decrypt all wireless traffic that he/she captures.
Integrity	high	• If the attacker is able to get complete control over the router he/she can create a man-in-the-middle situation where it is possible to modify the traffic.
Availability	high	• The attacker is able to change the configuration of the network such that it does not function anymore. • If the attacker is able to get complete control over the router, he/she is able to recruit the system as part of a botnet to launch Distributed Denial of Service (DDoS) attacks to a host on the Internet.

Table 2.3: Severity of the 'multiple A record attack' according the CIA triad.

- Add a firewall rule on the gateway router that blocks incoming network traffic from the internal interface that has a destination IP which matches the IP address assigned to the external interface. Note that this rule will have to be updated each time the IP address of the external interface changes.
- Update the gateway router its firmware. An update may remove the vulnerability of the TCP/IP stack but will also fix other security problems.

2.3.2 Entry point 2: Intermediary device

Any device which is authorized on the home network could potentially be used as an intermediary device to get access to the network. Any device that is connected to the network becomes a potential entry point to the home network. If an attacker could get access to a device it could use the privileges of that device. Devices can store login credentials or keys in order to get access to other devices on the network such as Network Attacked Storage (NAS), IP cameras, etc. There can also be a notion of trust between devices. Some devices may trust other devices such that the other device automatically grant access to the device that it trusts. By leverage this kind of 'linked' devices there may be a chain where it is possible to circumvent authentication. However, the attacker then is still required to obtain access to the first device in the chain. Obtaining this access can be done directly or indirectly.

Directly gaining access to a remote service which is reachable from the attacker's position can be done by a direct attack. In this way the attacker can connect to the remote service directly. If a vulnerability is present in the service it can be leveraged by an exploit.

It may also be possible to use a less secured secondary channel to obtain access to a system inside a network. An example of an attack using a secondary channel is the attack on the HbbTV terrestrial signal that can be used to infect Smart TVs with malware [OK14].

Indirectly An indirect attack does not require the target host to be directly reachable. To do this malware can be used. Malware can be spread in many ways; for instance by an unaware user which visits a malicious website or opens an infected e-mail. Even if the system if physically disconnected from any network it is often still possible to attack it indirectly. The attacker can use any data carriage device such as USB-sticks or CDs to put malware on it and leave it where an authorized user will find it. An authorized user might find this data carriage device and connect it to physical disconnected system. Once this is done the system gets infected immediately.

If the attacker gains sufficient privileges on the device in order to connect to other devices within the home network, he is also able to attack these devices.

2.3.3 Entry point 3: Wi-Fi network

Wireless networks exists in many forms, you may think of the FM radio signal, the mobile phone network, or the Wi-Fi networks located in your home or office. Instead of sending the signal through a copper wire or glass fiber, the signal can be sent through air as an electromagnetic signal. An advantage of a wireless network is that it allows devices to connect to it seamlessly as soon as they come within the range of the network. Making wireless devices more portable than wired ones. One of the reasons why wireless networks are fundamentally more insecure than wired connections is that any malevolent person who is within the range of the network can receive the signals that are sent out. Whereas the communication over a wired connection cannot be captured without physical access to the wire. To provide some level of privacy on wireless communications security protocols are developed.

History shows that the security protocols used within Wi-Fi networks have not always been secure enough. The Wi-Fi networks we know these days have been improved over the years. In 1997 IEEE introduced the first version of the Wi-Fi standard which is usually referred to as

the IEEE 802.11 standard. By this standardization manufacturers were able to make devices which are compatible with each other. The first security protocol for Wi-Fi networks is the infamous Wired Equivalent Privacy (WEP) protocol. Currently multiple attacks on the WEP security protocol exist that attempt to retrieve the authentication key within minutes with high probability [BHL06, TWP07]. The security protocol WPA2 which is specified by the IEEE 802.11i standard is still considered secure enough as long a strong password is used, because there are no efficient attacks known that perform better than dictionary attacks or brute force attacks.

Once the attacker manages to retrieve the authentication key of the wireless network he is able to connect to it using any Wi-Fi compliant device. Home networks usually have no restrictions for connecting one device to another. So getting access to a wireless network would mean that the attacker gets access to machines which usually are protected by the gateway firewall. For a more elaborate description of Wi-Fi networks see Appendix B. The next section summarizes the possible attacks on Wi-Fi networks.

2.3.3.1 Classification of attacks

Multiple attacks on IEEE 802.11 networks are known that can be executed using commonly available hardware. For most attacks the only thing that is required is a PC with a wireless network card which is able to inject frames.

Attacks on wireless networks can be divided into *passive* and *active* attacks. A passive attack does not affect any network traffic on the wireless network. These attacks passively capture network traffic that comes by; which makes it hard to detect by monitoring systems because no network traffic is generated. Although, there is a possibility that a monitoring system is able to detect a passive intruder using Request to Send (RTS) and Clear to Send (CTS) frames. Usually these frames are used to determine if the medium is clear and to reserve a block of time to send the data. An RTS frame is acknowledged automatically with a CTS frame by firmware and is usually beyond the control of the user's software [LYLO03]. In this way a passive intruder could reveal its presence. Although this detection mechanism cannot determine if a device is actually capturing traffic or is just a not-connected device. It can also be any other innocent device which is within the range of the network e.g. the laptop of an (innocent) neighbor. This detection mechanism cannot distinguish between an innocent and a malicious device. Therefore this detection mechanism may not be really useful in practice.

Active attacks do affect the network traffic by actively sending frames into the wireless network. This has as consequence that active attacks are better detectable by a monitoring system. However, active attacks have more attack possibilities, they have a higher success rate and have often a shorter running time than their passive counterparts. An attacker has to estimate what the monitoring capabilities are to remain undetected.

The number of attacks that are possible heavily depend on the configuration of a wireless network. We categorize attacks on wireless networks as follows:

- De-authentication
- Eavesdropping
- Traffic injection
- Traffic jamming

This categorization is based on the categorization done in the article "Security of Wireless Local Area Networks" by Chao Yang and Guofei Gu [YG13].

2.3.3.1.1 De-authentication

The goal of de-authentication attacks is to defeat the authentication mechanism of the network. In this way the attacker can impersonate a legitimate user to get more privileges. Typical de-authentication attacks are:

- *MAC spoofing:* every IEEE 802.11 interface has a 48-bit MAC address which is a unique number which identifies the interface. Some systems deploy MAC filtering to specify which network interfaces are allowed to access the network. When MAC filtering is applied an attacker can use MAC spoofing in order use the identity of an authorized device to obtain access to the network.

- *IP spoofing:* a client device can modify the source address in the IP header of a frame. This makes it possible to circumvent security mechanisms which filter on IP address.

- *Rogue Access Point:* A rogue access point is an unmanaged and unauthorized access point which is attached to a wired network [GK]. If someone has physical access to the network infrastructure he is able to deploy a rogue access point. Within a company an employee could do this just out of his own ease. The intention of the employee does not have to be malicious but it could expose the entire enterprise network to severe security risks. Especially when the access point is not configured securely, for instance by setting it up as an open network or using a weak security protocol like WEP. The range of this rogue access point could reach beyond the borders of the company building and will be logically equivalent to an Ethernet cable hanging outside the window of the company building.

- *Brute force attacks:* Often wireless networks use a single pre-shared key to authenticate their clients. It is possible to do a brute force or dictionary attack on these networks. A brute force attack checks every possible password until it has successfully retrieved the correct password. Where a dictionary attack checks every password which is contained in a dictionary file.

  Both the brute-force and the dictionary attacks can be very time consuming because it has to try a lot passwords before the correct password is found. Although currently high performance cloud services are available such as CloudCracker[2]. This service can perform a dictionary attack in parallel on multiple machines.

- *Attacks against security protocols:* This kind of attacks tries to break the security protocol which is used by the access point. The most infamous security protocol is WEP because it has several vulnerabilities which makes it possible to obtain the authentication key within several minutes. Tews et al. [TWP07] describe an attack which is able to obtain a 104-it WEP encryption key within 60 seconds with a probability of 50%. The paper describes both an active and a passive attack, where the active attack is faster and needs less captured traffic than the passive attack to be able to have a high probability of success.

  The active attack starts by generating network traffic. This is done by capturing ARP requests from the network which come from already connected clients, where after the attacker injects these ARP requests back into the network. The AP will respond to the ARP request by sending encrypted responses.

  Despite the responses being encrypted, they still are distinguishable from other traffic because of their fixed size and their destination to the broadcast address. The first 16 bytes of clear-text of an ARP packet have a fixed pattern which is made up of a 8 byte 802.11 Logical Link Control (LLC) header followed by 8 bytes of the ARP packet itself. This property makes it possible to reveal 16 bytes of the key stream associated to a certain IV. This is done by XORing the captured ARP packet with the fixed pattern. The first 16 bytes of the outcome will contain the first sixteen bytes of the key stream.

2.3.3.1.2 Eavesdropping

This kind of attacks eavesdrop wireless traffic by compromising the data which is send over a wireless communication channel. Due to the broadcasting nature of wireless networks anyone within the range of the network is able to receive the traffic which is sent over the air. An attacker could also listen to the other traffic between the client devices and the access point. When the

[2] `www.cloudcracker.com`

network is set up as an open network the attacker is able to capture the traffic in plain text. At least when there is no other higher layer encryption used.

Typical attacks to eavesdrop:

- *Wardriving:* Wi-Fi network access points advertise their presence by periodically broadcasting beacon messages with the settings of the network. An attacker could capture these beacon messages determine the signal strength, used channel, and type of hardware used [LYLO03]. Often also the SSID of the network is broadcast within these beacon messages. Although this functionality can also be turned off to make the network stealthier. The process of gathering this information about wireless networks is called *wardriving*. It is called wardriving because it can be done while driving in a vehicle equipped with wireless network equipment. There are several tools available to perform wardriving, some of them also support the mapping of GPS data where the locations of multiple networks can be mapped. Wardriving can be used as reconnaissance in order to find wireless networks which can be attacked.

  The wardriving can either be done in a passively or in an active way. When wardriving is performed passively the discovery is performed by only listening to the beacons that are sent by the wireless access points. Within these beacons the access point announces his presence and provides connectivity information about the network. Within the beacon there are provided several attributes such as the SSID which identifies the wireless network, and the supported security protocol e.g. WEP, WPA, WPA2 etc. This passive variant is hard to detect since there is in principle no interaction happening with the network that is being detected.

  The active variant of war driving actively sends probe requests, where the access point will respond to. The access point which receives such a probe request will reply with a presence announcement beacon. Sending probe requests causes that it is on the one hand easier to detect wardriving but also that the process of wardriving becomes more accurate. Especially when performing passive wardriving from within a driving vehicle it could happen that it fails to detect one or more access points. As mentioned before, an access point periodically sends a beacon to advertise its presence. Is possible that the wardriver does not receive a beacon in the time that he/she is within range of the access point. In this case the access point remains undetected for the wardriver. The attacker is able to broadcast probe requests whereon the access point will immediately respond with a beacon.

- *Traffic capturing:* Since wireless communication is transmitted through the air, a device which is within range of the wireless network is able to capture this communication. Network interfaces normally drop all traffic that they receive but that is not addressed to them. However many network interfaces also support *promiscuous mode* which disables this dropping functionality and makes it possible to capture all traffic that reaches the network interface. There are freely available tools that can do this capturing.

  If there is used a Wi-Fi security protocol such as WEP, WPA, or WPA2 most network traffic is sent in an encrypted form (some management messages are sent in plain text). This means that the attacker needs to beat the encryption to get the plain text of the communication. In the case of WPA or WPA2 which is set up with a pre-shared key it is possible to decrypt all captured traffic if the attacker knows the pre-shared key.

2.3.3.1.3 Traffic injection

An attacker can inject traffic into the network as if it is sent from a legitimate user. In this way an attacker could send for instance re-configuration messages to the access point to weaken the security mechanisms.

- *Replay:* The attacker can capture network traffic which is sent by a legitimate device and send this same network traffic to the access point later on. This type of attack is called the

replay attack. In this way the attacker does not have to be able to decrypt and encrypt the data which is sent over the communication channel. The attacker only has to reveal what kind of data is within the encrypted data. Some security mechanisms defend against this kind of attacks.

- *Session Hijacking:* When an attacker intercepts legitimate authenticated conversations session IDs he can use the active session of a legitimate user.

  For instance, a HTTP website can use a cookie to store a session ID on the clients machine. The client machine is then able to prove with the session ID that he is the same user as before. This mechanism is required since HTTP is a stateless protocol. If an attacker is able to confiscate the session ID cookie, he is able to act as the user which is logged in.

- *Man-in-the-middle attacks:* This kind of attack is done by making two communicating parties communicate through the attacker. This attack is best explained using an example. Suppose there are two parties Alice and Bob which want to communicate. The attacker Eve performs a Man-in-the-middle attack. Eve convinces Alice that she is Bob and also convinces Bob that she is Alice. In this way the parties Alice and Bob think that they are communicating with each other, without being aware that Eve is in between. The parties Alice and Bob have the perception that they are communicating over a private channel. When the attacker is in the man-in-the-middle position he is able to capture all the transmitted data between the parties, but also intercept, modify and impersonate the communication.

  The attacker can perform the man-in-the-middle attack either from a rogue access point or via a client device which is connected to the network where all traffic is relayed to.

2.3.3.1.4 Traffic jamming

Traffic jamming is a category of attacks where the availability of the wireless network is affected. This category of attacks is also often referred to as *Denial of Service attacks (DoS)*. DoS attacks on wireless networks can be done in various ways. With the right equipment an attacker is able to send noise on the radio frequency of the wireless network. In this way the frames sent over the air become corrupted and will not reach their destination.

When the attacker is connected to the network it could also perform a DoS attack by spamming messages into the wireless network. These messages generate a lot of network traffic which utilizes a lot of bandwidth. Other users on the network will then notice that they have no or a slower connection.

It is also possible to send malformed packets into the network which keep the access point busy or even disable an access point for a particular amount of time.

2.3.3.2 Attack 3: Evil-twin attack

A specific attack against the authentication mechanism of the WPA2 security protocol is the *Evil-twin attack*. Basically it is a brute-force attack against the authentication mechanism, but the special thing about this attack is that the wireless network that is under attack does not have to be in range. Only a device that previously has connected to the wireless network is required to be able to do a brute force attack to retrieve the network's pre-shared key.

2.3.3.2.1 Situation

As mentioned before, to retrieve the pre-shared key of a WPA2 network, the network itself does not have to be present to execute the evil-twin attack. Only a victim device that has connected to it in the past is sufficient. Another requirement for the attack to be successful is that the victim device needs to be in discovery mode (i.e. unconnected and searching for Wi-Fi networks). Figure 2.3 illustrates the situation that is required for the evil-twin attack.

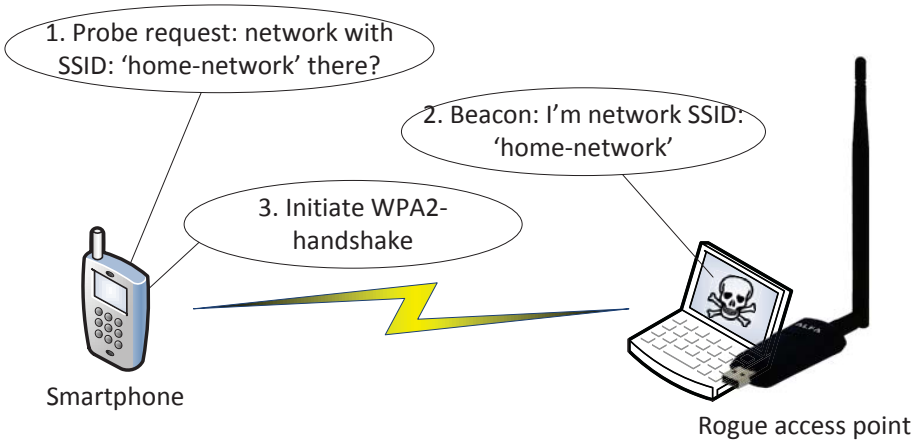


Figure 2.3: Situation of the Evil-twin attack on WPA2

2.3.3.2.2 Attack scenario

This attack is made possible because wireless devices are very willing to connect to a network with a known Service Set Identifier (SSID) without user interaction. The SSID of the network is the human readable name of the network. When the wireless network is not within range, the first objective of the attacker will be to retrieve a SSID where a wireless device has connected to before. If a wireless device is not connected to a wireless network, but it is searching for wireless networks, then it goes into discovery mode. During this discovery mode the wireless device sends probe requests that include a list of the wireless networks they have been connected to. Access points will immediately respond on these probe requests with a beacon. However, an attacker is also able to capture these probe requests. Once the attacker has captured a probe request of a wireless device he/she knows the networks where that wireless device previously has connected to.

The attacker now has to set up a rogue access point with a SSID that occurred in the list of the probe request in combination of the same security protocol used. Note that the attacker does not know the pre-shared key. If the wireless device discovers the rogue access point with the known SSID it will often try to connect to it automatically (without user interaction) [DZM05]. When the wireless device attempts to connect to the rogue access point there is initiated a WPA-handshake, which will fail because the rogue access point does not obtain the pre-shared key. For more detailed information about the WPA-handshake see Section B.1.4 of Appendix B. With the captured (failed) WPA-handshake the attacker is already able to do an offline brute force attack to retrieve the password.

2.3.3.2.3 Impact of the attack

The most disturbing part about this attack is that it can be executed without the Wi-Fi network to be present. Many people carry around their phones with Wi-Fi enabled. These devices are very willing to connect to Wi-Fi networks while there is poor verification of the access point.

Suppose the evil-twin attack is successful and the attacker has retrieved the pre-shared key of the Wi-Fi network. The severity of this according to the CIA triad is shown in Table 2.4.

The scalability of the evil-twin attack is poor. It may be possible to set up a rogue access point that continuously:

1. monitors for probe requests containing SSIDs;
2. sets up WPA2 networks with the discovered SSIDs;

Critical characteristic	Severity	Once an attacker is able to retrieve the pre-shared key of the WPA2 network he is able to:
Confidentiality	high	• Connect a device to the wireless network. This device can then be used to attack other devices within the network. • Decrypt all wireless network traffic.
Integrity	high	• Set up a rouge access point where the devices will successfully connect to. Then the rogue access point is able to create a man-in-the-middle situation.
Availability	low	• Connect a device to the wireless network that used the bandwidth of the network.1

Table 2.4: Severity of the 'evil-twin attack' according to the CIA triad.

3. waits for the devices to initiate a WPA handshake such that it can be captured.

However, once a WPA handshake is captured it still needs to be brute forced, which is resource intensive.

2.3.3.2.4 Recommendations and mitigations

This attack requires that Wi-Fi devices are in discovery mode. By turning off Wi-Fi functionality when it is not used, this attack will not be successful. However, this might not be a desirable countermeasure because it is not convenient. People may also forget to turn it off. For some devices it is possible to turn off automatic connecting to Wi-Fi networks. Then the user has to explicitly approve that the device connect to a Wi-Fi network.

Since the WPA handshake is cracked using a brute force / dictionary attack choosing a strong password as pre-shared key will mitigate the vulnerability of the attack. What a strong password is, is described in Section 3.1.1.3 in the next chapter.

2.4 What are current techniques to secure a home network?

In this section we discuss what the current techniques are to secure a home network. Section 2.4.1 summarizes some of the available vulnerability scanners. In section 2.4.2 firewalls are discussed. Intrusion Detection Systems are treated in Section 2.4.3. Finally, we discuss the use of virus scanners in Section 2.4.4.

2.4.1 Vulnerability scanners

A vulnerability scanner does what the name suggests: it scans for vulnerabilities within the system and makes an assessment based on its discoveries. This kind of tool may also be of interest for intruders, since they can point out the weak spots of a system. Vulnerability scanners have many forms, each focusing on a particular aspect of a system (e.g. network, web applications, database, host-based etc.). The kind of vulnerability scanners we are interested in are the ones which could be useful within a home network. These include the network-based vulnerability scanners and the

Name	Type	Description	Commercial / open-source
Nmap	port-scanner	Nmap scans for ports which are open on a specific host. It can do host discovery on a network and also version detection of running services or operating system. This tool is often used as a reconnaissance tool to find entry points into hosts in a network.	open-source
Nessus	network vulnerability scanner	Nessus is currently a product from the company Tenable network security. They provide different variants of their vulnerability scanner ranging from products that assess home networks to big corporate networks. Nessus originally started as an open-source project but was closed by Tenable.	commercial
OpenVAS	network vulnerability scanner	OpenVAS started as a fork of the Nessus scanning tool after Tenable closed the source of the Nessus project. The security scanner is accompanied with a daily updated feed of Network Vulnerability Tests.	open-source
Nexpose	network vulnerability scanner	Nexpose is a commercial vulnerability scanner developed by the company Rapid7, which is also the company behind the Metasploit framework. Nexpose has different editions which vary from Enterprise edition to a free community edition.	commercial
Qualys vulnerability management	network vulnerability scanner	The company Qualys also provides a solution for vulnerability management. Qualys delivers their product as a service from cloud datacenters. In this way they provide a service which they claim to be always up-to-date.	commercial

Table 2.5: Some existing vulnerability scanners

host-based vulnerability scanners. Table 2.5 shows some of these vulnerability scanners that are available.

2.4.2 Firewall

A firewall is a network component that filters network traffic that passes through it. It decides whether to drop or allow certain traffic based on a policy. A policy consists of a set of rules which specify whether a certain type of traffic should be allowed or dropped. A firewall can be either be classified as stateless or stateful.

Stateless: The first firewalls were designed as static packet filters. These firewalls analyzed per packet if it was allowed or dropped. A stateless firewall is a static filter which does not maintain any information about the connection. It is the simplest variant of a firewall with limited capabilities.

Stateful: When a firewall keeps track of the open connections and tries to find patterns in these streams of packets we call it stateful. With stateful firewalls more complex rules can be specified such as: only if a connection is set up from host A to host B then the packets that host B replies to host A are accepted.

Firewalls can also be categorized into the different abstraction layers they work on. The one firewall only considers one packet at a time while determining if the packet should be dropped or not while the other takes an entire conversation of packets into account.

	Network layer	Transport layer	Application layer
Stateless	x		
Stateful		x	x

Table 2.6: Overview of OSI-layers and stateless/stateful firewalls.

Network layer A network layer firewall filters network traffic on package level. If the incoming packet does not match an allowed pattern it will be silently dropped or rejected. This type of firewall does not maintain a state of a connection, which means that every packet is treated separately. The information used to filter the packets often includes the source or destination IP address, and the destination port number.

Transport layer A firewall that works on the transport layer filters traffic on connection level. In this way the firewall is able to maintain a state of the connection and determine relations between the packets. This kind of firewalls can be set up such that incoming traffic is only accepted if the connection is initialized from the private network.

Application layer The most extensive firewalls are able to filter traffic on application layer. An application layer firewall is able to interpret application layer protocols and build a state based on the protocol.

In Table 2.6 is shown an overview of on which layer stateful and stateless firewalls have to process the traffic.

Often a firewall is placed in between the private network and the Internet. Within home networks the firewall is usually present on the gateway device since it is exactly on the border between the private network and the Internet. This firewall is then able to filter the traffic that enters and leaves the private network.

A typical firewall rule is as follows: deny all incoming traffic originating from the external network (i.e. the Internet) unless it is instantiated from an internal host. Figure 2.4 illustrates this rule. The firewall is shown as a red wall, where the host left to the firewall represents the internal network. On the other side of the firewall is the external network shown as the Internet. Above the dotted line we can see that when the external host tries to contact the internal host directly the firewall will block this request. The other case is shown below the dotted line, if the internal host first sends a request to the external host this will be allowed and remembered by the firewall. Once the external host receives the request it sends a response back to the internal host. When the response reaches the firewall, the firewall will remember that there has been an internal request in advance and allow the response.

There is the possibility to deploy a firewall on a host within a network. These host-based firewalls only monitor the traffic that go to, or originate from that specific host. An approach is to put firewalls on each host within the network which would make infrastructure based firewalls unnecessary. Although this approach is not feasible within home networks since there are devices which do not run a firewall or are not even capable of running it. Also managing the policies of all the host-based firewalls have to be managed such that internal traffic is still possible.

2.4.3 Intrusion Detection System

Intrusion Detection Systems (IDS) are used to monitor a system for intrusions. IDSs are the 'burglar alarms' or 'intrusion alarms' of computer security. When an intrusion is detected by the IDS it will cause an alarm. This alarm is then delivered to someone who can respond and take appropriate actions like ousting the intruder or calling external authorities [Axe00].

An intrusion detection system can monitor different aspects of a system. The IDS can monitor the host its system and/or application logs in order to detect intrusions, but also by examining network traffic.

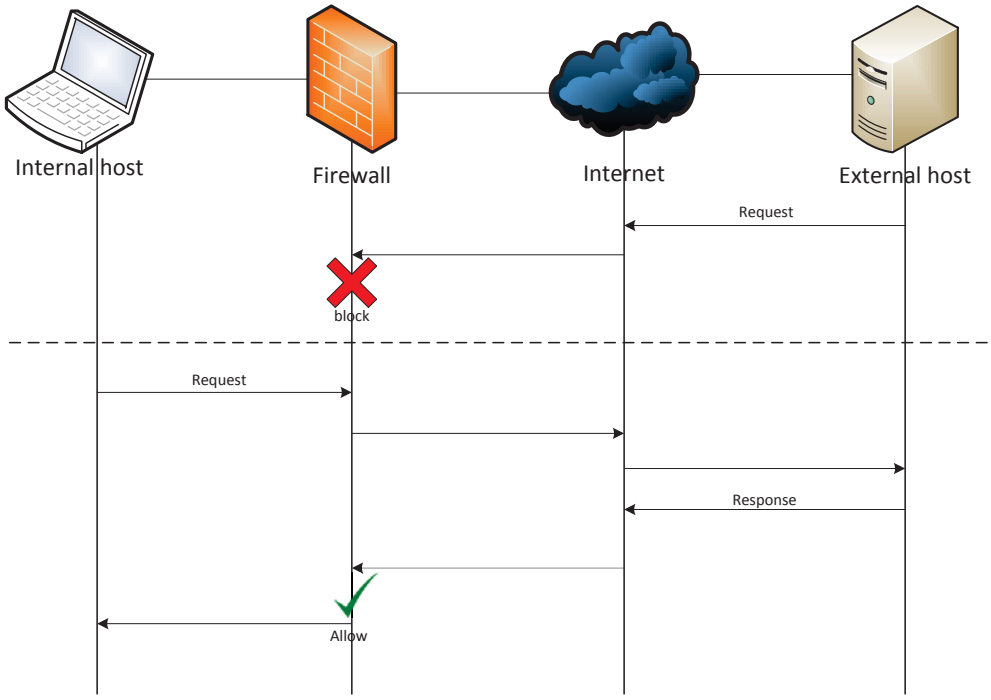


Figure 2.4: Example of a firewall rule: "deny all incoming traffic originating from the external network unless it is instantiated from an internal host".

Liao [LLLT13] divides IDS in four different classes. Each class recognizes a different type of suspicious activities and does their monitoring in a fundamentally different manner.

Host-based IDS (HIDS) Monitors and collects the characteristics on hosts containing sensitive information, servers running public services, and suspicious activities on a host. This kind of IDS monitor the logs on a host in order to detect if malicious actions take place. An HIDS is the only type of IDS that can analyze end-to-end encrypted communications, because its host is one of the end points of such communication channel. Even though it is possible to monitor all incoming and outgoing traffic on the host, it is hard to detect intrusions accurately since usually the network based context is lacking. Running the IDS on the host has as disadvantage that it consumes the hosts' resources such as CPU and memory.

Network-based IDS (NIDS) A NIDS monitors network traffic on the wired network where it can either operate *inline* or *passive*. An *inline NIDS* is placed on an intersection point of the network. Traffic is being inspected before it is allowed to pass the intersection. This causes additional network latency and reduced throughput. On the other hand inline NIDS have more possibilities to actively block certain traffic than passive NIDS.

A passive NIDS eavesdrops a communication line and searches for intrusions in this way. This has as consequence that NIDS have fewer possibilities to block unwanted traffic. On the other hand it less affects the network latency and the throughput.

Wireless-based IDS (WIDS) A Wireless-based IDS is a network monitor which detects attacks on a wireless network. To do this these WIDS capture the wireless network traffic and inspect it for malicious activities. A WIDS usually consists of multiple 'sensors' which are installed within the range of the wireless network where they have a dedicated task to capture and analyze the traffic. Disadvantages of these WIDS sensors are that they cannot avoid evasion techniques and that they are susceptible for jamming attacks.

Network Behavior Analysis (NBA) An NBA system searches for abnormal traffic flows which could indicate that an attack is being executed.

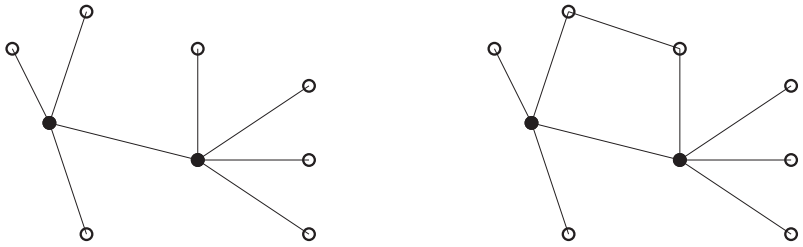
Mixed IDS (MIDS) The MIDS is a combination of any of the previous four IDS types. Combining the abilities of multiple IDS variants makes it possible to increase the effectiveness of the IDS.

Each of these classes can be implemented using their own detection methodology. There are basically three common approaches to do detection of intrusion: *anomaly-based detection*, *signature-based detection* and *stateful protocol analysis*.

Anomaly-based detection: Applies statistics on network traffic in order to distinguish between normal and abnormal behavior. In this way anomaly-based detection can detect new attacks. The downside of this kind of detection is that it is susceptible to high false positive rates (i.e. raising an alarm when there is no attack).

Signature-based detection: Is a detection based on pre-determined signatures of attacks. Before an attack can be detected there has to be a matching signature available within the IDS. Therefore the performance of a signature-based IDS heavily depends on a well defined security policy. Attacks which are unknown to the IDS database will not be detected. The advantage of signature-based detection is that it has typically a lower false positive ratio than anomaly based detection.

Stateful Protocol Analysis: To be able to do more advanced intrusion detection, the state of network traffic can be tracked. An IDS with stateful protocol analysis is able to track connections and tries to find unusual behavior in it. Both anomaly-based and signature-based detection do not maintain any kind of state of the network traffic.



Vertex cover No vertex cover

Figure 2.5: Example of a graph of network topology with IDS sensors.

An IDS uses either one of these detection methodologies or combines multiple methodologies in order to get the best of all worlds. The common drawback of an IDS is that it does not provide a totally accurate detection of intrusions. There will be intrusions possible which will remain undetected. An IDS can have a high detection ratio, but if it has a high false positive ratio the alarms which are generated will be less meaningful.

Another reason that an IDS is not always able to provide a totally accurate detection for intrusions is that it is unable to monitor all the traffic in a network. The IDS cannot monitor network traffic that does not pass through its interface. For example if the communication between two devices has a route around the IDS.

Suppose we have a wired computer network and we want to determine if the sensors of an IDS are able to capture all network traffic. We can then construct a graph of the topology of the network and indicate for each node whether an IDS sensor is deployed on it. If there exist two different non-IDS-nodes that have a route between them without passing an IDS-node then there is a possibility that some traffic is missed. This problem can be reduced into determining if the IDS sensors form a vertex cover in the network topology graph. A vertex cover is a set of vertices of a graph, such that each edge of the graph is incident to at least one vertex of the vertex cover. Figure 2.5 shows two example network topologies with IDS sensors. The vertices in the graphs represent the hosts in the network and the lines indicate whether these hosts can communicate. A filled vertex indicates that the host runs an IDS sensor which is able to monitor all the traffic on its interfaces. The graph with the vertex cover is able to capture all the traffic which flows through the network. Because of the extra edge in the right graph it is possible that the two top hosts communicate directly without an IDS sensor is able to inspect the traffic.

Do we need to monitor all the traffic in the network? It would make it possible to form a complete state of the network traffic, but it may be unnecessary to reach a sufficient detection rate. Monitoring all network traffic will require IDS sensors on many places which will require much more resources than normally is available on consumer level network equipment. However Hugelshofer show in [HSHR09] that it is possible to run an IDS on high-end consumer level network equipment with a restricted set of patterns where the IDS looks for.

For Wireless Intrusion Detection Systems it is possible to cover the range of the wireless network with sensor nodes which capture and examine the traffic that is sent through the air. Even when the wireless traffic is encrypted by a security protocol like WPA2 still some intrusion detection is possible. Some attacks on IEEE 802.11 networks try to impersonate the access point (AP) by injecting management frames with a source MAC address of the AP. Monitoring of this traffic is possible because the signals are broadcast through air.

Table 2.7 shows an overview of some well known intrusion detection systems.

2.4.4 Virus scanners

A Virus scanner is a tool to prevent, detect and remove malware. Malware is software that has the intent of compromising the confidentiality, integrity or availability of data and applications.

Name	Class	Description
Snort IDS	NIDS	Snort is an open source network intrusion prevention system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts.
Suricata	NIDS	Suricata is an open source Network IDS, IPS and Network Security Monitoring engine. It is owned by a non-profit foundation, the Open Information Security Foundation (OISF).
Samhain	HIDS	The Samhain host-based intrusion detection system (HIDS) provides file integrity checking and log file monitoring/analysis, as well as rootkit detection, port monitoring, detection of rogue executables, and hidden processes.
OSSEC	HIDS	OSSEC is an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response. It runs on most operating systems, including Linux, MacOS, Solaris, HP-UX, AIX and Windows.
Bro Network Security Monitor	NIDS	The Bro Network Security Monitor is a network monitoring framework which can be used to build a NIDS. Additional features are collecting network measurements, conducting forensic investigations, and traffic baselining.
Kismet	WIDS	Kismet is an 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. Kismet identifies networks by passively collecting packets and detecting standard named networks. It is also capable detecting hidden networks.

Table 2.7: Existing intrusion detection systems

Malware exists in many forms, a summary the different types of malware is given in Appendix C.

Virus scanners use (or combine) different detection techniques which be divided into three categories [IM07]:

Signature-based Signature-based detection attempts to model the malicious behavior of malware and uses this model in the detection of malware. This model of malicious behavior is often referred to as a signature. For this detection approach there is a repository required that contains a collection of all the signatures. Currently, to create these signatures there is human expertise required. One of the major drawbacks of the signature-based method for malware detection is that it cannot detect attacks where no signature is stored in the repository.

Anomaly-based The anomaly-based detection technique tries to distinguish between normal and malicious behavior by observing how the majority of the programs in the system behave, and try to detect anomalies in this behavior. Usually, anomaly-based detection occurs in two phases: a training phase and a detection phase. During the training phase the detector attempts to learn the normal behavior. After the training phase the anomaly detection method goes into a detection phase where it monitors the behavior based on its perception of normal behavior.

The big advantage of anomaly-based detection is that it is able to detect malware that is not known in advance. However, there is typically a high false alarm rate associated with most anomaly-based detection techniques.

Specification-based Specification-based detection techniques are a special type of anomaly-based detection techniques. The difference is that specification-based techniques leverage some specification or rule set of what is valid behavior in order to decide the maliciousness of a program under inspection. Programs violating the specification are considered anomalous and usually, malicious. With this technique there are less false alarms in comparison with anomaly detection.

In specification-based detection, the training phase is the attainment of some rule set, which specifies all the valid behavior any program can exhibit for the system being protected or the program under inspection. The main limitation of specification-based detection is that it is often difficult to specify completely and accurately the entire set of valid behaviors a system should exhibit.

The capabilities of a malware detection technique can be expressed in how many cases the technique makes the correct and wrong decision. When an item is scanned by the detection technique we have two cases: the item is infected with malware, or it is not infected with malware.

True positive Malware is present, and the detection technique detects malware. In this case the detection technique is correct, so the rate of true positives should be as high as possible.

False positive Malware is not present, but the detection technique detects malware. In this case the detection technique is incorrect. The ratio of false positives should be as low as possible. However, a high false positive ratio is less severe than a high true negative ratio since there is no malware present and only an erroneous warning is raised.

True negative Malware is present, but the detection technique does not detect it. In this case the detection technique is incorrect. The ratio of true negatives should be as low as possible and even lower than the false positive ratio.

False negative Malware is not present, and the detection technique does not detect it. In this case the detection technique responds correctly, therefore this ratio should be as high as possible.

Detection technique	True Positive	False Positive	True Negative	False Negative
Signature-based	low	low	high	high
Anomaly-based	medium	high	medium	low
Specification-based	medium	medium	medium	medium

Table 2.8: Detection ratios per malware detection technique

Table 2.8 shows the detection ratios per detection technique. The performance of signature-based techniques heavily rely on the used signature repository, because it is only possible to create signatures for known malware the true positive ratio is low. Although when there is no malware present, the signature-based detection technique will make few wrong decisions. Anomaly-based detection techniques perform better in detecting malware, although this detection technique is not water-proof. The biggest disadvantage of anomaly-based detection techniques is that the false positive ratio is big, which could make its detections less valuable. Further, we can see that specification-based detection techniques are a sort of compromise. Compared to signature-based detection techniques, specification-based detection techniques perform better on the true positives and the true negatives. Specification-based detection techniques also perform better than anomaly-based in a sense that they have a lower false positive ratio.

We see that no detection technique performs perfect, however for signature-based techniques, when putting much effort into creating signatures it may be possible to obtain a higher true positive ratio. Combining of detecting techniques may also increase the overall detection performance.

2.5 Where does current home network security lack?

One of the problems with home network security is that, because of the lack of knowledge of the user, often the available security mechanisms are not fully utilized. To make it easier for a user to configure its home network there is a need for a tool that can inspect the home network like a security expert. Based on its inspection this application could give recommendations that explain how to improve the security, or even better, applies the improvements themselves. This intrusion prevention approach is discussed further in Section 2.5.1.

Suppose that the security mechanisms of a home network are configured optimal. Still malicious events are possible. To be able to detect these malicious events active monitoring is required, which brings us to the field of intrusion detection, described in Section 2.5.2.

2.5.1 Intrusion prevention

A decent configuration of the home network can already prevent many intrusions. It may still be possible to intrude the home network with a decent configuration, but then it is likely that the attacker considers attacking the network as too much effort. However, configuring the home network is not an easy task for a consumer. The home network has many settings and the configuration can be distributed over multiple hosts in the network. It is important that all the devices are configured properly because one device can already make the entire network vulnerable for attacks.

To get an overview of the weak spots in a network there exist network vulnerability scanners. These tools determine for each host which operating system is running, which other software runs on the device and which services the host provides to the network. Once this scanning of hosts is finished a detailed report is generated. Unfortunately this report is often incomprehensible for the user because it is meant for an expert such as a system administrator.

Suppose the user is able to identify the weak spots in the configuration of the network, then still remains the task of fixing the configuration. The user may not know how and where to solve these flaws in the configuration. This task might be taken over by a configuration tool. There is

(by our knowledge) no tool available that combines the functionality of a network vulnerability scanner with instrumenting network infrastructure devices.

2.5.2 Intrusion detection

In home networks the task of intrusion detection is often done on hosts by virus scanners and firewalls. Unfortunately not every host within the home network is able to run a virus scanner or firewall. These hosts rely on a secure network.

However, the home network typically does not have any monitoring tools available that can take countermeasures or at least notify the user in case of a malicious event. Systems that are able to monitor network traffic for malicious already exist as Intrusion Detection Systems (IDS). Typically IDS are used in large enterprise networks. Current consumer level network infrastructure devices such as routers, switches and wireless access points have a dedicated task and are typically not able to perform additional tasks such as running an IDS. Next generation network infrastructure devices will be more powerful which would make it possible to run an IDS on the infrastructure of the network.

Chapter 3

Intrusion prevention & detection

To improve home network security we took two approaches. The first approach was to prevent intrusions by improving the configuration of the home network. A proof-of-concept application is developed that is able to retrieve and assess the configuration of a home network. Section 3.1 describes how the configuration is retrieved, and assessed.

The second approach was to explore the possibilities of an intrusion detection system (IDS) within a home network. Even if a home network is configured well, then it is still possible that malicious events happen. For example, a malware infected device is connected to the home network. The infected device is able to discover other hosts on the network and infect these devices. For an intruder that has gained access to the home network, a logical first step would be to explore which hosts are connected to the network, and which services these devices run. Tools that do this are port scanners, such as Nmap [1]. We want to see how an IDS can be deployed within the home network to detect such a port scan. Besides that we also want to use a wireless intrusion detection system (WIDS) to detect attacks on the Wi-Fi network. The attack that is chosen for the attack scenario is the de-auth attack. How this de-auth attack works, and how the IDS and WIDS are deployed is described in Section 3.2.

The global overview of the use cases that are addressed in the implementation are shown in Figure 3.1. There are three use cases for the (authorized) user: check configuration, change configuration and active monitor.

[1] Nmap web-page: `https://nmap.org`

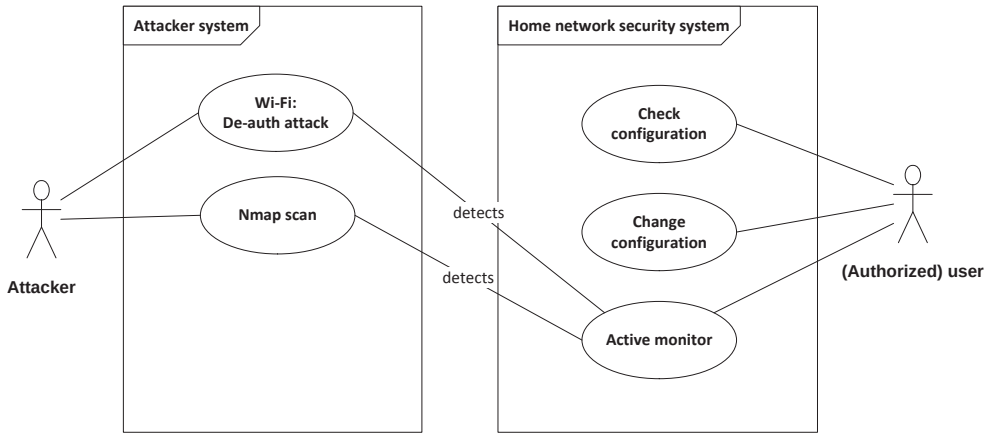


Figure 3.1: High level use cases of the implementation.

3.1 Intrusion prevention

3.1.1 Configuration assessment

To determine if a system is secure an approach is to act as an intruder, and see which actions are possible that bring the intruder closer to its goal. Performing an action might enable other actions. It can be seen as a game that has as goal to intrude a system using a sequence of actions. The 'easiest' sequence of actions that exists to reach some intrusive goal determines the level of security within the system. This high level explanation is the idea behind *attack graphs*. Attack graphs are used to determine for each system state which actions are possible that move the system into another state. In Section 3.1.1.1 we discuss attack graphs and how these can be applied within home networks.

Despite that the idea of attack graphs is appealing. We found that modeling a home network accurately enough to determine which exploits are applicable is too difficult. Additionally, attack graphs rely on an up-to-date vulnerability database where is determined accurately, in which state which vulnerabilities are present. Unfortunately, to our knowledge, there is no such database freely available. Therefore is decided to look for other approaches to assess a configuration.

The different approach is to first do a manual security analysis on a certain aspect of home network security, and then construct an *assessment flowchart* based on this analysis. Assessment flowcharts are introduced as a model to assess some configuration of a system, and are further discussed in Section 3.1.1.2.

With both attack graphs and assessment flowcharts an assessment can be made of configuration. In many cases a password is one of the settings that is part of a configuration. Password authentication is often used to guard the access to the system, which makes the security of the system dependent on the strength of the password. To secure a system that uses password authentication well, it is important to choose a strong password. But what is a strong password? Some passwords might be easy to be guessed by a human, while others can be retrieved fast by a computer that tries all possible passwords. Multiple techniques to determine the strength of a password already exist. Section 3.1.1.3 explains some techniques to determine the strength of a password.

3.1.1.1 Attack graphs

An attack on a system often consists of multiple actions. Each action exploits a certain functionality or vulnerability in the system to change the state of the network such that another exploit can be applied. Exploits often require very specific preconditions in order to be applicable. An intruder will try to increase his access to the system in order to enable more exploits. In the end, after a chain of exploits the intruder may reach its goal. With the following attack model which was based on the description of Sheyner [She04] we try to create a model to find all possible attack chains. From there we determine which chains pose the biggest weaknesses in the system and formulate recommendations to mitigate these weaknesses.

We define the *Attack model* $\mathcal{W} = (S, \tau, \{s_0\}, S, S, D)$ as a Büchi model. A Büchi model is defined as follows:

Definition Given a set of atomic propositions AP, a Büchi automaton over the alphabet $A = 2^{AP}$ is a 6-tuple $\mathcal{B} = (S, \tau, S_0, S_a, S_f, D)$ with finite state set S, transmission relation $\tau \subset S \times S$, initial state set $S_0 \subset S$, a set $S_a \subset S$ of acceptance states, a set $S_f \subset S$ of final states, and a labeling $D : S \to 2^A$ of states with sets of letters from A.

Within the system we define three agents $\mathcal{I} = \{E, D, S\}$. Where E is the attacker, D the defender and S the system under attack. Each agent $i \in \mathcal{I}$ has a set of possible actions A_i which it can execute. The total set of actions is defined as $\bigcup_{i \in \mathcal{I}} A_i$.

The attack model $\mathcal{W}$ of a home network consists of the following components:

1. H, a set of hosts connected to the network.
2. C, a connectivity relation expressing the network topology and inter-host reachability.
3. T, a relation expressing trust between hosts.
4. I, a model of the intruder.
5. A, a set of individual actions (exploits) that the intruder can use to construct attack scenarios.
6. Ids, a model of the intrusion detection system.

Each host $h \in H$ is defined as a tuple $(id, svcs, sw, vuls)$. The id is an unique identifier for each host, which we choose to be the MAC address of the network interface. A host can provide one or more services to other devices on the network. The vulnerabilities in these services can be exploited in order to get more access to that host. Therefore for each host we also determine the set of services *svcs*. The entries in this set consist of a service name and a port number. Next to the services that are running there is also other software operating on the host, the set of other software is defined as sw. The services and software which is running on a host mostly determine which vulnerabilities are present on a host. Therefore, also a set of host-specific vulnerabilities $vuls$ is determined.

The connectivity between host is also important to determine if a certain exploit can be launched from one to another host. For now we assume that firewalls can restrict the network traffic on port level. Therefore we define connectivity as a ternary relation $C \subset H \times H \times P$ where P is an Integer port number. Here $C(h_1, h_2, p)$ means that host h_1 can reach host h_2 on port p.

A host h_1 can *trust* another host h_2. This means that host h_2 has access to host h_1. For instance a device can store the credentials of a Wi-Fi network if it has logged in. From that moment anyone who has access to that device can access the wireless network.

We want to model the state of the intruder in some way. The actions that an intruder can perform also depend on its gained knowledge. Important knowledge for an intruder to start with is which hosts are available on the network and the logical topology (i.e. connectivity). If the intruder gains knowledge about this he or she can identify vulnerable hosts and the entry points of the network. The intruder's knowledge also includes login credentials of users within the network. With this information the intruder can impersonate as a legitimate user in order to get the same

Agent $i \in \mathcal{I}$	S_i	A_i
E	I	A
D	Ids	$alarm$
S	$H \times C \times T$	$\emptyset$

Table 3.1: The agents within the model with their corresponding states and actions.

privilege level as that user. In order to quantify the privilege level of the intruder we also include in the intruder's model the privilege level to each host. We define this privilege level as the function $plvl : H \to \{none, user, root\}$. There is a strict total order on the privilege levels: $none \leq user \leq root$. With the root privilege level we mean that this user has full access to a device, while with user level privilege there are some restrictions. If the privilege level is equal to none there is the same level of privilege as a user which is not authenticated.

Each action that can be executed by an attacker is defined as the triple (r, h_s, h_t). The attribute r is a rule that describes how the intruder can affect the system and which information he obtains. This rule consists of four components:

- *intruder preconditions:* Specifies conditions about the knowledge and privilege levels of the intruder.
- *network preconditions:* Specifies conditions about the target host state, network connectivity, trust, services, and vulnerabilities that must hold before an action can be launched.
- *intruder effects:* An action can give the intruder additional knowledge about the system or can give him additional privilege rights.
- *network effects:* This component describe the effects that the launched action has on the network.

Next to the action attribute r we have attributes h_s and h_t are hosts in H. These attributes describe the host from where the attack is launched and the target of the attack respectively. An action is typically executed by an intruder, but there can also be an Intrusion Detection System (IDS) available within the network which could raise an alarm. Therefore we introduce a special type of alarm action which can be executed by the Defender agent.

Assessment graphs can potentially say something about the security of a system. Although the attack graph model of a home network that we have specified in Section 3.1.1.1 is already quite complex, it is not yet accurate enough to exactly model the exact state of the home network. From this model it is not yet possible to accurately determine if a certain action or exploit can be executed. To start off with the connectivity relation, this relies on many factors such as firewalls, routing settings, and NAT settings. Also there is a difficulty with defining the preconditions of the exploits. In practice there is no such a database available which describes precisely enough when an exploit could be executed.

Additionally there are some scaling problems with the state-space of the model. The state-space does not scale with the number of devices and services. Therefore is decided to use a different approach.

3.1.1.2 Assessment flowcharts

Many configuration options are possible within a home network, some may be less secure than other. It is not always easy to say if some setting in the configuration is secure or not, because it depends on the circumstances where the system is in. For instance, some device in the home network can run a network service that has a known vulnerability which is exploitable. The severity of this vulnerable service depends on whether the service is reachable from someone outside the home network. In this case there should be checked if there exists a port forwarding rule in the

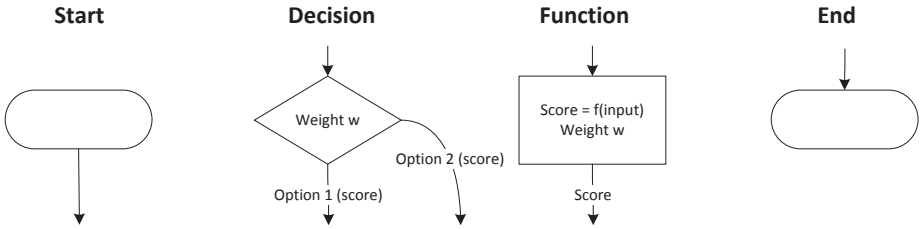


Figure 3.2: The possible nodes in an assessment flowchart.

configuration which makes the vulnerable service directly reachable from the Internet. If such a port forwarding rule exists the overall security of the network gets affected by it.

Taking the circumstances of a system into account makes the assessment process more dynamic. To do this we introduce assessment flowcharts which are used to compose a dynamic assessment. Assessment flowcharts are a model to assess arbitrary configurations of systems. The idea is that a security expert can, based on a security analysis, design an assessment flowchart. This assessment flowchart can then be executed automatically by the system.

3.1.1.2.1 Components

An assessment flowchart consists of multiple nodes that are connected together in the same fashion as a flowchart. The assessment flowchart starts from a single 'Start' node and ends into a single 'End' node. In between the 'Start' and 'End' node there are 'Decision' and 'Function' nodes connected in some way. Figure 3.2 shows how each node looks like.

Start: Each assessment flowchart has one start node to indicate the start point of the assessment. A start node can only have one outgoing arrow.

Decision: The decision node is used to check a setting of a configuration. Based on the outcome, a decision is taken for the direction the assessment flowchart proceeds. To each direction is a score assigned. The score is a number between 0 and 1 that indicates how secure the setting is. A 'Decision' node can have two or more directions in which it can proceed, one for every possible option.

Function: The 'Function' node takes some configuration setting as input and determines based on this input a score. As in the 'Decision' node the score is a number between 0 and 1 that is used to indicate how secure the setting is. A typical example of a 'Function' node is the assessment of a password. The input of a password assessment function node would be the password itself. Based on the properties the given password has, a score is calculated.

End: There is one end node in the assessment flowchart. This node can have multiple incoming arrows but has no outgoing arrow.

To indicate the relative importance between 'Decision' and 'Function' nodes weights are introduced. For each 'Decision' and 'Function' node a weight is assigned, which is expressed as a positive number.

3.1.1.2.2 Final score

The final score of a configuration is calculated by going through the assessment flowchart. For every node on the path a result is calculated by multiplying the weight with its corresponding

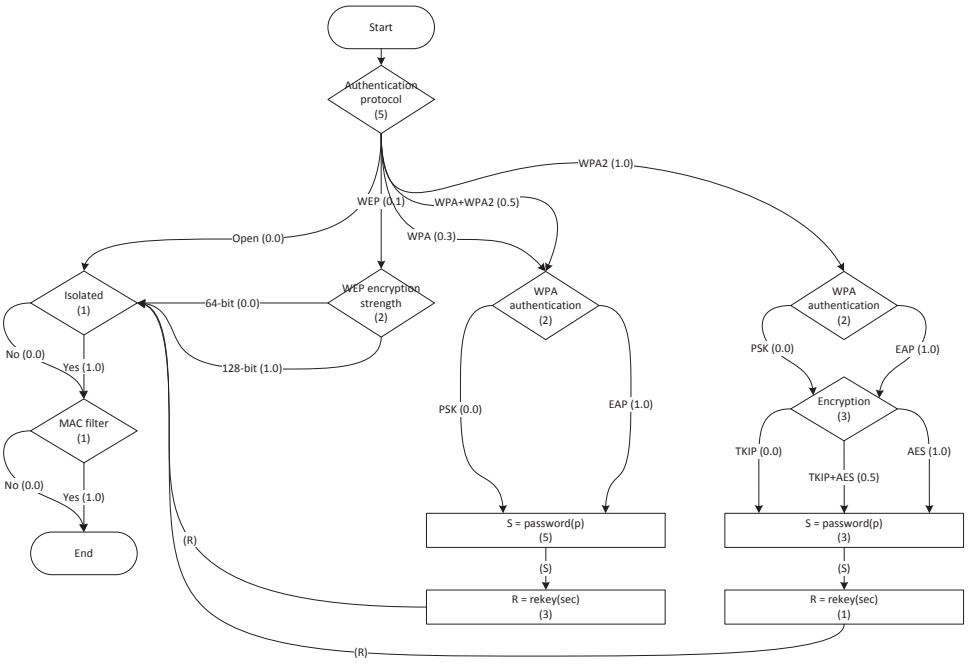


Figure 3.3: Example of an assessment flowchart for the settings of a Wi-Fi access point.

score. Once the program has traversed the entire assessment flowchart, all results are summed up and then normalized into a value between 0 and 1. Normalizing is done by determining the result of the maximum path within the flowchart i.e. the most secure path.

3.1.1.2.3 Example assessment flowchart

As example Figure 3.3 shows the assessment flowchart of a Wi-Fi access point which is implemented in the proof of concept application. The first decision node in the assessment flowchart checks what authentication protocol is used. If the network uses no authentication protocol (i.e. an open network) it will be assigned a score of 0.0. However, if the access point is configured with the WPA2 authentication protocol it will get a score of 1.0 because it is currently the strongest authentication protocol that can be used for Wi-Fi networks. Depending on the authentication protocol that is used the decision node will proceed in another branch of the assessment flowchart. In this way it is possible to assess a specific sub-configuration of a certain type of configuration. For WPA or WPA2 are for instance different settings to be checked than with an open network.

3.1.1.2.4 Security levels

Once the system has gone through the assessment flowchart a final result is computed. A question that remains is: when is a final result considered secure enough? For this we introduce security levels that provide thresholds for the final results of assessment flowcharts. The thresholds for the security levels can be determined by a security expert. This security expert may for example define three security levels as shown in Table 3.2. In this example we get three intervals $[0, 0.5]$, $(0.5, 0.8]$ and $(0.8, 1.0]$ which are mapped to the security levels low, medium, high respectively.

An assessment consists of multiple instances of different types of assessment flowcharts. For example, there can be created multiple instances of the assessment flowchart that we saw before

Security level	Threshold
low	0.5
medium	0.8
high	1.0

Table 3.2: An example of security levels with their thresholds.

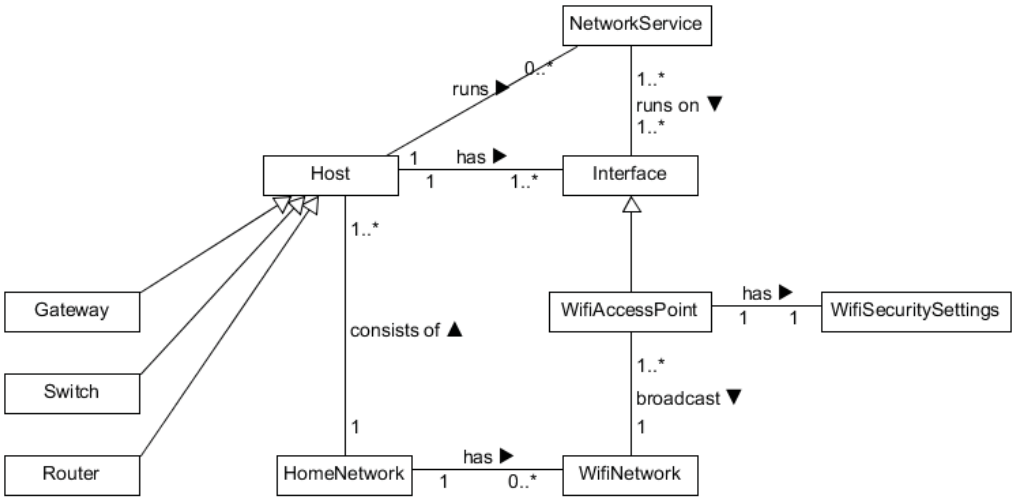


Figure 3.4: Domain model of a home network

in Figure 3.3, one instance for each wireless access point that is present in the home network. Besides the wireless access point configuration in the network, there can be different aspects of the network be assessed. The result of the program once it has run through all instances of assessment flowcharts is a report with a list of final scores for each assessment flowchart. The overall security level of the report will be the minimal security level of all the assessments in the report.

3.1.1.2.5 Domain model

The domain model shown in Figure 3.4 is designed to store the information that is collected from the devices in the home network. Ultimately this domain model could be transformed into an Entity Relation Model (ERD) such that the data can be stored in a relational database.

To explain the domain model we start at the *HomeNetwork* entity that contains one or more *Host* and *WifiNetwork* entities. Each *Host* entity either can be a *Gateway, Switch, Router* or a normal host. For every type of device there can be device specific information stored. A *Host* may run a *NetworkService* such as a HTTP or SSH service. Each *NetworkService* can be provided on one or more *Interface* entities. For instance the SSH service for the gateway router might run both on the internal and the external interface, but the HTTP service that provides the router configuration page is only provided via the internal interface. The Interface entity can be a special wireless interface that enables other devices to connect to it. Such interfaces are WifiAccessPoint entities. WifiAccessPoints broadcast a WifiNetwork using certain Wi-Fi security settings (i.e. entity WifiSecuritySettings).

3.1.1.3 Assessing passwords

Passwords are still a very common method for authentication. But is this kind of authentication secure enough for the application in home networks? One could argue that passwords are a secure enough method to do authentication in case there is used a strong password. A disadvantage of passwords is that passwords are easily copied, and there is no method to determine how many times this already happened.

Passwords might not be a perfect authentication method. However, it is often the only authentication method that is available on a device. Let for now accept that passwords are used as authentication method. To make the authentication as secure as possible we want to assess the strength of passwords. This brings us to the next question: *what is a strong password*. Steven Furnell describes some common used best practices for passwords [Fur07]:

- Use a long password, e.g. 8 or more characters.
- The password should contain both upper case and lower case letters.
- The password should contain numbers.
- The password should contain special characters.
- It should not be possible that a password can be found in a dictionary.
- Do not choose passwords that someone who knows you is able to guess.
- Do not use choose the same password for all systems you use.
- Change your password on a regular basis.

If a *brute force attack* is executed an exhaustive search is done over all the possible passwords. When the to be recovered password is 10 characters long and consists of lower case symbols, upper case symbols and numerical symbols, there are $(26 + 26 + 10)^10 \approx 8.39 \cdot 10^{17}$ different possible passwords. Suppose there would exist a 3 GHz processor which is able to check a password every clock cycle. Note that this is already much faster than a consumer based computer can handle. It would still take about 9 years in order to check all possibilities and 4.5 years on average. This small calculation would pose that passwords are a strong authentication method.

Unfortunately it is not as nice as it seems to be. People are not very good in remembering difficult passwords and often come up with a password in a common format and, at least, pronounceable. Matteo Dell'Amico et al. show in their paper [DMR10] that, within the password datasets they use, the average password length is around 8 characters. In two of the three datasets around 51% of the passwords use only lower-case letters. The policy of the other dataset enforced the usage of numbers. As mentioned before, people tend to use a common patterns for their passwords. Almost 20% of the passwords in the dataset with the 'enforced-number' policy consisted of lower-case letters followed by a single '1'.

Ideally a secure password should apply to all above best practices and is also good memorable. Choosing such a password is not an easy task and could get in the way of the user doing his work. Dell'Amico mentions this as a reason why some users choose to use a weak password [DMR10]. There is definitely a trade-off between usability and security when choosing a good password.

To retrieve a password it is possible to find it by guessing. There are several approaches in order to do this, every time the challenge is to minimize the amount of guesses needed. Checking if a guessed password is correct takes resources like power and time. A rational attacker would only choose to guess a password if the amount of expected guesses g, times the cost c of checking if a password is correct is less than the profit p of a successful password recovery, i.e. $g \cdot c < p$.

To recover a password a strategy could be to perform a *dictionary attack*. The dictionary attack will try every word in a dictionary file that is provided until the correct password is found. The success probability of the dictionary attack heavily relies on the quality of the dictionary and the type of password that is used. When a random password is used, a dictionary will often be unsuccessful (or the random password consists of a dictionary word by accident).

3.1.1.3.1 Assessment

How to measure the strength of a password? With the strength of a password we mean how hard it is to guess the password. This could be expressed as a measurement of the expected amount of effort an attacker has to perform. For instance the expected time an attack would take. A difficulty with expressing the time that an attack on the password would take is that the processing speed of the attacker is unknown.

An indication of the strength of a password can also be calculated by estimating its entropy. The entropy is a statistical parameter which measures in a certain sense, how much information is produced on the average for each letter of a text in the language [Gui06].

Let $\mathcal{X}$ be the set of possible symbols (i.e. passwords) in the language, let $X \in \mathcal{X}$, $X \sim \mathbb{P}$ and $p_x = \mathbb{P}(x)$. Then equation 3.1 gives a definition of the Shannon entropy.

$$H(X) = \sum_{x \in \mathcal{X}} p_x \log_2 \frac{1}{p_x}. \tag{3.1}$$

The Shannon entropy can be seen as a lower bound on the average number of binary (yes/no) questions that you need to ask about the RV in order to learn the outcome x [Š14]. The question is now what the minimum entropy should be of a password. NIST states that for passwords that are used between 2011 and 2030 should have minimal entropy of 112 bits[2].

Now we apply the equation 3.1 for Shannon entropy to calculate the entropy of a certain password. The distribution of X has to be known in advance. If a password is chosen uniformly at random, then $\forall x, y \in \mathcal{X} : p_x = p_y$, which means that every character has an equal probability to occur. We then can derive the following formula:

$$\begin{aligned} H(X) &= \sum_{x \in \mathcal{X}} p_x \log_2 \frac{1}{p_x} && (3.2)\\ &= \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log_2 \frac{1}{\frac{1}{|\mathcal{X}|}} && (3.3)\\ &= \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log_2 |\mathcal{X}| && (3.4)\\ &= |\mathcal{X}| \frac{1}{|\mathcal{X}|} \log_2 |\mathcal{X}| && (3.5)\\ &= \log_2 |\mathcal{X}| && (3.6) \end{aligned}$$

In this way we only would have to solve $\log_2 |\mathcal{X}| = 112 \rightarrow |\mathcal{X}| = 2^{112}$. We would then have to be able to find a set of symbols and a sequence length which is able to represent 2^{112} different passwords. Suppose we pick all readable characters of the ASCII character set we would have (126-31) = 95 different symbols. With a sequence of 18 characters chosen uniformly at random out of the set of readable ASCII characters we would be able to meet the recommendation of NIST (e.g. $H(X) = \log_2(95^{18}) \approx 118.26$). A password of 18 characters is from a usability viewpoint already very long and differs much from the average password length of 8 characters.

However this calculation assumes that the password is chosen uniformly at random. In practice it turns out that people do not use these random passwords they either use passwords that:

- are pronounceable
- are easy to remember
- often consist of words
- consist of common patterns (e.g. a word followed by a number)

[2]Key length - Cryptographic Key Length Recommendation: `http://www.keylength.com`

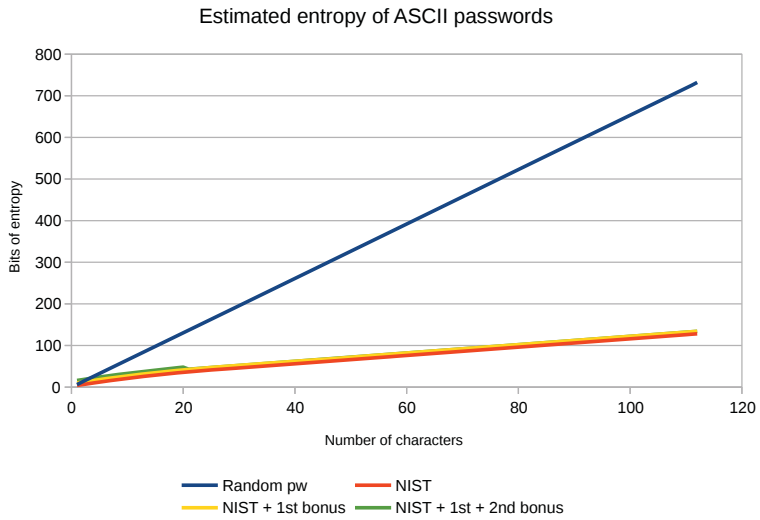


Figure 3.5: Estimated entropy of a password according its length.

This has as consequence that the assumption that the characters are chosen uniformly at random does not hold. In 1951 Claude Shannon already published a paper which describes how to predict the entropy of printed English [Sha51]. Based on this NIST made a recommendation to guess the password strength of a user chosen password [Gui06]:

- The entropy of the first character is taken to be 4 bits.
- The entropy of the next 7 characters is 2 bits per character.
- For the 9th through the 20th character the entropy is taken to be 1.5 bits per character.
- For characters 21 and above the entropy is taken to be 1 bit per character.
- A "bonus" of 6 bits of entropy is assigned for a composition rule that requires both upper case and non-alphabetic characters.
- A bonus of up to 6 bits of entropy is added for an extensive dictionary check. The assumption is that most of the guessing entropy benefits for a dictionary test accrue to relatively short passwords, because any long password that can be remembered must necessarily be a "pass-phrase" composed of dictionary words, so the bonus declines to zero at 20 characters.

Figure 3.5 shows a graph of the estimated entropy of both the Shannon entropy (shown as: Random pw) and the NIST entropy of passwords. The NIST entropy is shown in three variants. The red line (shown as: NIST) shows the estimation without any of the bonuses applied. The yellow line (shown as: NIST + 1st bonus) shows the entropy of the NIST entropy when the first bonus is applied. Finally the green line (shown as: NIST + 1st + 2nd bonus) shows the NIST estimation with both the 1st bonus and 2nd bonus. In this case we can see that after 20 characters the bonus of 6-bits of entropy reduces to zero. For all entropy estimations we can observe that the entropy increases when the number of characters increases. The password strength estimation by the Shannon entropy is more optimistic than the NIST estimation. However, the Shannon entropy assumes that the password is chosen at random out of 95 symbols.

According to the NIST recommendations a secure password should have 112 bits of entropy. When a random password is chosen a password of 19 characters would be sufficient. In the case of a non-random chosen password the NIST estimation should be used, which comes down to a password consisting of 90 characters if the first bonus is applied. A password of this length would be too long to be convenient.

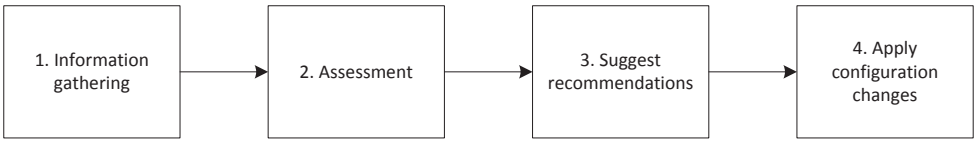


Figure 3.6: The stages of the intrusion prevention solution

3.1.2 Implementation

To find out if the proposed solution would work there is created a proof-of-concept application. The application runs on an Android smart-phone and has as main functionality to improve the configuration of a home network. The application requires the smart-phone to be connected to the home network.

The application is designed to do its job in four stages. The first stage is about *information gathering* and collects information about the configuration of the network. The information is gathered by observing the network where the host is connected to, but also by logging into network infrastructure devices such as: routers, switches, and access points.

Once the system finished gathering information it will proceed to the *assessment* stage. In this stage several components get assessed which in the end will result in an assessment report. After the assessment report is generated the application should give suggestions to improve the security of the network in the *suggest recommendations* stage. Unfortunately this stage is not yet implemented in the current implementation of the application. After the suggest recommendations stage the application should enter the *apply configuration changes* stage. In some cases the application will able to apply some of the recommendations to improve the home network configuration itself. Figure 3.6 shows a summary of the phases that the application runs through.

In the following sections we will discuss the stages in more detail in a chronological order.

3.1.2.1 Stage 1: Information gathering

The information gathering stage collects data about the home network configuration from different sources. Because the device that runs the application is connected to the home network, already some information can be extracted by the device itself. For instance if the device is connected via Wi-Fi, it is able to determine what security protocol is used for the wireless network. Another way to retrieve information about the home network configuration is by logging into network infrastructure devices. This is a very accurate method to retrieve the configuration of a certain device. Besides that, to apply the configuration in a later stage it is also required to login into the device. A disadvantage of this approach is that different devices will use different protocols manage the device. Some might only support configuration via a web-page while others also support configuration via command-line over SSH or telnet.

There is much disparity between network infrastructure devices and there is no standardized configuration interface available. This causes that custom support has to be built for every different device. There may be many different devices supported, while only a small subset of these supported devices will be present in the home network. To reduce the size of the application a plug-in mechanism could be used in combination with a central repository that contains plug-ins for support for a certain device. When the application detects a network infrastructure device and it does not have support for that device available, it can query the repository to request for a plug-in that provides the support. The process of retrieving this support is shown in Figure 3.7.

The application gathers the available information from every compatible network infrastructure device in the network and puts it into a data structure conform the domain model shown in Section 3.1.1.2.5. Once the gathering of the data is finished and everything is stored into the domain model

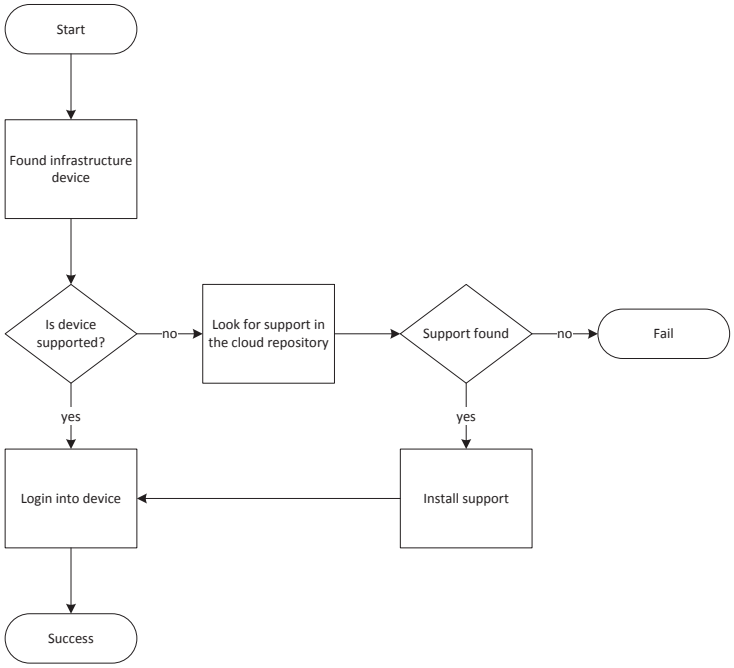


Figure 3.7: Flowchart of the process of determining if support is available.

the application proceeds to the second stage: assessment.

3.1.2.2 Stage 2: Assessment

After the required data about the home network configuration is gathered the application proceeds into the assessment stage. In this stage several aspects of the configuration are assessed by the system. To give a judgment about the security of these aspects we use the flowcharts that are explained in Section 3.1.1.2. For the assessment of Wi-Fi access point configurations the example assessment flowchart given in Section 3.1.1.2.3 is used.

3.1.2.3 Stage 3: Generate recommendations

From the previous stage we get a report which gives an overview of the assessments that are done. The overall security level is determined by the minimum security level of the assessments. If this overall security level is not equal to the highest possible security level then there is room for improvement. Now remains to determine what has to be improved in order to increase the overall security level of the system. For this the program only considers the problematic assessments which have a security level that is equal to the overall security level.

3.1.2.4 Stage 4: Apply configuration stages

For some recommendations that are done by the previous stages, it may be possible to apply them automatically by the program. To change the configuration of home network devices it is required for the application to log into the device.

3.2 Intrusion detection

To examine if intrusion detection can be done within home networks we defined two events of malicious behavior to detect:

1. a host performs a port scan in the network;

2. a wireless host injects a forged de-auth package into the wireless network.

The reason to select port scanning as malicious event to detect is because its a common reconnaissance action to get an impression about the hosts present in the network including the services they run. Port scans can be done in all kind of ways but are certainly not something a normal user might do.

The second malicious event is the de-auth package injection attack on a Wi-Fi network. De-auth packets are used by the access point to disconnect all its connected devices. Normally the access point only sends these messages in case of an expected reboot. An attacker is able to forge and inject de-auth packages using freely available software such as Aircrack-ng.[3] The de-auth packages can be used to perform a DoS attack by continuously sending these packages. Another possibility is to enforce devices to re-connect to the access point. The connection between a Wi-Fi device and WPA or WPA2 Wi-Fi access point is established using a WPA handshake. Once the attacker is able to capture one or more of these WPA handshakes he is able to perform a brute force attack to retrieve the password. To detect this attack a wireless intrusion detection system (WIDS) is required, which belong to a different class of devices.

In the following sections we discuss how the test setup with the IDS and WIDS. The test setup is built to resemble the situation of a real home network. Section 3.2.1 explains of what components the setup consists and why it is designed in this way. How the IDS and WIDS are set up and how we demonstrate that they are able to detect the attacks is shown in Section 3.2.2 and 3.2.3 respectively. Finally, we discuss the results of the experiments in 3.2.4.

3.2.1 Home network test setup

To resemble the situation of a real home network there is built a test setup. The setup consists of two networks: the home network, and the Internet. To the home network part are four laptops and a smart-phone connected via an Ethernet or a Wi-Fi connection. The infrastructure that provides these connections consist of a gateway router and a switch.

The Internet part of the setup consists of a switch that connects two ‘external’ hosts to the home network. One of these hosts acts as the ISP of the home network by ‘sharing’ its Wi-Fi connection to the Internet over its Ethernet connection. The other host connected to the Internet part of the network is used to perform the external attacks on the Home network. The complete network topology of the test setup is shown in Figure 3.8.

3.2.1.1 Gateway router

The gateway router is assembled during the project; the parts were selected to mimic the processing power of a high-end gateway router that is on the market within five years. The device has two Ethernet interfaces to connect to the home network part of the setup and to the Internet part of the network. Besides that the gateway router also has two Wi-Fi interfaces, one to set up as access point, and the other to set up as monitoring interface. Table 3.3 gives an overview of the selected components of the gateway router.

The gateway router’s operating system is a minimal installation of the Linux distribution Debian, version 8.1. Debian is not a router distribution, so to make the system function first some modifications had to be done. The basic steps that were taken are:

[3] Webpage:http://aircrack-ng.org

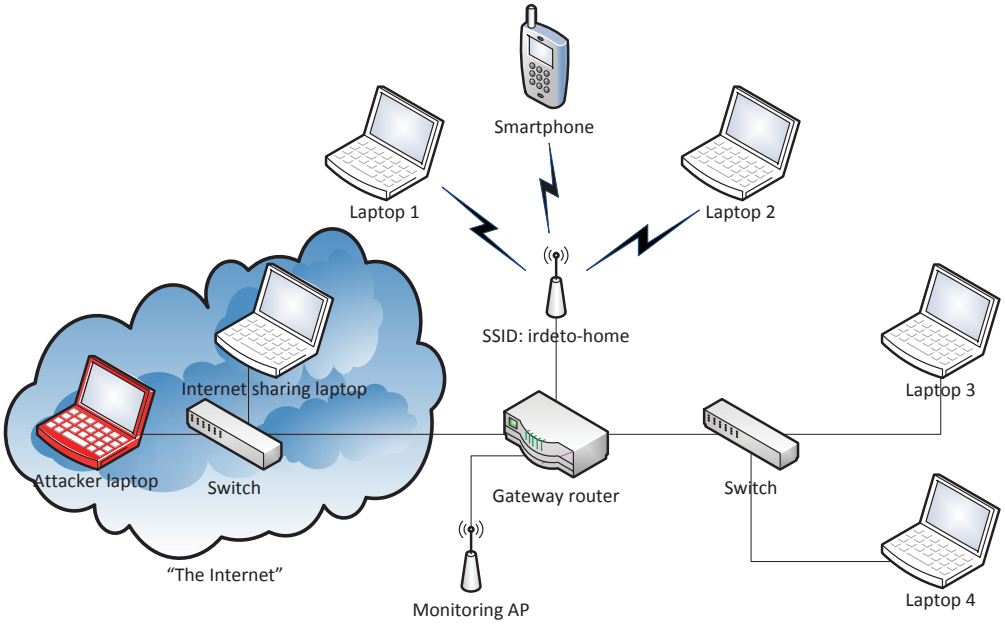


Figure 3.8: Network topology of the test setup.

Component	Description
CPU	Intel Celeron J1900 (2.0 GHz) quad-core processor
Motherboard	Gigabyte J1900N-D3V, Micro ITX motherboard, 2 Ethernet sockets
Memory	Crucial 4GB DDR3-1600
Storage	SSD, Trancend SSD370, 32GB
Wireless adapter 1	Intel Dual Band Wireless-AC 7260
Wireless adapter 2	Alfa AWUS036NEH 802.11n/b/g Long-Rang USB Dongle

Table 3.3: Components of the gateway router.

- Enable forwarding of traffic in the Linux kernel. By default the Linux kernel drops all traffic that is not destined to the machine itself.
- Setup the IPtables firewall such that all traffic from the Internet part of the setup is blocked, unless the connection is instantiated from an internal host.
- Configure the wireless interface as an WPA2 access point.
- Install and configure a DHCP server such that the hosts connected to the home network part of the setup.

Once this was done the system functions as a gateway router.

3.2.1.2 Overview deployment

The test setup already mimics a possible real home network. It has a gateway router that connects the internal home network to an external network (i.e. the Internet). And besides that, the network has both a wired and a wireless part that are unified as a single network. Any host, no matter if it is connected via Ethernet or Wi-Fi, can connect to any other host within the home network.

Now the network is ready for the IDS and the WIDS to be installed on the gateway router.

3.2.2 Setting up the IDS

The installation of an IDS requires an analysis of the type of traffic that is expected and where to place such an IDS. IDSs can be configured specifically for a certain

In chapter 2, Section 2.4.3 we already discussed the placement of an IDS.

3.2.2.1 Snort IDS

Snort is an open source network-based intrusion detection system (NIDS). It is able to perform real-time traffic analysis and packet logging on IP networks. Snort uses a dataset with rules that specify patterns of potential malevolent network traffic. There exist tools to update this rules dataset such that Snort is able to detect the most recent attacks.

For the test setup we have used the community rules dataset that is provided by the organization of Snort. This organization also provides a more elaborate rules dataset for paid subscribers. However, for our purposes the community rules dataset is sufficient.

3.2.2.2 IDS placement

The Snort IDS is able to run multiple instances distributed over multiple hosts on a network. Each instance monitors traffic passing by on a certain interface (e.g. Wi-Fi or Ethernet interface). In the test setup all Snort instances run on the gateway router. This is a design decision which already creates a restriction on the detection capabilities of the IDS. For instance, the IDS is not able to monitor the traffic between laptop 3 and laptop 4 (see Figure 3.8). This is because these two hosts are connected via a switch. If the one host sends traffic to the other host, it will first reach the switch, and then the switch will forward the traffic directly to the other host. So in this case the traffic does not reach the IDS, and therefore the IDS is not capable of monitoring the traffic.

However, to remain undetected for the IDS an intruder needs to know the topology of the network in order to evade IDS. Usually the network topology and the location where the IDS is deployed, is unknown to the intruder. Deploying the IDS only at the gateway router may therefore be sufficient to do intrusion detection within home networks.

As is shown in Figure 3.9, the gateway router runs multiple instances of the Snort IDS, one on each interface where traffic flows through:

- *IDS-1:* Monitors the traffic that passes through the `eth1` interface. This interface is connected to the Internet part of the network.

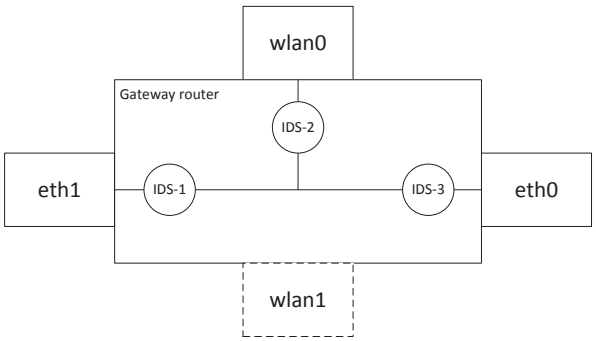


Figure 3.9: Three Snort instances on the gateway router.

- *IDS-2:* Monitors the traffic that is sent over the wireless network, but only on network layer level. Internally this interface is the `wlan0` interface. The Wi-Fi interface that broadcasts a WPA2 network.
- *IDS-3:* Monitors the traffic that passes through the Ethernet interface `eth0`. This interface is connected to the home network part of the setup.

The lines between the interfaces indicate between which interfaces there is traffic possible. Interface `wlan1` is depicted as a dotted line box, this is because it only performs monitoring and is not connected to any network.

By setting up the instances of Snort in this way two of the three entry points that we defined in Chapter 2, Section 2.3 are monitored. Entry point 1 - the gateway, is completely monitored since all interfaces where traffic flows through are monitored. This includes the traffic that is destined to the gateway router itself. Entry point 3 - the Wi-Fi network, is partially monitored. The network traffic is be monitored on the network layer level. All attacks that are done on the lower layers of Wi-Fi (e.g. physical layer attacks) will remain undetected to the IDS. This is the reason why besides an IDS also an WIDS is required. Only entry point 2 - via an intermediary device is not monitored entirely. Especially in the case of an attack through a secondary channel (i.e. an channel different than the home network infrastructure).

If one of the instances would be removed the detection capabilities of the IDS would decrease. If IDS-1 or IDS-3 would be removed, malicious events that are destined directly to the gateway router would remain undetected. Removing IDS-2 would cause that traffic between two hosts connected via the Wi-Fi network remains undetected.

3.2.2.3 Program stack

Running only the Snort IDS instances is not sufficient to do reliable intrusion detection. To process, store and visualize the alarms there are additional tools required. In this section we will discuss the functionality of all the tools that are used to set up Snort properly.

3.2.2.3.1 Barnyard2

Barnyard2 is used to make Snort function more reliable. Barnyard2 is a so called spooler: an application that watches some log file and processes it such that can be stored in a database. Snort itself is also capable of writing its detections directly to a database. The problem is however, that if there is much traffic, the Snort instance will be very busy processing the traffic. Once Snort's buffers overflow, it will start to miss traffic. This reduces the detection ratio of the IDS. Therefore we use Barnyard2, which is started as a separate process. Snort outputs its detections into some log file, and Barnyard2 processes these logs and outputs them to a MySQL database.

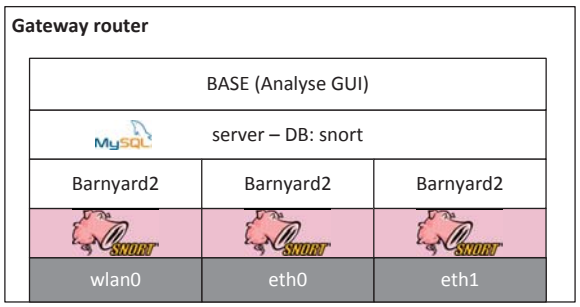


Figure 3.10: Used program stack for the Snort IDS.

3.2.2.3.2 MySQL

To store the detections of Snort a MySQL database is used. The data-structure of this database supports multiple instances of Snort.

3.2.2.3.3 BASE

Basic Analysis and Security Engine (BASE) is a web front-end to query and analyses alerts coming from the Snort IDS. This tool is used to see if the Snort IDS is capable of detecting an attack.

3.2.2.3.4 Program stack overview

When Snort, Barnyard2, MySQL and BASE are put together we get a program stack that is shown in Figure 3.10. For every interface that routes traffic there is an instance of Snort and Barnyard2. The instances of Barnyard2 all output the detected events into the MySQL database. To visualize the detected events the BASE analysis web-interface is used.

This setup is well suited to be distributed over multiple hosts within the network. Another host in the network can then be set up to run one or more instances of Snort and Barnyard2. Barnyard2 has then to be configured such that it writes its detection events to a central MySQL database, instead to local MySQL database. Another thing that then has to be setup is secure communication to the central database. Otherwise the database connection can be eavesdropped. Setting up a secure database connection can be done by a VPN or a SSH connection.

3.2.2.4 Demonstration

To show that the Snort IDS setup is capable of detecting the port scan, there is performed a port scan from multiple hosts to multiple destination hosts in the test setup. The default SYN/ACK port scan of the `Nmap` tool is used. This scan method determines if a certain port on some host is open by initiating a TCP connection by sending an so called SYN packet to that host on that port. Then the port scan waits for a response. If the host responds, its response will either be ACK or RST. An ACK packet indicates that the port on that host is listening for incoming connections. When an RST packet is received it means that the port is not listening.

Table 3.4 gives an overview of the experiments that have been done. The host names in the table correspond to the host names used in Figure 3.8. In the table a ✓ symbol represents a successful detection of the attack by the IDS, the ✗ symbol stands for no detection, and N/A indicates that the destination host is not reachable or is the host itself.

The detection results in Table 3.4 show that in most cases the port scan can be detected. Only when Laptop 3 performs a port scan to Laptop 4 it remains undetected. This was expected

		Target				
		Gateway router	Laptop 1	Laptop 2	Laptop 3	Laptop 4
Source	Attacker laptop	✓	N/A	N/A	N/A	N/A
	Laptop 1	✓	N/A	✓	✓	✓
	Laptop 3	✓	✓	✓	N/A	✗

Table 3.4: Detection of port-scan from launched from a source host to a target host.

because, as explained earlier in Section 3.2.2.2, the traffic between Laptop 3 and Laptop 4 never reaches the IDS.

3.2.3 Setting up the WIDS

To detect Wi-Fi specific attacks a network layer IDS such as Snort is not sufficient. Monitoring on a network traffic layer lower than the network layer is required in order to be able to detect some Wi-Fi attacks.

3.2.3.1 Kismet

To perform the wireless intrusion detection the tool Kismet is used. Kismet is a Wi-Fi layer 2 intrusion detection system, and besides that, it is also a wireless network detector and sniffer. For the purpose of this experiment only the WIDS functionality of Kismet is used.

3.2.3.2 WIDS placement

The placement requirements of a WIDS are different from an IDS. WIDS scan for any Wi-Fi communication on the 2.4GHz or 5GHz band. These systems monitor any Wi-Fi traffic that it receives. So, for WIDS, the physical location becomes more important than the logical location within the network. With logical location within the network we mean, to which device the WIDS is connected on the network topology map.

Monitoring the entire range of the Wi-Fi access points in the network is not sufficient to detect attacks on connected Wi-Fi devices. Suppose we have the situation that is depicted in Figure 3.11. Here the Home network AP also functions as a WIDS with a range that is indicated with by the dotted circle. A laptop is connected to the home network AP. Then there is some rogue AP that can reach the laptop, but not the home network AP where the WIDS runs. In this situation the rogue AP is able to do an attack on the laptop, without the home network AP being able to detect it.

3.2.3.3 Demonstration

To demonstrate the de-auth attack being detected, the attack is executed by a laptop with the `Aircrack-ng.` toolbox. The `Aircrack-ng.` toolbox is an open-source toolbox with tools to crack WEP and WPA keys but also to perform the de-auth attack.

Before the attack is executed the test setup is prepared such that Kismet is configured such that it functions as a WIDS. Besides that a Wi-Fi device is connected to the test setup access point, this as a control mechanism to check if wireless devices actually disconnect when the attack is executed.

To perform the attack, the following commands are executed as root:

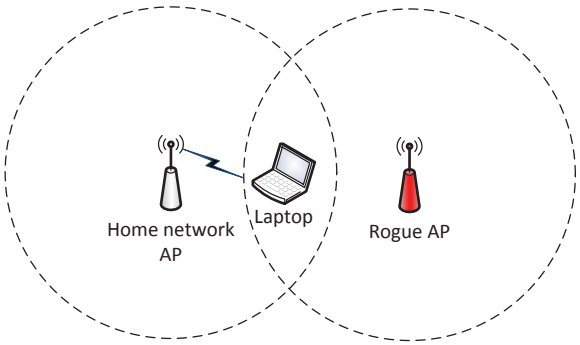


Figure 3.11: Detection restriction for a WIDS.

1. Create a monitor interface:
   `airmon-ng start wlan0`
2. Scan for broadcasting Wi-Fi APs:
   `airodump-ng mon0`
3. Pick the MAC address of the AP that you want to impersonate, and execute the attack (replace `<MAC>` with the MAC address of the AP):
   `aireplay-ng --deauth 0 -a <MAC> mon0`

3.2.4 Results

The combination of the IDS and the WIDS show to be effective in detecting a port scan and a de-auth attack. However, the performance depends on where and how the IDS and WIDS are deployed. The placement of the IDS on the router gateway seems to be adequate to do detection in a home network. Because of the presence of switches in the home network infrastructure, the IDS is unable to monitor all traffic that flows through the network. This reduces the detection capabilities of the IDS. However, it still is an improvement on the security of a home network since, currently, no detection of malicious traffic is done at all.

The deployment of a WIDS seems to be more troublesome than for an IDS. This is because it is difficult to cover the entire receiving area of the Wi-Fi devices that are connected to the home network AP. Deploying the WIDS solely on the gateway router will cause that some attacks will remain undetected. However, the same reason as for the IDS applies, deploying an WIDS is still an improvement on the security of a home network because, usually there is no detection done at all.

Both the IDS and the WIDS are capable of detecting many more types of attacks than the attacks that are tested during this project. For the Snort IDS there is a tool `Pulled pork` that updates the Snort rules dataset. In this way the IDS is capable of detecting recent attacks. The Kismet WIDS does not have such an update mechanism. The necessity is less, because fewer attacks on Wi-Fi protocols are developed than on other network protocols.

Chapter 4

Conclusion

During this project we experienced that home networks can be poorly secured, which causes that home network devices can be at risk, including the data they contain. One of the problems is the unawareness of the average consumer about the threats that exist to their home network devices. The user has insufficient knowledge about computer networks and computer security to make grounded decisions when they setup or change the configuration of their home network. It is difficult to see the exact consequences of a configuration change on the network. Often the available home network equipment already supports sufficient mechanisms to provide security to a decent level. However, these capabilities are not always utilized because of a poor configuration. Something that makes the user lose the overview is that the home network configuration consists of many settings, which can be distributed over multiple devices in the network.

4.1 Home network security

Each device that is connected to a home network can potentially contain private data such as: documents, photos, videos etc. Some of these devices also have additional sensors, such as microphones or cameras. If an intruder would retrieve access to such device he/she is potentially able to confiscate the privacy of the user in real-time (e.g. when the intruder is able to view the webcam images).

It is also important that network infrastructure devices such as routers, switches and Wi-Fi access points are protected. Once an intruder gains access to these devices he is potentially able to setup a man-in-the-middle situation where he/she can affect the confidentiality and integrity of secure connections.

We defined two profiles of hackers that could have a motivation for intruding a home network: the curious neighbor and a member of a criminal organization The entry points that these hackers would use to get access to the home network will be the following:

1. Gateway router
2. Intermediary device
3. Wi-Fi network

To secure for attacks to these entry points it is important that the home network is properly configured. This belongs to the approach of intrusion prevention. Assuming that the home network is properly configured, malicious events are still possible. To do intrusion detection by monitoring network traffic in real-time, there exist Intrusion Detection Systems (IDS).

4.2 Intrusion prevention

To improve on intrusion prevention in home networks there is aimed for the use of existing techniques that are already available by the home network infrastructure. This comes down to checking a home network configuration, posing recommendations and finally instrument network infrastructure devices to make improvements on the home network.

An Android proof-of-concept application is made that is capable of retrieving the configuration of a DD-WRT router. Based on the retrieved configuration the Android application performs an assessment. To create these assessments we introduce *assessment flowcharts*, which is a model to assess an arbitrary configuration. The idea behind it is that a security expert creates such an assessment flowchart and that the system is able to execute it. The implementation of generating recommendations and instrumenting infrastructure devices still needs to be done.

The Android application shows that the taken approach is functional. However, retrieving the configuration and instrumenting the infrastructure devices is device dependent.

4.3 Intrusion detection

For the intrusion detection part of the project, we investigated the possibility of using IDS in home networks. To demonstrate the ability to detect attacks there is built a test setup of a home network. All the equipment used in the test setup consisted of consumer level devices, except the gateway router. Current consumer level gateway routers do not have enough computing power to run additional resource intensive tasks such as running a IDS. Therefore, a gateway router is built that should have the processing power of a high-end gateway router that is on the market within five years.

We limited our scope of intrusion detection to two attack scenarios:

1. detect a port scan on a host;
2. detect a de-auth attack on the Wi-Fi network.

These scenarios were used to investigate and demonstrate the detecting capabilities of these IDS. To detect port scans we deployed Snort IDS including on the gateway router. For the detection ratio of an IDS its deployment location is an important factor. The IDS must be able to monitor sufficient network traffic in order to perform well. Experiments were done to see from which source host, to which destination host, the IDS is able to detect a port scan. In almost every experiment the IDS was able to detect the port scan.

The only experiment where the IDS was unable to detect the port scan was when the source and destination host were connected to the same switch. This was expected because the IDS is not on the route in between the two hosts. Therefore the traffic of the port scan never reaches the IDS.

Detecting the de-auth attack is done by deploying the Kismet wireless intrusion detection system (WIDS) on the gateway router. Kismet succeeded in detecting the de-auth attack. However, it still remains the question: how well this detection performs. An attack could possibly be executed such that the de-auth attack reaches a host but not the WIDS. This would mean that the de-auth attack remains undetected.

Our experiments on the test setup show that intrusion detection is possible within home networks. However, the deployment process requires expert knowledge about where to place the IDS instances and how to configure them. Also the detection alarms of an IDS have to be made more meaningful to the user. Preferably the system should take actions by itself in case of a detection of a malicious event, and notify the user about this action.

Appendix A

Home networks

Within homes there are multiple endpoint devices such as laptops, tablets, IP cameras etc. that rely on an Internet connection. The connection from these endpoint devices to the Internet has to be provided by a network infrastructure. Usually houses contain a private home network infrastructure that provides the Internet connection to the endpoint devices.

It is possible to graph the network by drawing all devices as vertices and indicate the connections between the devices with an edge in between the devices. This kind of map of the network is called a network topology. Within network topologies the type of device can be indicated with icons of that particular device. Connections between devices can be wired, but some are also wireless. To indicate that two devices are wireless connected we use lightning-shaped edges.

Figure A.1 shows an example of a home network topology. In the Figure there are shown different types of endpoint devices such as laptops, smart phones, a printer, a tablet, a smart TV, a baby camera etc. Some of these devices may require access to the Internet or at least access to other endpoint devices within the home.

Section A.3 gives summary of the devices that are common within a home network infrastructure.

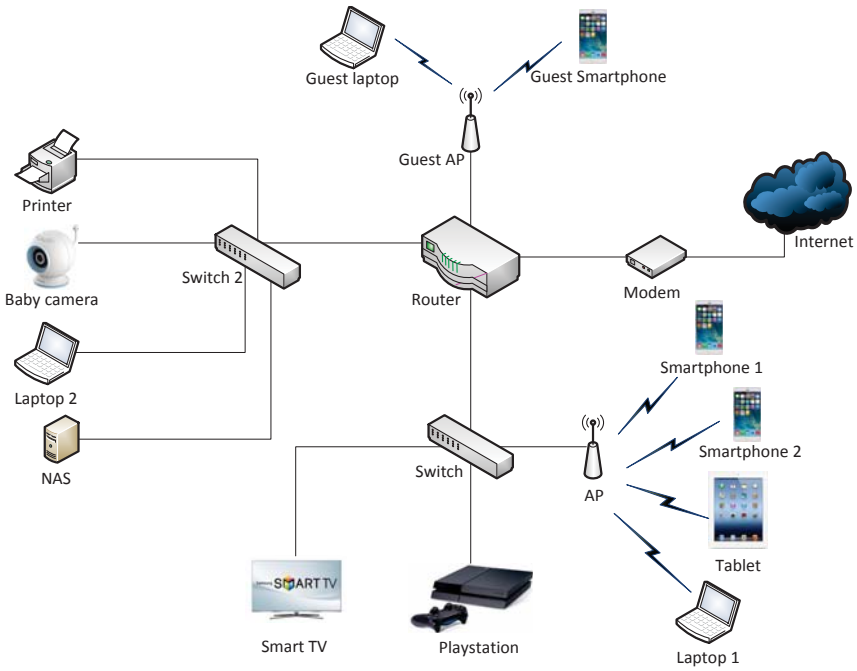


Figure A.1: Example of a home network architecture

A.1 Network scales

Networks can be deployed within several scales. A network can be deployed within a square meter but can also have a worldwide reach like the Internet. The smallest scale networks are referred to as Personal Area Networks (PAN). These networks have a reach for a limited amount of meters. An example would be the connection between a computer and a mouse. A scale bigger would be a Local Area Network (LAN). This scale of network ranges between 10 and 1000 meters. An example of an LAN is the IEEE 802.3 Ethernet or IEEE 802.11 Wi-Fi network which we use to connect computers to. There also exist bigger networks which can span an entire city, these networks are the Metropolitan Area Networks. A cable television network is a good example of a Metropolitan Area Network. These networks span often area of the order of magnitude of 10 km. The last class of networks that spans entire countries, continents or even the entire world is referred to as Wide Area Networks. The Internet is a good example of such a WAN. These classifications of size are summarized in Table A.1.

Inter-device distance	Devices located in the same	Example
1m	square meter	Personal area network (PAN)
10m 100m 1km	room building campus	Local area network (LAN)
10km	city	Metropolitan area network
100km 1000km 10.000km	country continent planet	Wide area network (WAN)

Table A.1: Classification of interconnected devices by scale [TW10].

A.2 Network Address Translation (NAT)

Currently most ISPs still provide IPv4 addresses to their customers. Since there are not enough IPv4 addresses to provide every device within the world a unique address, ISPs usually assign one IP address per customer. This phenomenon of insufficient amount of IPv4 addresses is also referred to as *IPv4 address exhaustion*. To be able to share one IP address with multiple devices the router deploys Network Address Translation (NAT). NAT is used to translate IP addresses of the one address realm into IP addresses of another address realm. In the case of a home network there is one WAN IP address shared with multiple LAN IP addresses. The LAN IP addresses are often assigned within the `192.168.x.y` range, where the `x` and `y` can be any natural number between 0 and 254. The `192.168.x.y` range is specified by the RFC1918 [Rek96] as a IP address space for private internets. No WAN IP addresses will have an IP address within this address space. In this way it is possible to distinguish local traffic from the outgoing traffic by only observing the address of the packets.

A NAT can be deployed in various ways. RFC 1918 describes the following four variations:

Full Cone: A full cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host, by sending a packet to the mapped external address.

Restricted Cone: A restricted cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

Port Restricted Cone: A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

Symmetric: A symmetric NAT is one where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

A.3 Home network infrastructure

A home network infrastructure can consist of both a wired part and a wireless part. The wired part of the home network is typically an IEEE 802.3 network, also referred to as an Ethernet network.

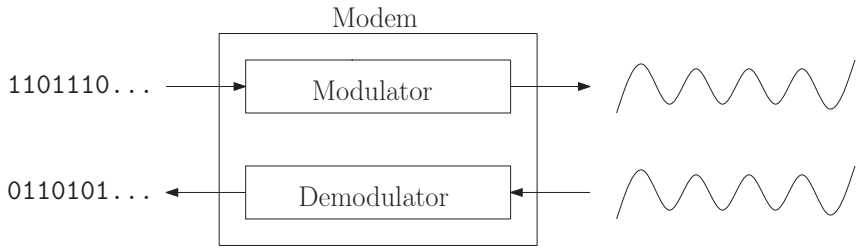


Figure A.2: Modem consisting of a modulator and a demodulator

Wireless networks within a home network are often Wi-Fi networks as defined in the IEEE 802.11 standard. IEEE 802.11 is a set of standards that describe Media Access Control (MAC) layer and Physical (PHY) layer specifications for the communication within a Wireless Local Area Network (WLAN). The wired Ethernet network and the wireless Wi-Fi network within a home network can form a single IP network. Any device which is connected either via Ethernet or via Wi-Fi can then connect to any other device on the network.

A.3.1 Modem

Modems exist in many varieties: telephone modems, DSL modems, cable modems, glass-fiber modems, wireless modems, etc. What these modems have in common is that they transform one type of signal into another type of signal. For instance, a cable modem transforms the signal sent over a coax cable into an Ethernet signal sent over an UTP-cable.

The term modem is short for *modulator demodulator* [TW10]. A modem has both a modulator and a demodulator part. The modulator transforms a digital signal into an analogue signal and sends it over the physical channel. The demodulator works the other way around, it receives an analogue signal from a physical channel and transforms it into a digital signal. The demodulation and modulation step are illustrated in Figure A.2.

A.3.2 Router

A router connects two or more computer networks, and is responsible for the routing of the traffic. The router typically handles traffic on the network layer (3rd OSI-layer) as packets. With destination IP address of each packet the router can determine to which network it has to be transferred to in order to reach its destination. For this the router maintains a table of the hosts that are connected to it, and to which socket these are connected.

In home networks a router is often combined with a Wi-Fi access point and a modem. Such all-in-one solutions are meant to make the router a central gateway device that connects the home network with the Internet.

A.3.3 Network hub

To increase the connectivity of a network a *network hub* can be used to expand the amount of Ethernet sockets available within the network. These devices are used as cheap variant of a network switch. All traffic that reaches the network hub on a certain socket will be forwarded to all other sockets. In this way the network hub only has to operate on the physical layer (1st OSI-layer) of the network. However, the undirected forwarding nature of a network hub has the disadvantage that more bandwidth is used than with a network switch.

A.3.4 Network switch

For an efficient way to increase the connectivity of a network a *network switch* can be used. Like a network hub a network switch is a device which connects other devices together on an Ethernet

network. The difference with a network hub is that when traffic is reaching the switch it's only forwarded to the sockets where it is addressed to. The advantage over a network hub is that less bandwidth is used on channels that are not on the route of traffic. Home network switches typically operate on the data link layer (2nd OSI-layer), so routing is done based on Media Access Control (MAC) addresses. There are also more advanced network switches available which operate on the network layer (3rd OSI-layer). These network switches are therefore able to do more sophisticated routing than the data link layer switches. Network switches can be divided into two different types. We have the *unmanaged switches* which are plug-and-play network switches without configuration options. Unmanaged switches are often used within home networks or small office networks since they are less expensive than the *managed switches* which do have configuration options. Common options for a managed switch to support is the ability to deploy Virtual Local Area Networks (IEEE 802.1Q), and network access control (IEEE 802.1X). Unfortunately these managed switches are typically fairly expensive, and will therefore not occur often within a home network.

A.3.5 Wi-Fi access point

In home networks it is common that Wi-Fi networks are deployed by Wi-Fi access points. It is also possible that client devices set up a connection between each other (i.e. ad-hoc), but this is not common within home networks. In the home network multiple access points can be deployed to increase the range of the Wi-Fi network. Wi-Fi access points are often combined with a router.

Appendix B

Wi-Fi networks

Home networks often deploy a Wi-Fi (IEEE 802.11) network. A Wi-Fi network is a wireless network which communicates using electromagnetic signals on the 2.4 or 5 GHz band. Within a home network the Wi-Fi network is deployed by one or more wireless access points. The wireless access point can be a dedicated device, but can also be included within the router.

The range of a Wi-Fi network depends on many factors such as, broadcasting power, the type of antenna, frequency-band used, and obstacles that are present in the area. To increase the range of the Wi-Fi network *Wi-Fi range extenders* can be used. These devices behave as the wireless equivalent of a network hub, they forward all traffic they receive from the AP to a client, but also the other way around. Because a wireless range extender is an extra hub on the network, it creates additional latency.

However, usually the range is not restricted to the border of a building. A device that is located outside building could potentially be able to connect to a Wi-Fi network inside a building. This device is also able to silently eavesdrop the traffic that is sent over the Wi-Fi network. Those devices outside a building are capable of receiving the Wi-Fi signals already show the necessity of security mechanisms within Wi-Fi networks.

In the next section, we will look into the IEEE 802.11 standard and its amendments that specify the first and second layer protocols of Wi-Fi. Then in Section B.2 we discuss the possible security configurations of Wi-Fi networks and their vulnerabilities.

B.1 IEEE 802.11 Standard

The IEEE 802.11 Standard defines a medium access control and several physical layer protocols which are used for wireless connectivity within LAN networks [Gro12]. Wi-Fi is actually a certification label for products for wireless networks which follow the IEEE 802.11 Standard. This standard is part of the IEEE 802 family which defines the network connectivity for local and metropolitan area networks. Within this family of standards can be found a lot of other types of connections such as: IEEE 802.3: Ethernet, IEEE 802.15: Wireless PAN, IEEE 802.16 Broadband Wireless Access. These members of the IEEE 802 family can have on their turn again sub-standards and form in this way a hierarchical structure of standards.

B.1.1 Amendments

The IEEE 802.11 standard itself has several sub-standards which are called amendments which are indicated by postfixed with one or more letters to the IEEE 802.11 name. Common used 802.11 amendments for home networks are IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, IEEE 802.11ac. There exist many more amendments but these are rarely used and supported by current consumer wireless network devices. The amendments support different frequencies, modulations and ultimately different data rates. The b,g and n amendments support transmitting

frequencies on the 2.4GHz band. The *n* amendment, next to the 2.4 GHz band also supports transmitting on the 5GHz band. The *a* amendment supports both 5GHz and 3.7GHz. The transmission on the licensed 3.7GHz band is only allowed within the United States. Finally, there is the *ac* amendment which solely operates on the 5GHz band.

B.1.2 Modulations

The original IEEE 802.11 standard supports wireless channels via radio signals or via infra-red [Var03]. For the radio signals the standard specifies three different modulations:

- Frequency Hopping Spread Spectrum (FHSS)
- Digital Sequence Spread Spectrum (DSSS)
- Orthogonal Frequency Division Multiplexing (OFDM)

However, FHSS is not used any more in current used amendments. The IEEE 802.11 amendments that are currently used in home networks use either the DSSS or the OFDM modulation. The *b* and *g* amendments support the DSSS modulation. The *a*, *g*, *n* and *ac* use the OFDM modulation. An overview of the most common IEEE 802.11 amendments is shown in Table B.1.

B.1.3 Operating modes

The IEEE 802.11 standard [Gro12] supports two types of operating modes for a wireless LAN.

Ad-hoc: An operating mode where a wireless station (referred to as STA in the standard) can communicate directly to another wireless station without the intervention of an access point (AP). A wireless station then creates a so called Independent Basic Service Set (IBSS) which is a set of all wireless stations which are able to communicate with each other. Each wireless station within this IBSS is identified by a Basic Service Set Identifier (BSSID) which is the Media Access Control (MAC) address of each wireless station. Wireless stations can either enter and leave the IBSS dynamically, this can happen when they turn on, turn off, come within range, or go out of range. When a wireless station is connected to another STA they are called *associated*, but this does not mean that the STA is authenticated.

Infrastructure: Is an operating mode where instead of an IBSS a Basic Service Set (BSS) is formed. Within this operating mode there is at least one wireless station which acts as an AP. An AP is connected to a Distribution System (DS) that takes care of transmitting the traffic over a different type of network than a Wi-Fi network. The DS can be any type of network, while in home it is usually an Ethernet network (IEEE 802.3).

A BSS in combination with a DS allows the creation of networks of arbitrary size and complexity. The IEEE 802.11 standard refers to this kind of network as the Extended Service Set (ESS) network. The ESS is formed as the union of all BSSs with the same Service Set Identifier (SSID) which are connected by a DS. Only the wireless 802.11 components of the network belong to the ESS, so the DS is not part of it. In Figure B.1 is shown a graphical overview of the IEEE 802.11 Components.

B.1.4 Connecting to a Wi-Fi network

Wi-Fi clients within an IEEE 802.11 managed network can be in three different states. When a client device enters the perimeter of a network is unauthenticated and unassociated. When a client device is unauthenticated it means that it has not yet proven its identity to the network. There can be multiple access points within the Wi-Fi network. A client device connects to a single access point. When the device successfully connects to such an access point it is called associated.

Typically the process of a client device connecting to a Wi-Fi network happens as follows. The access points in the Wi-Fi network broadcasts beacons to advertise their presence. Once the

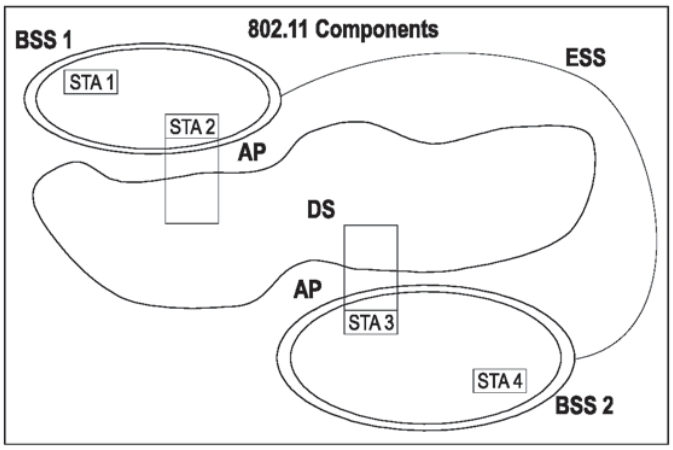


Figure B.1: ESS

client device receives such a beacon of an access point with a known SSID, it will try to connect to it. Depending on the authentication protocol the Wi-Fi network uses, the device will send an authentication request to the access point. If the client device is allowed to access the network the AP will reply with a successful authentication message. From this point on the device is authenticated but is still not associated. Now the client device will send an association request in order to connect to an access point. If the access point and client device are compatible with each other the access point will respond with a 'successful association' message. The state diagram shown in Figure B.2 shows an overview of the possible states.

By design, there are already some weaknesses within the IEEE 802.11 protocol. The first one is that the authenticity of access points is usually not guaranteed, especially not in home networks. When a device once connected to a Wi-Fi network it will remember the SSID of that network and will usually connect to it automatically when it is within range. This functionality can be very convenient and it contributes to a ubiquitous experience of networks. But the downside is that someone could set up a rogue network with the same SSID and the same security settings. When the device detects this rogue network it will assume that it is the network it connected to before and will try to connect to it. If the original network is secured with a password it will probably fail in connecting to the rogue network because the rogue network does not have the original credentials. As we will see within Section B.2.4 that even with the current strongest Wi-Fi security protocol WPA2, enough information can be extracted from the failed connection-attempt to perform an offline dictionary or brute-force attack on the password.

B.1.5 Summary

In the years there are introduced many different amendments to the IEEE 802.11 standard. The purpose of these amendments was typically to speed up the connection, but only a few were implemented by manufacturers. Each amendment supports either DSSS or OFDM or both and can operate on a certain frequency band. An overview of the most widely used IEEE 802.11 amendments are shown in Table B.1.

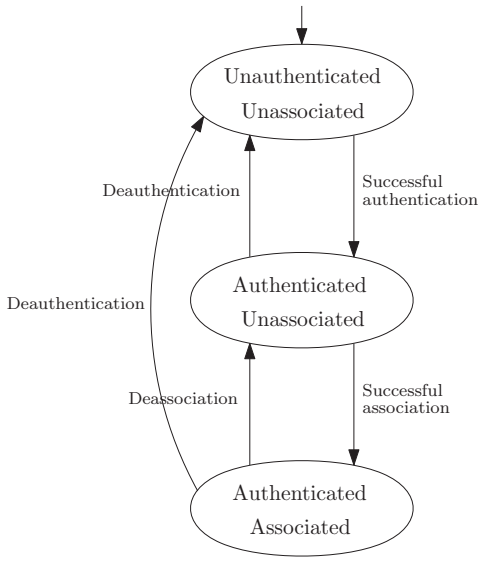


Figure B.2: State diagram of IEEE 802.11 connection process.

802.11 Amendment	release date	frequency (GHz)	bandwidth (MHz)	modulation	data rate (Mbit/s)
a	Sep 1999	5	20	OFDM	6-54
		3.7			
b	Sep 1999	2.4	22	DSSS	1 - 11
g	Jun 2003	2.4	20	OFDM / DSSS	6 - 54
n	Oct 2009	2.4 / 5	20	OFDM	7.2 - 72.2
			40		15 - 150
ac	Dec 2013	5	20	OFDM	7.2 - 96.3
			40		15 - 200
			80		32.5 - 433.3
			160		65 - 866.7

Table B.1: Current used IEEE 802.11 amendments

B.2 Configuration of the Wi-Fi network

Many attacks on Wi-Fi networks depend on a poor configuration. In the following sections we will discuss the four major variations of Wi-Fi configuration:

- Open network
- Wired Equivalent Privacy (WEP)
- Wi-Fi Protected Access (WPA)
- Wi-Fi Protected Access 2 (WPA2)

To stay compatible with older Wi-Fi devices, access point manufacturers keep supporting old (insecure) security protocols such as WEP.

B.2.1 Open network

A Wi-Fi network can be set up as an open network where it does not use any Wi-Fi security protocol. Often this is done at public places such as bars, restaurants, libraries etc. in order to provide easy Internet access to their visitors. Using open networks may seem very convenient but comes also with some security threats:

- *Authentication:* There are very few authentication possibilities within an open network. Devices are distinguished only by their MAC address. It is possible to apply MAC filtering in order to restrict which device can access the Wi-Fi network. This is not a very reliable authentication method since MAC addresses can be spoofed.
- *Eavesdropping and Interception of wireless traffic:* The traffic which is send over the air is not encrypted by the wireless network protocol. Anyone within the perimeter of the network is able to capture the network traffic. This means that the confidentiality of the data has to be preserved by other layers of the network stack. As mentioned in the traffic eavesdropping category it is possible to set up a rogue access point and perform a man-in-the-middle attack. There is no reliable verification mechanism available for the authenticity of open access points. A client device may remember the MAC addresses of the previously used access points and only connect to the open network with known APs. This is not a good solution because the attacker could spoof the MAC address of its rogue access point into the MAC address of a legitimate access point to circumvent this countermeasure. Another reason why this solution would not work in general is that there are open networks which have multiple access points to make the connection available in a big building. In this way the client device can only connect to the access points which he has connected to before by explicit allowance of the user. The user does not have a method to verify if the access point can be trusted.

  An open network does not provide any protection against network injection. It is possible that a device injects frames into the network with the same MAC and IP address of a legitimate device. Because of the ability of packet injection it is also possible to capture traffic and send it later on again into the network. So an open network does not provide any protection against replay attacks.

  The confidentiality and integrity of the communication over an open Wi-Fi network has to be provided by higher layer protocols.
- *Rogue access point:* An attacker could set up a rogue access point with the SSID as the legitimate network. A device which has previously connected to the legitimate open access point trusts any network with that same SSID and will often try to connect to it automatically. In this way, an attacker is able to perform a man in the middle attack because he has full control over the rogue access point and can therefore eavesdrop and modify the traffic passing by.

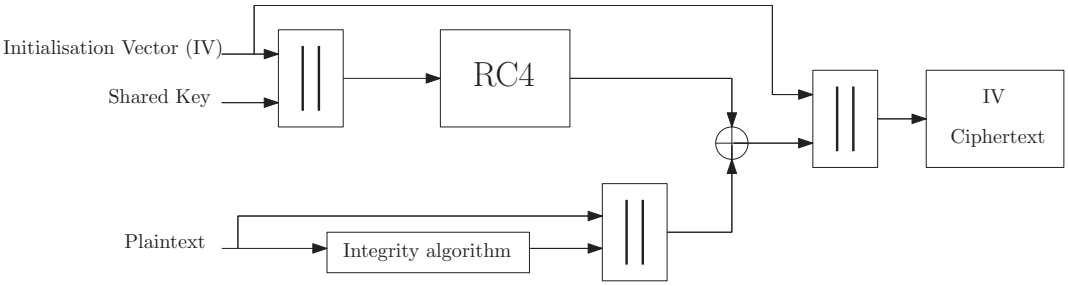


Figure B.3: WEP encryption process

In order to mitigate these vulnerabilities a VPN connection can be used. A VPN connection can set up an encrypted channel to communicate with a trusted network. Some VPN protocols, however, are vulnerable to a man-in-the-middle attack that allow an attacker to drop the encryption.

B.2.2 Wireless Equivalent Privacy (WEP)

Wireless Equivalent Privacy (WEP) was one of the first security mechanisms for wireless LAN, that was introduced in September 1999 as part of the IEEE 802.11 security standard [KSS14]. The idea behind WEP was to provide a mechanism for security which is equivalently strong as that of wired networks. The WEP security mechanism can operate as an open system where it permits all clients access to the network. This operating mode does not provide authentication security. Next to the open system operating mode there is the shared key system where a client requires to have the WEP encryption key to be able to access the network.

To encrypt the traffic between the access point and the client WEP uses a RC4 symmetric stream cipher algorithm. The IEEE 802.11-1999 standard specifies a 64-bit encryption, but later it also supported 128-bit [BHL06]. When sending a packet, a 24-bit Integration Vector (IV) will be chosen at random. This IV will be appended by the pre-shared key which is 104-bit long in the case of 128-bit WEP encryption. Together this will produce a 128-bit seed for the RC4 cipher which will on his turn produce a key sequence. In parallel a CRC-32 checksum will be calculated of the message that has to be sent (also called plain-text). This checksum will then be appended with the plain-text Now we have both a key-stream and a plain-text with a checksum. These two strings will be binary XOR-ed, which results in the encrypted message, called the cipher-text Finally the cipher-text will be appended to the IV. The IV is sent in plain-text because it is needed for the receiver to be able to decrypt the message. The WEP encryption process is shown graphically in Figure B.3 [Vac12].

The decryption process works quite similar. The receiver knows the pre-shared key and receives the cipher-text and the IV. First, the receiver will append the pre-shared key to the IV to create the seed for the RC4 cipher.
Now the RC4 cipher can reproduce the correct key-sequence. Then, the ciphered and the generated key-sequence is binary XOR-ed to retrieve the plain-text with the checksum appended to it.
This works because XOR-ing a plain-text P twice with the same key K will result in the original value (i.e. $P = (P \oplus K) \oplus K$). Note that the only thing that is sent over the air is the cipher-text which is $P \oplus K$. Finally, there will be calculated a CRC-32 checksum over the decrypted plain-text and if the calculated checksum is equivalent to the received checksum it the message will be accepted and the decryption is successful.

B.2.2.1 Vulnerabilities

There are known multiple vulnerabilities for the WEP encryption. It is already known for more than a decade that WEP is insecure and is therefore deprecated since then. Yet, current network equipment often still supports WEP. The vulnerabilities of WEP come from the incorrect use of

the RC4 stream cipher and the poor choice of the CRC-32 checksum to validate data integrity.

One of the weak spots of WEP encryption is the IV. For each packet a 24-bit IV is chosen at random. This IV is appended by the pre-shared key and is then used as a seed for the RC4 stream cipher. Because the pre-shared key is constant, there are 2^{24} different key streams. With enough network traffic the probability is high that the same IV is used twice, which makes it possible to reveal multiple parts of the key stream. In [BHL06] is even noted that in practice clients only use a limited number of IV's to generate the key-streams, increasing the probability to get twice the same IV. In the end, it becomes possible to determine the authentication key used.

According to [BHL06] there is another vulnerability present within the *Shared key Authentication* mechanism. When a client wants to authenticate the AP sends a clear-text challenge to the authenticating client. The client responds by sending the encrypted version of the challenge. Now, both the clear-text and the cipher-text are transmitted over the air which makes it able to retrieve a part of the key-stream of a certain IV. In the IEEE 802.11 standard this vulnerability is already known and the re-use of that IV is discouraged.

The weaknesses of WEP are summarized as follows:

- The 24-bit IV is too short. The probability that the IV will be reused within a short time is high with sufficient amount of traffic.
- The method to create a key using the IV is susceptible to weak keys.
- No protection against message replay.
- The message integrity is insufficiently protected. In this way fake messages could be forged.
- The authentication key uses the master key, where there is no mechanism built in to update the keys.

Further Martin Beck and Erik Tews describe in their paper [TB09] the chop-chop attack on the WEP protocol. With this attack it becomes possible to decrypt a packet by resending slightly modified versions of the packet and observing the response of the access point. Before the data is encrypted a 4 byte CRC-32 checksum (which is referred to as the ICV) is created and appended to the data. Then the both the data and the checksum is encrypted. The chop-chop attack starts by capturing a packet and removing the last byte of data of it. Now with high probability the ICV value is not correct any more, which has to be corrected before the packet can be resent. Since the attacker does not have the plain-text of the package he can only guess the value of the removed byte. The attack will then try every possibility of that byte, calculate the new ICV value of that guess and send it to the access point. At some point the packet will be accepted and the plain-text byte is known by the attacker. The attacker can now proceed with determining the next bytes of the packet until the entire packet is decrypted.

B.2.3 Wi-Fi Protected Access (WPA)

The WPA security protocol was used as an intermediate solution before the IEEE 802.11i standard was ratified. It was introduced to patch the vulnerabilities of WEP while still using the same hardware. To apply these patches WPA implements the Temporal Key Integrity Protocol (TKIP) on top of the WEP architecture. TKIP is designed to be compatible with the legacy hardware of WEP.

For every packet that is sent using the WPA protocol a different key is generated. Instead of the WEP method of appending the pre-shared key to the IV, the key is determined by a key mixing algorithm which uses a TKIP sequence number, the transmitters address and an encryption key as input. In parallel, to ensure data integrity and protection against forgery attacks the WPA protocol generates a Message Integrity Code (MIC) using a 64-bit Michael key and the plain-text packet. Within WPA terminology this plain text packet is called the MAC Service Data Unit (MSDU). After the MIC of the MSDU is calculated the MSDU, MIC and the TKIP sequence number age given as input to a module which will fragment the MSDU into smaller MAC Protocol

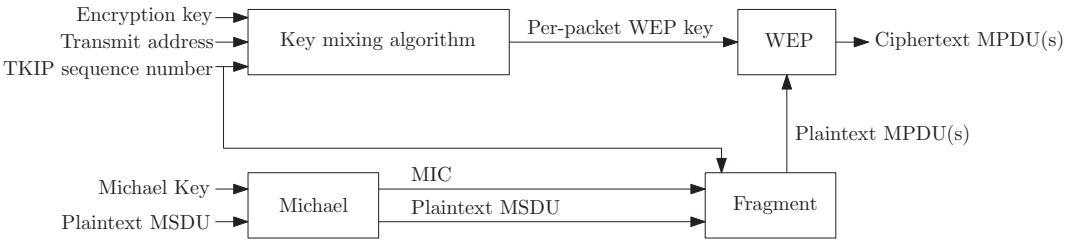


Figure B.4: WPA encryption process

Data Units (MPDU). Then the resulting MPDU packets together with the per-packet WEP key will be used as input for the WEP protocol. This process is graphically summarized in Figure B.4.

This architecture is designed to mitigate the weaknesses of the WEP protocol. To reduce the probability of using an already used IV they increased its size. The weak key attacks are made more difficult because each packet gets a different encryption key. And the integrity of the packets is increased using the Michael algorithm which produces a MIC.

There are two variants of WPA: WPA-PSK, which uses a pre-shared key and WPA-Enterprise, which lets clients authenticate with their own credentials using the Extensible Authentication Protocol (EAP). The WPA enterprise variant is harder to set up because use of EAP requires setting up an authentication server system which is able to check the credentials.

B.2.3.1 Vulnerabilities

Since WPA uses the same hardware as WEP, it is still vulnerable for some attacks which were already known for WEP. For instance packets still can be decrypted using the Beck and Tews attack which is a variation of the chop-chop attack on WEP [VP13]. As with WEP, the chop-chop attack this attack determines byte by byte the plain-text of a packet. To mitigate the vulnerability for the Beck and Tews attack a short re-keying interval should be used (e.g. 120 seconds) [TB09].

B.2.4 Wi-Fi Protected Access II (WPA2)

In 2004 the IEEE ratified the 802.11i standard which is referred to as Wi-Fi Protected Access II (WPA2). WPA2 is not backwards compatible with WEP and WPA since different hardware is required. This standard is completely redesigned in order to address the weaknesses in WEP and WPA. As with WPA, it is possible to use WPA2 within two modes: the pre-shared key (PSK) mode or the enterprise mode which uses IEEE 802.1X authentication. The PSK mode uses a 8 to 63 characters long passphrase to generate a 256-bit PSK. Every client device which connects to the network uses the same passphrase. This makes it convenient to use within a home network, but also creates a security issue. If someone knows the passphrase of a WPA2 network he is able to decrypt the traffic that he is able to receive.

The enterprise mode of WPA2 requires an authentication server as extra party. The authentication server manages the credentials for every user. In this way it is not possible any more to decrypt all the traffic of the network when the credentials of one user are confiscated.

WPA2 can be set up to use either TKIP and CCMP encryption or both. When both methods are used, CCMP will be preferred but TKIP will be used as a fall-back. TKIP in combination with WPA2 works the same as with WPA which is discussed in Section B.2.3.

The WPA2 protocol builds a secure communication context between an AP and a client with the 4-way-handshake shown in Figure B.5. The 4-way-handshake makes both the access point and the client prove that they know the Pairwise Master Key (PMK). If the network is configured with a pre-shared key (PSK), the PMK will be computed out of the PSK.

Now we will give a rough description of the 4-way-handshake, based on the explanation given by Changhua He et al. in [HM04]:

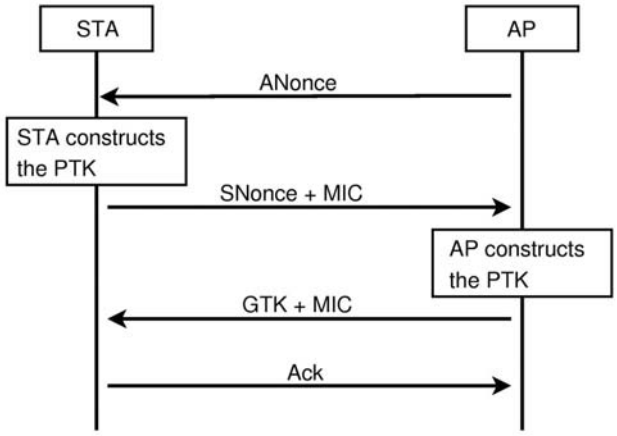


Figure B.5: WPA2 4-way handshake

1. **AP sends ANonce**: The access point sends a random generated nonce (ANonce) to the client.
2. **Client sends SNonce and MIC**: The client responds with another random generated nonce (SNonce) and a Message Integrity Code (MIC). The most important elements that are used to calculate the MIC are: PMK, SNonce, and a sequence number.
3. **AP sends GTK and MIC** The access point sends an encrypted version of the GTK and a MIC. This time the most important input attributes are: PMK, ANonce, GTK and the sequence number increased by one.
4. **The Client sends an ACK message**. The last phase of the 4-way-handshake is an ACK message.

B.2.5 Wi-Fi Protected Setup (WPS)

In 2006 the Wi-Fi Protected Setup (WPS) standard was introduced to ease the way to connect devices to home network access points. Originally the user has to provide the ESSID (i.e. name of the network) and the pre-shared key in order to get access to the network. WPS solves this 'problem' of having to enter passphrases by providing three additional ways to authenticate a device[All14]:

Pin: An eight digit pin can be used to authenticate to the network. This pin is often printed on the access point. Theoretically this ping would have 10^8 different possibilities. However, due to a design flaw it is possible to check if the first four digits of a pin is correct [Vie11]. Another weak spot is that the last digit of the pin is used as a checksum. This means that the number of possible pins reduces from 10^8 to $11.000 \approx 10^{4.04}$ combinations. According to the experiments done by Stefan Viehböck it takes worst case 3.97 hours to try all combinations if the access point does not use a lock-down mechanism. A lock-down mechanism is a mitigation for a brute force attack. Af a number attempts the access point will not respond to an authentication attempt. However, this lock-down mechanism is not always implemented by router manufacturers.

Push button connect: The user has to push a button on both the access point and the device that has to be connected in order make the connection. The push button connect functionality then be enabled until the new device is connected or a time-out of two minutes is superseded.

Property	WEP	WPA	WPA2
Cipher	RC4	RC4	AES
Key size	40/108 bit	128 bit encryption 64 bit authentication	128 bit
Key life	24-bit IV	48/128 bit IV	48/128 bit IV
Packet key	Concatenated	Mixing function	Not needed
Data integrity	CRC-32	MIC (Michael)	CCM
Replay detection	None	Enforce IV sequencing	Enforce IV sequencing
Header integrity	None	MIC (Michael)	CCM
Key management	None	EAP based (802.1X)	EAP based (802.1X)

Table B.2: Comparison of Wi-Fi security protocols [MH07]

Near Field Communication (NFC): WPS has two variants of authentication via Near Field Communication (NFC). The first variant is to set up a Wi-Fi Direct connection between two NFC devices to transfer data over Wi-Fi between these devices. The idea is that the two devices are tapped together and that they are able to set up a secure network together.

The other variant uses NFC tags that provide network settings to connect to a network. Suppose a user wants to connect its device to the Wi-Fi network. He then only has to tap his device on the NFC tag in order to connect to it. The NFC tag could be attached to the access point but can also be separate. Separate NFC tags can be very convenient to connect immovable devices such as air conditioners, thermostats etc. to the Wi-Fi network. The user only has to bring the mobile NFC tag to the immovable device to connect it to the Wi-Fi network.

From a security viewpoint there could be an advantage of using such NFC tags over a shared password to authenticate to a Wi-Fi network. Suppose only these NFC tags are used for authentication, then a maximum strength password as pre-shared key can be chosen without the user having to remember it. A maximum strength password for WPA2 would be a 63 character password with random characters. In this case a malevolent user has to gain physical access to the NFC tag in order to connect to the network, performing a brute force attack on the pre-shared key would take a very long time (e.g. multiple years).

There are many devices on the market which have the WPS functionality enabled by default. For some of these devices it is not even possible to disable WPS functionality.

B.2.6 Overview IEEE 802.11 security protocols

Table B.2 shows an overview of the IEEE 802.11 security protocols. During the iterations from WEP, to WPA to WPA2 it can be observed that the key size is increased, that the RC4 cipher has been replaced with the AES cipher, and that the replay vulnerability of WEP is fixed. This all happened because of increased computing power and the vulnerabilities that have been found in WEP and WPA.

Appendix C

Malware

Malware is short for malicious software which can be used by an attacker to get access to a machine. The National Institute of Standards and Technology (NIST) defines malware as follows:

> "A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity or availability of the victim's data, applications, or operating or otherwise annoying or disrupting the victim." [MKN05]

There are many forms of malware, where some are less dangerous than others. *Adware* for instance, is used to show additional advertisements to the user and is therefore less severe than *spyware* which actively collects private information and sends it to a server. The following list shows common known variants of malware.

Adware: Malicious software which gets installed on a victim machine in order to show unwanted advertising. Some adware adds additional advertisements embedded within sites or generates popups.

Back-door (trapdoor): A program or mechanism in order to grant a malicious user access to the system. Back-doors can be installed by attackers when they have gained access to a system in order to retain access to the system or to make access easier.

Flooder (DoS Client): Generate a large volume of data to attack networked computer systems.

Key-loggers: Captures keystrokes on a compromised system. A program which could be installed by an attacker in order to log all the keystrokes of the user. Key-loggers are often used to steal login credentials and other personal information.

Logic bomb: Software which can be seen as the digital version of a ticking bomb. The software will be dormant until a predefined condition is met. Then it is triggered and will perform an unauthorized act.

Ransomware: Malicious software which makes files unreadable for the user by encrypting them. Often it searches for pictures or documents in order to digitally kidnap them. The user then gets blackmailed by the attacker, where the user has to pay money to get his files back.

Rootkit: Set of hacker tools that enable an attacker to obtain root-level access to a system.

Rogueware / scareware: Is malicious software which poses to be a known, or at least a legitimate software product. A common form of rogueware is a fake virus-scanner which pretends as it has found malware on the users machine. In this way it tries to scare the user into buying their fake virus-scanner.

Spyware: Malware which is used to collect information from a compromised machine and sends this to another system. This kind of software spies on its victims with key-loggers, screen or webcam capturers, or by scanning files on the system for sensitive information.

Trojan horse: A computer program which appears to have a useful function, but also has a hidden and potentially malicious function.

Virus: A form of malware which, when executed, tries to replicate itself into other executable machine or script code. Viruses have the property that they 'infect' files or programs in order to work; i.e. not a standalone executable but make themselves part of an existing file or executable.

A virus has typically four phases during its lifetime.

1. *Dormant phase:* The virus is idle, until it will be activated by some event.
2. *Propagation phase:* The virus infects another program or file by copying itself to it. To evade detection some viruses use techniques to hide their presence.
3. *Triggering phase:* The virus will remain idle until some triggering event takes place.
4. *Execution phase:* After the virus is triggered it will perform the function for which it was intended.

Worm: A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.

Zombie, bot: Can be activated to attack other machines. This kind of malware is most of the time idle, but can be activated remotely. It is often used to perform large scale Distributed Denial of Service (DDoS) attacks.

Not all malware can be categorized into a single variant, some may combine several variants of malware.

C.1 Motivation for malware

A lot malware is developed the last decades. But what is this malware used for? A rational attacker would want to get any benefit out of creating and spreading malware. This benefit could be in the form of money or activism.

The intrusive possibilities a malware depend on the knowledge of its designer about the vulnerabilities present in the system which the malware is designed for. Where one vulnerability could give the attacker only limited additional access, the other vulnerability will grant the attacker full root access to the system.

A goal for malware could be:

- Corrupt system or data files.
- Steal personal information (e.g. passwords, credit card information)
- Recruit a system in order to participate within a bot-net.
- Steal computational resources as with the phenomena of bitcoin mining malware [1].

C.2 Detection of malware propagation

Different forms of malware have different ways of propagating. Not all types of malware have a propagation mechanism built in. Some rely on external propagation mechanisms such as drive-by downloads on websites or Trojan horses sent attached to an e-mail. NIST states that only viruses and worms are in principle self-replicating [MKN05].

Some self-replicating methods could be detected by performing network traffic analysis. For instance if a host suddenly starts to connect to many IP addresses within the network it could be an indication of scanning activities.

[1] `http://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/93/cybercriminals-unleash-bitcoinmining-malware`

Bibliography

[All14] Wi-Fi Alliance. Wi-fi certified wi-fi protected setup - easing the user experience for home and small office wi-fi networks. `http://www.wi-fi.org/file/wi-fi-certified-wi-fi-protected-setup-easing-the-user-experience-for-home-and-small-office-wi`, March 2014.

[Axe00] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. Technical report, Technical report, 2000.

[Bar01] Richard Barber. Hackers profiled?who are they and what are their motivations? *Computer Fraud & Security*, 2001(2):14–17, 2001.

[BHL06] Andrea Bittau, Mark Handley, and Joshua Lackey. The final nail in wep's coffin. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–pp. IEEE, 2006.

[CH13] Yulia Cherdantseva and Jeremy Hilton. A reference model of information assurance & security. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 546–555. IEEE, 2013.

[DMR10] Matteo Dell'Amico, Pietro Michiardi, and Yves Roudier. Password strength: An empirical analysis. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[DZM05] Dino A Dai Zovi and Shane A Macaulay. Attacking automatic wireless network selection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 365–372. IEEE, 2005.

[Fur07] Steven Furnell. An assessment of website password practices. *Computers & Security*, 26(7):445–451, 2007.

[GK] Hemant Chaskar Gopinath K.N. All you wanted to know about wifi rogue access points.

[Gro12] IEEE 802 Standard Working Group. Wireless lan medium access control (mac) and physical layer (phy) specification. IEEE Std 802.11-2012, IEEE, 2012.

[Gui06] NIST Electronic Authentication Guideline. Nist special publication 800-63 version 1.0. 2, 2006.

[Hef10] Craig Heffner. Remote attacks agains soho routers, 2010.

[HM04] Changhua He and John C Mitchell. Analysis of the 802.11 i 4-way handshake. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 43–50. ACM, 2004.

[HSHR09] Fabian Hugelshofer, Paul Smith, David Hutchison, and Nicholas JP Race. Openlids: a lightweight intrusion detection system for wireless mesh networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 309–320. ACM, 2009.

[IM07] Nwokedi Idika and Aditya P Mathur. A survey of malware detection techniques. *Purdue University*, 48, 2007.

[KSS14] Lakshmi Kurup, Ms Vidhi Shah, and Mr Dhaval Shah. Comparative study of attacks on security protocols. *identity*, 3(8), 2014.

[LLLT13] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

[LYLO03] Yu-Xi Lim, TS Yer, John Levine, and Henry L Owen. Wireless intrusion detection and response. In *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, pages 68–75. IEEE, 2003.

[MH07] Moffat Mathews and Ray Hunt. Evolution of wireless lan security architecture to ieee 802.11 i (wpa2). In *Proceedings of the fourth IASTED Asian Conference on Communication Systems and Networks*, 2007.

[MKN05] Peter Mell, Karen Kent, and Joseph Nusbaum. *Guide to malware incident prevention and handling: Recommendations of the National Institute of Standards and Technology*. US Department of Commerce, National Institute of Standards and Technology, 2005.

[OK14] Yossef Oren and Angelos D Keromytis. From the aether to the ethernet–attacking the internet using broadcast digital television. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA*, pages 353–368, 2014.

[Pas06] Brian A Pashel. Teaching students to hack: ethical implications in teaching students to hack at the university level. In *Proceedings of the 3rd annual conference on Information security curriculum development*, pages 197–200. ACM, 2006.

[PCGE08] Erika Shehan Poole, Marshini Chetty, Rebecca E Grinter, and W Keith Edwards. More than meets the eye: transforming the user experience of home network management. In *Proceedings of the 7th ACM conference on Designing interactive systems*, pages 455–464. ACM, 2008.

[Rek96] et al Rekhter. Rfc 1918. `http://tools.ietf.org/html/rfc1918`, February 1996.

[Sha51] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

[She04] Oleg Mikhail Sheyner. *Scenario graphs and attack graphs*. PhD thesis, School of Computer Science, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, April 2004.

[Shi00] R. Shirey. RFC 2828 - Internet Security Glossary. RFC 2828, May 2000.

[SSVE04] Andrew Simmonds, Peter Sandilands, and Louis Van Ekert. An ontology for network security attacks. In *Applied Computing*, pages 317–323. Springer, 2004.

[Sta10] William Stallings. *Network Security Essentials: Applications and Standards*. Prentice Hall Press, Upper Saddle River, NJ, USA, 4th edition, 2010.

[TB09] Erik Tews and Martin Beck. Practical attacks against wep and wpa. In *Proceedings of the second ACM conference on Wireless network security*, pages 79–86. ACM, 2009.

[TW10] Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edition, 2010.

[TWP07] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit wep in less than 60 seconds. In *Information Security Applications*, pages 188–202. Springer, 2007.

[Vac12] John R Vacca. *Computer and information security handbook*, chapter 15. Newnes, 2012.

[Var03] Upkar Varshney. The status and future of 802.11-based wlans. *Computer*, 36(6):102–105, 2003.

[Vie11] Stefan Viehböck. Brute forcing wi-fi protected setup. `https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf`, December 2011.

[VP13] Mathy Vanhoef and Frank Piessens. Practical verification of wpa-tkip vulnerabilities. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 427–436. ACM, 2013.

[Š14] Boris Škorić. *Physical aspects of digital security - Lecture Notes*. TU Eindhoven, Dept. of Mathematics and Computer Science, Security Group (SEC), 2014.

[WM11] M. Whitman and H. Mattord. *Principles of Information Security*. Cengage Learning, 2011.

[YG13] Chao Yang and Guofei Gu. Security in wireless local area networks. In *Wireless Network Security*, pages 39–58. Springer, 2013.