MASTER

Securing the home network

Stelma, J.

Award date:
2015

Link to publication
Securing the Home Network

Master Thesis

Jaap Stelma

Supervisor:
Dr. D.S. (Dmitri) Jarnikov PDEng

Graduation Committee:
Prof. Dr. J.J. (Johan) Lukkien
Dr. D.S. (Dmitri) Jarnikov PDEng
Dr. J.I. (Jerry) Hartog, den

Version:
1.2 (10 August 2015)

Publication Date:
31 August 2016

Eindhoven, August 2015
Abstract

Network security is protection and precaution taken against breaches of confidentiality, integrity, availability, authenticity, and accountability. The core technologies required to protect against the threats are identification and access control. In home networks, identities of devices can easily be spoofed and access control is not usable. To overcome these challenges we introduce a centrally managed firewall with identity protection.

For identification of devices we have two possible improvements: use a more secure identification method or protect the identities. Changing the identification method of devices is hard when no modification to the devices is allowed. Protection, however, is possible. Using device and network profilers, we can detect spoofed and unknown devices in the network. This provides us with the possibility to act against the harmful devices.

The second core technology we address for network security is access control. Firewalls are the main technology for access control between devices. Most, but not all, networked devices already provide a firewall, but in a disabled state. Users often lack the knowledge to correctly configure the firewalls and, therefore, leave them default. In order to overcome the management problem of firewalls, we centralize the configuration.

Using the centralized configuration, home users have one location to configure the firewalls for all systems in a central location. To overcome the usability challenges experienced in configuring firewalls, we also replace the detailed policies used to manage firewalls by usage scenarios such as Browsing the web. In the background, these scenarios are converted back to the policies required for the enforcement with firewalls.

Enforcement of the policies is delegated as much as possible to devices connected in the network. Since not all devices support firewalls or remote configuration of them, we also have to enforce some policies centrally. To enforce communication policies for unsupported devices, we isolate the unsupported devices and force them to communicate through a central node which has the firewall policies deployed. The combination of isolation and distributed firewalls introduces at least one managed firewall in the path of all possible communication links in the network. Hence, this system can provide access control between all devices connected to the network.
## Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
</tbody>
</table>

### 1 Introduction
- 1.1 Motivation ........................................ 1
- 1.2 Problem Description ................................... 1
  - 1.2.1 Constraints .................................... 2
- 1.3 Relevant Work ...................................... 2
- 1.4 Outline .......................................... 3

### 2 Network Security
- 2.1 Security .......................................... 5
- 2.2 Network Security ..................................... 5
- 2.3 Typical Attacks .................................... 6
  - 2.3.1 Passive Attacks .................................. 7
  - 2.3.2 Active Attacks .................................. 7
  - 2.3.3 Non-aggressive Active Attacks ......................... 7
  - 2.3.4 Attack Schemes .................................. 7
    - 2.3.4.1 Directed Attack ............................ 8
    - 2.3.4.2 Undirected Attack ........................... 8
- 2.4 Network Security Threats ................................ 8
- 2.5 Security Measures .................................... 8
  - 2.5.1 Scenarios ..................................... 9
    - 2.5.1.1 Scenario 1: Guest Devices ....................... 9
    - 2.5.1.2 Scenario 2: No Internet ........................ 10
    - 2.5.1.3 Scenario 3: Limited local network ............... 10
    - 2.5.1.4 Scenario 4: No access from the internet ........... 10
    - 2.5.1.5 Scenario 5: a web service (http + https) is accessible from the internet .......... 11

### 3 Home Network
- 3.1 Home Networks ...................................... 13
  - 3.1.1 Endpoints ..................................... 13
  - 3.1.2 Infrastructure ................................... 14
    - 3.1.2.1 Infrastructure Devices ........................ 14
    - 3.1.2.2 Wired Connections ........................... 15
    - 3.1.2.3 Wireless Connections ........................ 15
  - 3.1.3 Topology ..................................... 15
- 3.2 Technologies for Security ................................. 16
  - 3.2.1 Services ...................................... 16

---

Securing the Home Network v
CONTENTS

3.2.1.1 Authentication ................................ 16
3.2.1.2 Access Control ................................ 17
3.2.1.3 Data Confidentiality .......................... 17
3.2.1.4 Data Integrity ................................ 17
3.2.1.5 Nonrepudiation ............................... 18
3.2.1.6 Completeness ................................. 18
3.2.2 Mechanisms ...................................... 19
3.2.2.1 Encipherment ............................... 19
3.2.2.2 Digital Signature ............................ 21
3.2.2.3 Access Control ............................... 21
3.2.2.4 Data Integrity ............................... 21
3.2.2.5 Authentication Exchange ....................... 22
3.2.2.6 Traffic Padding ............................... 22
3.2.2.7 Routing Control ............................. 22
3.2.2.8 Notarization ................................. 22
3.3 Challenges with Current Security .......................... 23
3.3.1 The Mechanisms in Home Networks ...................... 23
3.3.2 General Challenges ................................ 24
3.4 Security Performance for the Home Network ................. 24

4 Solutions .............................................. 27
4.1 Missing Root of Trust .................................. 27
4.1.1 Authentication Details ............................. 27
4.1.1.1 Solution 1: Stronger policies ..................... 27
4.1.1.2 Solution 2: Time-based One-time Passwords ...... 28
4.1.1.3 Solution 3: Prevent sniffing ..................... 28
4.1.1.4 Solution 4: Proof without sending the credentials 28
4.1.2 Cryptography ..................................... 28
4.1.2.1 Solution 1: PKI ................................ 28
4.1.2.2 Solution 2: Trusted communication ............. 29
4.1.3 Possession of Entities ............................. 29
4.1.3.1 Solution 1: Advanced fingerprints .............. 29
4.1.3.2 Solution 2: Detect & Evade .................... 30
4.1.3.3 Solution 3: Prevent .......................... 30
4.1.4 One-time Passwords ............................... 30
4.1.5 Applying Authentication to the Home Network ............ 31
4.2 Configuration ........................................ 31
4.2.1 Usability ......................................... 31
4.2.2 Restricting Access ............................... 32
4.2.2.1 Separating networks .......................... 32
4.2.2.2 Firewall ...................................... 34
4.2.2.3 Deployment ................................. 35
4.2.2.4 Remote Management .......................... 38
4.2.2.5 Availability ................................. 38
4.2.3 Policies .......................................... 39
4.2.3.1 Information need ............................. 39
4.2.3.2 Availability of information ..................... 40
4.2.3.3 Translating the Information .................... 41
4.2.4 Applying Access Control to the Home Network ............ 41
## System Design  

5.1 Identity Management  
5.1.1 Device Profile  
5.1.2 Device Profiler  
5.1.3 Network Profiler  
5.1.4 Device Groups  
5.1.5 Use cases and Usage Scenarios  
5.1.5.1 Registration Process  
5.1.5.2 Spoofed Identity  
5.1.5.3 New Service Discovered  
5.1.5.4 Unknown Device event  

5.2 DHCP Server  
5.2.1 Subnet  
5.2.2 Lease  
5.2.3 Lease Provider  
5.2.4 Use cases and Usage Scenarios  

5.3 Access Control Mechanism  
5.3.1 Policy  
5.3.2 Scenario  
5.3.3 Scenario Templates  
5.3.4 Translator/Pusher  
5.3.5 Use Cases and Usage Scenarios  
5.3.5.1 New Service Discovered  

5.4 Event Manager  
5.4.1 Basic Event  
5.4.2 Specific Events  

5.5 Review of Design  
5.5.1 Mechanisms  
5.5.2 Scenarios  
5.5.3 Assessment  

## Conclusion  

6.1 Overview of Our Work  
6.2 Limitations and Future Work  
6.3 Concluding Remarks  

Bibliography  

Appendix  

A Class diagrams  

B DHCP Lease Change  

C Template Scenarios  

D Events  
D.1 Identity Management Service  
D.2 DHCP  
D.3 Access Control Mechanism
List of Figures

2.1 The Security Requirements Triad (CIA triad) ..................... 6
3.1 Network topology ........................................... 16
4.1 virtual separated networks through subnets .................... 33
4.2 Performance and security of different level of firewalls ......... 35
4.3 Central access control ....................................... 36
4.4 Inline access control ....................................... 37
4.5 Endpoint access control ..................................... 38
5.1 Component overview ......................................... 43
5.2 Component overview for the identity manager ................... 44
5.3 Use case diagram for the registration process .................. 47
5.4 Component overview for the DHCP server ........................ 48
5.5 Decision tree for the lease type ................................ 49
5.6 Component overview for the access control mechanism ............ 50
5.7 Use case diagram for applying a newly detected service .......... 53
A.1 Class diagram for the identity management component ............ 68
A.2 Class diagram for the DHCP server component ................... 69
A.3 Class diagram for the access control mechanism component ....... 70
A.4 Class diagram for the event Manager component ................ 71
## List of Tables

3.1 Objectives provided by services. .............................................. 19
3.2 Services provided by mechanisms. ............................................. 20
4.1 Basic routing table for a device with IP 192.168.0.2. .................. 33
4.2 Availability of technologies for different operating systems. ............. 39
B.1 Old to new device state and lease type. ................................... 73
Chapter 1

Introduction

1.1 Motivation

One of the trends in Information technology topics is the concept of Internet of Things. IoT, the abbreviation for Internet of Things, is often discussed in lectures and papers. This concept refers to smart objects, i.e., everyday physical things enhanced by a small electronic device to provide local intelligence and connectivity to the cyberspace established by the Internet[41]. This on itself is not a novelty, however, the scale at which manufacturers are making devices smart is. A smart home is one of the applications of this concept in our everyday life.

Smart homes are homes where technology control all kinds of aspects in and around the home. Examples of aspects involving the personal environment are the in-home climate, light control, water control. However, smart homes could also focus on personal health. In Sweden a demonstration project, SmartBo[28], was built to improve the quality of living for elderly. In this smart home devices and sensors control lighting, windows, doors, locks, water outlets, electrical power and stoves, as well as visual and tactile signaling devices, speech synthesizers, and Braille displays for the visually impaired. The main concerns when designing these systems are usability and functionality. Security is prioritized lower or ignored.

Generally people tend not to worry about the security of devices and primarily focus on the usability and functionality [13]. This means that to sell products companies also need to prioritize usability and functionality at the cost of security. The implications of the weakened security of smart objects are already noticed by the media [66][67][71]. Also in many popular hacking communities a huge interest in hacking smart devices is shown. Especially IP-cameras are a point of interest as many people open them up for the internet [35][36].

In a survey we conducted on CEBIT 2015, we asked a number of smart home developers on the state of security in their products. Unfortunately, not a single manufacturer could provide us with a satisfying answer. The most common reply given by manufacturers was that the environment, i.e., the network itself must provide security. The security they do provide is limited to password logins transported over a clear means of communication. This, however, can be seen by anyone on the network.

1.2 Problem Description

The flood of various kinds of devices can make the home very smart, but also very insecure with respect to threats for the network itself. The goal of this project is to develop one or multiple tools which can be used to improve network security within homes. Security, however, has different priorities for different users. In a fully secured network, the devices end up not connected. On the other hand, a fully connected network which allows everything is also not desired. Users often are aware of the functionality which they would like to gain from the network, hence the user can decide what behavior they trust even though it might introduce security leaks.
CHAPTER 1. INTRODUCTION

To tackle the problems we currently see in home networks we use the following question as a guideline while discussing the various mechanisms. How can we improve network security in current home networks with no integration to devices currently connected to the network while keeping the end user in control? In Section 1.4 we discuss the outline of this paper and various sub-questions we use to answer the research question. The product of the study is an overview of the available technologies which can be used to improve the security of the home network and a system design implementing several of these technologies.

1.2.1 Constraints

The resulting mechanisms and products need to be compatible with all products which can be found within the home network. The amount of different connected devices is tremendous. Unfortunately, all those devices have different support and the common amount of supported technologies may be very small. Also, for many devices it is difficult to actually provide them with new solutions. Nevertheless, the mechanisms should still be able to also support those devices. Hence, the first constraint is:

Devices may be configured to do something, however, software needed to execute the task should already be present.

An exception to this constraint exists for one device. That device is allowed to have a modified network stack and added software capabilities to provide more security. The device, however, must be a part of the infrastructure required for the network. The most logical choice for this device is the internet gateway/router which provides internet access to the network. This constraint is captured as:

For one network infrastructure device, preferably the internet gateway/router, the network stack and software may be modified.

The last requirements are based on the research problem which states that the user must stay in control of the level of security. Effectively, this means that the system requires a user interface with the option to configure the different mechanisms used. However, many users will be non-expert users, hence the configuration of the mechanisms should be user-friendly. The following constraints capture these requirements:

A user can control what level of security is desired in the network.
The user interface needs to be usable.

The company which supports this masters project, also has a requirement regarding the user interface. The solutions must be integrable with various other products which are develop for home home networks. To supply this functionality, an API is needed. The API needs to take usability into account. This leads to the last constraint:

Configuration needs to be possible through an API.

1.3 Relevant Work

In this study, we discuss various papers and studies relevant to the topic network security. These papers either provide a very abstract view of network security or only focus on specific security challenges. We did not find any studies which correspond to the problem discussed in this paper. There are, however, hardware and software solution for network security. All of these solutions can be found within enterprise networks.

An enterprise network is a network used by large companies. These companies are anxious for corporate espionage, hence the security of their network is very important. Many lessons can be learned from providing security within enterprise networks. Many of the mechanisms and technologies discussed within this study are based on solutions found within the enterprise networks. Examples of such technologies are central configuration management, and a more secure user and device identities.

Specialized hardware is required which support all kinds of management and security mechanisms not required for a network itself. Currently, the support is only found in devices that are ten
times more expensive than equipment typically found in home networks. All these extra mechanisms have a high configuration loads where even experts with years of experience may struggle with. Hence implementing the mechanisms directly in home networks will not achieve a usable solution within the home environment.

1.4 Outline

To provide an answer for the research problem and to design the end product we first need to solve various sub-problems. The first sub-problem is to define what security is. In Chapter 2 we will use the following questions as the guideline to solve this problem.

- What is network security?
- How can we compare network security in various situations?

In the requirements of the research problem, we stated that the solution may not require extra software integrated into the existing network. For one device, preferably the internet gateway/router, an exception to this constraint is allowed. Hence, we are allowed to modify the software on the router / gateway. Unfortunately controlling only a single device may not be sufficient to protect an entire network. Hence to improve security we need to leverage existing tools and technologies present at networked devices to secure the entire network. In Chapter 3 we will discuss the technologies and the security challenges found in home networks. Furthermore, we analyze the functionality and ease of use of current networks. The following questions will be answered:

- What is a home network?
- What technologies are available?
- What are the challenges for security?

In Chapter 4 we discuss various solutions to the identified challenges for security in home networks. We argue the various hardware and software requirements of the solutions and discuss how well the solutions meet the prerequisites. In this chapter, we will answer the questions:

- What are solutions to the security challenges?
- How / why do those solutions solve challenges in network security?
- What are the requirements of the solutions and how do these fit with the prerequisites?

We conclude the analysis of possibilities with a system design. Chapter 5 presents this system design. We analyze security, usability, and functionality the new network/system design.

- How do we incorporate the various solutions to challenges into one environment?
- What is the level of security, functionality and ease of use?

In chapter 6 we conclude this project with a discussion of our work. We will also compare the classic network setup with our proposed system. Lastly, we discuss the limitations and possible future work for this project.
Chapter 2

Network Security

We see various forms of security around us. Gates, prisons, alarms, cameras, insurance policies are just a few of the countless examples of methods to provide security. In this chapter, we want to focus on the methods available to secure networks. The central question discussed is: What is network security? We also discuss some metrics which can be used to compare various scenarios.

First, we discuss the definition of security in Section 2.1. In Section 2.2 we discuss how the definition of security can be applied to networks. Furthermore, we address the objectives of network security in this section. To verify the completeness of these objectives, we analyze typical attacks in Section 2.3. In Section 2.4 we conclude the attack analysis by discussing the threats and relating them to the objectives. Lastly, in Section 2.5 we define some metrics which can be used to measure the level of security.

2.1 Security

The word security originate from the Latin word sēcūritās which has its origin in sēcūrus. This translate to without (sē-) care(-cūrus). The discretionary provides multiple definitions which correspond to the Latin meaning. Below you will find a subset of the definitions presented in [9].

- Freedom from danger, risk, etc.; safety.
- Freedom from care, anxiety, or doubt; well-founded confidence.
- Freedom from ...
- Protection or precautions taken against escape; custody.
- Precautions taken to guard against crime, attack, sabotage, espionage, etc.
- Protection or precautions taken against ...

We can derive two key definitions from the above. The first definition is to have freedom of threats which may endanger a person or institution and his/hers/its environment. The second definition it to provide measures to protect a person, institution, or his/hers/its environment against threats. Often these definitions can be converted to an objective which requires protection such as personal safety or custody of criminals.

2.2 Network Security

In a local network, we want to protect the data transferred over the network and the resources connected to the network. When we apply the second definition of security to the network we get...
the definition: *Network security is to provide measures to protect the network and its environment against threats.*

To be able to protect against threats, we first need to identify what we would like to protect. There already exists many literature [16][23][45][48][64][68] in this field. These study all share the same main conclusion to what needs protection. We call these objectives of network security.

There are three key objectives in network security known as the Security Requirements Triad or CIA Triad[68], Figure 2.1. The first objective is *confidentiality*. Confidentiality assures that private or confidential information is not made available or disclosed to unauthorized individuals. Confidentiality also covers privacy. This means that confidentiality also 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. The second objective is *integrity*. Integrity assures that information and programs are changed only in a specified and authorized manner. Also, it assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system. The last objective of the CIA Triad is *availability*. Availability assures that systems work promptly and services are not denied to users.

Security, however, is not limited to those three objectives. Some researchers argue that additional objectives are needed. The most common extension includes *authenticity* and *accountability* [68]. This new model is referred to as CIA+ [64].

Authenticity provides confidence in the validity of a transmission, a message, or message originator. This means that to reach authenticity we need to verify that users are who they say they are. Also, we need to verify that each input arriving at the system came from a trusted source. Accountability is the requirement for actions of an entity to be traced uniquely to that entity. This supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, we must be able to trace a security breach to a responsible party.

### 2.3 Typical Attacks

In the previous section, we have discussed to objectives of network security. However to protect those objectives we also need to learn about attacks and breaches of those objectives. Two types of attacks are distinguished by Stallings in [68]: passive attacks and active attacks. A passive attack monitors the network in an attempt to gather useful data available on the network. It does not affect resources on the network. An active attack modifies data in the network to alter resources available on the network. A special form of active attacks exists which send legitimate requests to targets. We identify these hacks as non-aggressive active hacks. Detailed explanations presented below are based on [64] and [68].
CHAPTER 2. NETWORK SECURITY

2.3.1 Passive Attacks
The goal of a passive attack is to obtain information transmitted over the network. Typical tactics used to perform these types of attacks are eavesdropping and monitoring. The goal of such an attack is the release of message contents or traffic analysis. Traffic analysis can be used to observe patterns in (encrypted) communication. The patterns could provide information about what data is communicated.

If a passive attack is executed between two communicating peers the communication between them is perceived as normal. This makes it hard to detect passive attacks. There exists some mechanisms to protect against these attacks. To obfuscate what is transmitted encryption can be used. This, however, does not solve traffic analysis.

2.3.2 Active Attacks
The goal of an active attack is to modify the behavior or resources available on a network. To execute an active attack data stream on the network are altered or false data streams are created on the network. There are four categories of active attacks: masquerade, replay, modification of messages, and denial of service.

In a masquerade the attacker pretends to be someone else. Using the stolen identity other, previously not possible actions can be executed using the spoofed identity. Usually, the action involves other active attacks such as a replay attack.

Replay combines a passive attack with an active attack. In the first the network is monitored and captured. In the second step captured data, often a request, is transmitted. The goal of such an attack is to produce an unauthorized effect.

Attacks that modify the contents of a message are categorized as modification of messages attacks. In these attacks the attacker modifies a part of a message to alter a system. For example when a command authorizes a user to access a file, the attacker can modify the command to allow another user access to that file.

The last type of active attack is used to prevent the use of a resource or service available on the network. This is called a denial of service attack. Tactics for these kinds of attacks are overloading the resource of service, or to disabling the network.

Typically in active attacks the attacker modifies the data stream. Most of the times modification can be detected relatively easily. A masquerade attack, however, can make identification of an active attack much harder. Protection against active attacks is more challenging than passive attacks. This is primarily caused by the diversity of the attacks themselves. Each class of attacks misuses different properties of the network traffic, therefore solutions are much more varying compared to passive attacks.

2.3.3 Non-aggressive Active Attacks
Non-aggressive active attacks are attacks that send legitimate requests to the target. These attacks assist in identifying vulnerabilities and gather more information about the target. Entire tool sets are built to perform these attacks. Like active attacks, this kind of attacks are easy to detect. The attacker announces his presence to get information and often performs many similar requests to the network. The network services misused for these attacks are vital for the operations of the network. When these requests are ignored or blocked it could mean that the network does not work anymore. Therefore, protection against these attacks is difficult.

2.3.4 Attack Schemes
Attackers often do not use a single attack to reach their goal. Multiple attacks are combined to a complete attack scheme. There are two main schemes [22]: directed and undirected attacks. Direct attacks are targeted at a selected resource on a network. Undirected attacks often set a trap and do not attack a pre-selected target.
CHAPTER 2. NETWORK SECURITY

2.3.4.1 Directed Attack

A directed attack can best be compared with a crime. The first step is reconnaissance. In this step, information is acquired about the target and its environment. Passive attacks, usually, are used to learn about the target. Although, non-aggressive active attacks can be used to do that as well.

The following step is to perform the crime itself. Using the information gathered in the first step the target is attacked. In this step, primarily active attacks are used. As discussed earlier active attacks are easy to recognize, especially from logs.

The last step of a hack is to cover up the evidence. Just like in a carefully planned crime, evidence of a hack is destroyed. This could be done by cleaning logs or by obfuscating the data transferred between the attacker and the target.

2.3.4.2 Undirected Attack

In an undirected attack, the attacker sets a trap. These traps are often placed on websites or included in downloads. Examples of such traps are malware and spyware. Once a target steps in a trap, usually, the attack is automatically executed. Just like directed attacks the attacker hides his tracks.

2.4 Network Security Threats

In the previous sections, we have discussed security and the objectives of security for networks. We have also discussed how attackers perform attacks on networks and resources on the networks. In this section, we will use this knowledge to identify the threats that may threaten security on a network.

The first class of attacks we discussed is passive attacks. These attacks focus on acquiring data which was not destined for them. Leakage of data is a breach of confidentiality, hence passive attacks breach confidentiality.

The second class of attacks we discussed is active attacks. The first type of active attacks we discussed where masquerade attacks. The goal of these attacks is to pose as someone else. This harms the authenticity of the data and the accountability of the posed subject. The second form of active attack is the replay attack.

Replay attacks are used to resend data. This targets accountability as the original sender is not aware of the replay. Also authenticity is harmed. Another type of attack was the modification of the data. This directly breaches the integrity and authenticity of the data. The last type of active attacks were focused on making services unavailable to the users. The goal of these attacks is to breach confidentiality.

The last class of attacks we discussed is non-aggressive active attacks. These attacks perform many legitimate requests to find public information. This is not a direct breach of confidentiality however it does focus on gaining information. This information usually is not something you want to disclose to the attacker. Therefore, it is a breach of confidentiality.

2.5 Security Measures

Already many metrics exists to assess network security. Most of them are specified to specific applications or platforms such as access control. For access control, examples of such metrics are the number of policies that do not match with the requirements [73]. Metrics like these are very strong in identifying usage weaknesses. However, a major disadvantage is that collecting these metrics can be a very time-consuming process. It involves analysis and interpretation of the requirements of the policies and the policies them self.

Another measure which can be used is based on CVSS-Based Individual Value Assignments. To calculate this metric, devices and the network itself is reviewed for known exploits. Each usable
exploit is assigned a weight. The sum results in the security grade. Unfortunately, the combination of attacks is not included in this metric. Some attacks individually might not score very high, however, combined they can still be very harmful. Frigault et al included attack graphs to those metrics to also model the combination of attacks [32][31]. The attack graphs are graphs which describe various attack paths to breach a system. Using this method, also a combination of attacks which lead to a great security threat are included in the score. Acquiring this metric can be done automatic and with relative ease compared to the platform-specific metrics. Unfortunately, this metric misses configuration mistakes made by the user.

For our project, we focus on the wanted and unwanted behavior in networks. Unwanted behavior can be measured by the number of connection allowed which should be blocked and the number connections which are blocked and should be allowed. While wanted behavior can be measured by the number of connections which should by allowed and are allowed and vice versa. We will use various scenarios which discuss some allowed and some non-allowed behavior. For each scenario, we describe the wanted and unwanted behavior. Subsequently to the scenarios we examine the availability of mechanism which provide security. We discuss how well the mechanisms can perform within their environment.

2.5.1 Scenarios

We use various scenarios to analyze the performance of different network security setups. In this section, we provide an overview of the scenarios. The described situations are incomplete, however, we will ignore undefined behavior / connections. Each scenario consists of a device which has (no) access to certain devices, and can (not) be accessed from certain devices. When rules overlap, the more specific rule has priority over more general rules.

2.5.1.1 Scenario 1: Guest Devices

The first scenario explains guest devices on the network. In many situations you want to allow guests to connect to the network and have access to the internet for normal usage, however, the guest should not have access to the rest of the network. In this scenario, we formalize the behavior required for guest devices.

<table>
<thead>
<tr>
<th>Access to</th>
<th>Access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Everyday internet use (web, mail and chat services).</td>
<td>• The gateway.</td>
</tr>
<tr>
<td>• Required services on networks and the internet (DNS, DHCP).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No access to</th>
<th>No access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Other services on the internet and network.</td>
<td>• Internet devices.</td>
</tr>
<tr>
<td>• Local devices other than the gateway.</td>
<td>• Devices on the local network other than the gateway.</td>
</tr>
</tbody>
</table>
2.5.1.2 Scenario 2: No Internet

In this scenario, we define a device which only has access to resources in the local network. The device is not allowed to communicate with the internet. Also, the device should not be reachable from the internet.

<table>
<thead>
<tr>
<th>Access to</th>
<th>Access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The local network.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No access to</th>
<th>No access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The internet.</td>
<td>• The internet.</td>
</tr>
</tbody>
</table>

2.5.1.3 Scenario 3: Limited local network

In some cases, you want your computer to communicate to the internet, your printer and other printers, but not to your IP camera. In this scenario, we propose a setup which will do something like that.

<table>
<thead>
<tr>
<th>Access to</th>
<th>Access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Computers in the local network.</td>
<td></td>
</tr>
<tr>
<td>• Printers in the local network to print.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No access to</th>
<th>No access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Local IP cameras.</td>
<td></td>
</tr>
</tbody>
</table>

2.5.1.4 Scenario 4: No access from the internet

For most devices in your network, you want to limit the access devices outside your network have to your device. This scenario describes such a behavior.

<table>
<thead>
<tr>
<th>Access to</th>
<th>Access from</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>No access to</th>
<th>No access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The internet.</td>
<td></td>
</tr>
</tbody>
</table>
2.5.1.5 Scenario 5: a web service (http + https) is accessible from the internet

In some cases, you want to run a service which can be used from the internet. An example is a web server which serves a personal page. This scenario describes such a behavior.

<table>
<thead>
<tr>
<th>Access to</th>
<th>Access from</th>
</tr>
</thead>
<tbody>
<tr>
<td>The internet for HTTP (port 80) requests.</td>
<td>No access to</td>
</tr>
<tr>
<td>The internet for HTTPS (port 443) requests.</td>
<td>No access from</td>
</tr>
</tbody>
</table>

Securing the Home Network
Chapter 3

Home Network

We encounter networks every day. At companies you work, in homes at schools, they are everywhere. The devices connected to these networks may differ a lot. In this chapter, we want to explore the networks found in homes today. As more devices are getting added to networks, we make an educated guess of the networks of tomorrow. Section 3.1 we provide an overview of the hardware used in home networks as well as typical network setups.

Differed grades of hardware have different feature sets. In Section 3.2 we identify features and technologies which can be present in networks to provide security. As a guideline for answering this question, we will use the ITU X.800 recommendation [2]. The recommendation proposes various technologies and methods to provide security. Unfortunately, not all security measures which can be taken are used in home networks. We will also discuss which devices in the home network support the different technologies presented.

Even though technologies to secure the network are present in home networks that does not necessarily mean that they are used. Some technologies have extra restriction or challenges which make usage hard. In Section 3.3 we identify the challenges and restrictions for security in home networks.

Lastly, in Section 3.4, we address the security performance of home networks with current security features. We will use the scoring mechanisms presented in Section 2.5 as a base for the performance. The challenges and restrictions discussed in Section 3.3 will be taken into account for this performance measure.

3.1 Home Networks

A network connects devices which each other. To make these connections an infrastructure is used. In this section, we identify which kinds of devices are and will be present in home networks. Also, we investigate the infrastructure used to connect the devices. In the last subsection, we discuss the topology of current and future home networks.

3.1.1 Endpoints

A broad range of networked devices is available for consumers. We call these devices endpoints. The most typical endpoints are computers, laptops, tablets, and phones. However, other endpoints like smart TV are also common in the current homes. Roughly we can divide the endpoints into three categories. The distinction between these categories is fuzzy at the borders.

Work stations. The first category is work stations. Endpoints in this category are mainly used for working. Typically they are shared with more users, but may also be used by only one user.

These endpoints run a powerful operating system such as Windows or Mac OS. This category is mostly populated by computers and laptops. Very powerful tablets are also included in this
category. Likewise, weaker laptops may fall in the category, personal devices.

Some of these endpoints are mobile. This subset of endpoints is categorized as mobile workstations. Mobile endpoints may connect with various other networks.

**Personal devices.** Many endpoints, such as a phone and tablet, are for personal use. These devices are covered in the second class of devices, personal devices. Endpoints in this class are less powerful and often have smaller screens. Their operating system is adjusted for their platform and therefore also less powerful. Current examples of such operating systems are Android, iOS, and Windows Phone.

All personal devices are mobile. Similar to mobile workstations these devices may connect to other networks.

**Connected home appliances.** This category is mainly focused on future endpoints. The current technological trend is to connect more home appliances to the network. Examples of such devices are refrigerators and coffee machines. Already more common are network connected printers, storage devices, and television. These and similar endpoints are also classified in this group.

Computing power for these endpoints is similar to the personal devices. The operating system, however, is more dedicated towards the tasks of the appliance. Often a lightweight Linux variant is used to operate these devices.

### 3.1.2 Infrastructure

Next to endpoints we also need devices which provide the logic required for the network. Furthermore, we need a device to connect the network with the internet. These devices are called infrastructure devices.

Various methods exist to provide the connections between endpoints and infrastructure devices. We distinguish two types. Wired and wireless connections. In home networks often a combination of the two is found.

#### 3.1.2.1 Infrastructure Devices

Infrastructure devices provide the logic required for the network. The devices make sure that traffic is redirected over the correct paths towards its destination. We will explain the most common devices within the next paragraphs.

**Broadband modem.** A modem converts digital signals to analog signals and vice versa [10] [69]. For a broadband modem, the digital signal is the internet. This connection itself can only be connected to a single device. The ISP often incorporates a router with the modem to allow multiple endpoints to connect with the network.

**Router.** The router is a gateway between multiple networks [42] [69]. The router routes the traffic from one network to another. Routers found in homes often distinguish a Local Area Network (LAN) and the Wide Area Network (WAN). The LAN contains the local devices while the WAN allows you to connect to a wider network such as the internet. All endpoints within the LAN communicate with the WAN via the router.

Routers often provide features such as a firewall and Network Address Translation (NAT). These provide filtering and extra routing capabilities to the router. ISPs often add features to the router to provide extra services such as telephone or television. Also, other extra features can be present in a router, for example, a file server. This makes the infrastructure device crossover with the endpoint devices. Especially higher priced routers contain these extra functionalities.

The router itself has only one port for the LAN and one port for the WAN. To connect multiple devices with the router a switch, optionally an access point is added in the same device on the LAN side of the router.
Switch. A switch is the core of a LAN. The switch connects all the wired devices [69]. It has a mechanism to send data to the correct physical port of the switch. This way data is sent directly to the correct recipient rather than all connected devices.

There are different kinds of switches. Basic, unmanaged, switches only provide this switching technology. Smarter, managed, switches also provide technologies for authenticating connected devices and creating multiple networks.

Access point. An access point provides wireless communication with devices [69]. Due to the nature of wireless connectivity everybody can view the data transmitted. However access points include some authorization which can also be used to encrypt the wirelessly transferred data. This way the data itself is protected. Often an access point is combined with a router.

3.1.2.2 Wired Connections

Ethernet cable. The most classical form of wired network connections is through Ethernet cables. It allows communication up to 10 Gbit/s [3], but currently in homes we see speeds up to 1 Gbit/s. Standardization of speeds up to 400 Gbit/s has started [1], but still requires a few years to finish and get adopted.

Ethernet cables are currently the most stable and fastest technology found in home networks. All versions based on the IEEE 802.3 standard are backward compatible. This means that a newer version does not require new devices.

Fiber cable. In some enterprise solutions communication, especially within the network infrastructure, relies on fiber cables. Optic cables have a great throughput and are fairly cheap [5]. Unfortunately, today devices in homes do not support fiber cables. Hence adaptation of fiber cables in a home would mean the use of adapters and new cabling.

Existing cables. Other technologies focus on reusing existing wires to provide networking. Examples of this are DOCSIS on coax cables and Powerline on power lines. The speed of those technologies is much slower and also much less stable [14]. Currently, theoretical speeds up to 1 Gbit/s are possible. Almost no endpoint device supports access to those networks natively and requires an adapter.

3.1.2.3 Wireless Connections

Wireless communication uses a single major standard. This standard is commonly referred to as WIFI but published as IEEE 802.11. This standard specified possible speeds up to 6.9 Gbit/s [56]. Like the standard for Ethernet, every version is backward compatible.

Wireless communication does have some drawbacks. Much interference exists on the networks [37]. A large number of available wireless connections cause this interference. Also, other technologies use the same communication frequency to communicate. Especially in big cities it may be difficult to set up a reliable connection. Another drawback is that the signal strength is greatly weakened when passing through walls. This makes the connections slow and even more prone to interference.

3.1.3 Topology

In the current homes, we have multiple networks. For the computers, fixed phone, TV, security systems, etc. Most of these networks are comparable in setup but use different means to communicate. In Figure 3.1a we show a network topology which currently is present in most homes. Switches are not included.

Currently, we see a transit from the dedicated networks for individual services towards a single IP network, the computer network. In Europe already many providers provide IPTV and fixed
phone lines are replaced by Voice Over IP. In Figure 3.1b we see a similar setup as in Figure 3.1a. However in this case all the devices are connected using a single network.

The demonstrated networks are still fairly basic in setup. In real networks, much more devices are present. Especially with the current IoT trend which will connect many home appliances with the internet.

![Network topology](image)

**Figure 3.1: Network topology.**

### 3.2 Technologies for Security

As shown within the possible attacks on network security many aspects come into play when protecting the security objectives (Section 2.2). The ITU X.800 [2] recommendation has described various services required to cover all the aspects of network security. The services need to be available and used within networks to provide security. The recommendation also includes many mechanisms which can be used to provide those services. In this section, we will give an overview of the various services and mechanisms. Furthermore, we will provide a mapping between the security objectives and services. For all the mechanisms, we will also briefly discuss existing products which use those mechanisms.

Definitions discussed in this section originate from the standard [2] and RFC 4949 [63].

#### 3.2.1 Services

Five main services are proposed by the recommendation: authentication, access control, data confidentiality, data integrity, and non-repudiation. Most of these services require each other. For instance access control relies on identities which can be authentication with the authentication service.

#### 3.2.1.1 Authentication

Authentication services provide authentication of communicating peers and the origin of the data. There are two variants of this authentication services. Peer Entity Authentication and Data-Origin Authentication.

**Peer entity authentication.** Peers transfer much data within networks. Often the data is only destined for a specific peer. The **Peer Entity Authentication** service can be used to verify the identity of a peer $a$ to another peer $b$ during communication. This helps protecting against a masquerade attack performed by peer $a$. Also, this service could help against unauthorized replays of data on the network.
Data-origin authentication. For most data, the original source should be known. The Data-Origin Authentication service provides a service which exactly supports this. The service provides the corroboration of the source of the data. Contrary to Peer Entity Authentication, no active connection is required. This means that the origin of data received in the past can be validated.

3.2.1.2 Access Control

A network, like the home network, has many available resources and devices. The access control service provides a service which restricts access from a resource or device to another resource or device. Various types of access, such as read and write, to a resource or device, can be distinguished by the service. To use this service we need to have authentication of the resources and devices.

3.2.1.3 Data Confidentiality

Data confidentiality services provide the protection of data from unauthorized disclosure. ITU distinguished four services for data confidentiality. The first two services discriminate on the connection type. The service provided by these two services is very similar. The two other services provide more specific confidentiality.

Connectionless confidentiality. The connectionless confidentiality service provides protection for data transfer which uses connectionless protocols. In connectionless protocols, each data packet addressed and routed based on information of that packet [12]. Most notable examples of protocols that use connectionless communication are IP and UDP. These are the major protocols used in networks to communicate with other peers.

Connection confidentiality. Communication which uses a connection protocol makes a direct connection between two peers [12]. There are two methods for providing connection: without or with virtual circuits. The TCP protocol is an example which does not use virtual circuits. TCP mimics a direct connection while the network itself is not aware of that connection. This means that for the network a TCP connection is still connectionless and each packet can take another route. The ITU X.614 recommendation does use virtual circuits to build a connection based on the packet layer protocol described in the ITU X.25 recommendation. When communication uses this protocol, all packet will travel using the same route over the network. The Connection confidentiality service provides confidentiality for these types of connections.

Selective field confidentiality. Selective field confidentiality services provide confidentiality for one or more parts of each packet. This can be used when only selected information needs encryption while other information does not require encryption.

Traffic-Flow Confidentiality. Even though traffic may be (partly) encrypted, it is still possible to gain indirect knowledge about the traffic. This is achieved by analysis the traffic. Traffic-Flow Confidentiality services protect against traffic analysis.

3.2.1.4 Data Integrity

Data integrity services detect unauthorized changes to data. This includes intentional changes, accidental changes, destruction, and loss. Note that it is difficult to protect against unauthorized changes, hence the services focus on detection of changes. A proper response taken when change is detected, offers protections of data integrity.

Connection integrity with(out) recovery. The connection integrity service detects any modification, insertion, deletion or replay of any data within an entire data stream. Recovery can be added to this service to attempt to recover from changes.
Selective field connection integrity. This service is similar to the previous service, however here only selected fields of a packet are checked for a breach of integrity.

Connectionless integrity. The connectionless integrity service only detects modifications of individual data packets. A limited form of replay could also be added to this service. This service has fewer possibilities to detect errors because it has no knowledge and availability of the data stream.

Selective field connectionless integrity. Selective field connectionless integrity is similar to Connectionless integrity. However, in this case, integrity is only determined for specific fields in a packet.

3.2.1.5 Nonrepudiation
The Nonrepudiation service provides protection against false denial of involvement in communication. The recommendation distinguishes two types of Nonrepudiation. The first type proofs the origin of a message while the second type provides a proof of delivery.

Nonrepudiation with proof of origin. When this service is used, the recipient of data is provided with a proof of the origin of the data. The proof protects against falsely dining that the data was sent by that sender.

Nonrepudiation with proof of delivery. This service provides proof of delivery of the data. In this case, the proof can show that the recipient has received the data when the recipient falsely claims it has not received the data.

3.2.1.6 Completeness
It is hard to argue if the set of services is sufficient to provide complete security. Using the security objectives, we argue the completeness of the services. Table 3.1 presents an overview of the completeness.

Confidentiality. A couple of services which help in providing confidentiality are present. The first service is *access control*. This service protects against unauthorized access to resources. While data is in transit the *data confidentiality* service protects against leakage of data.

When the platforms at the endpoints also provide confidentiality, the set of services would suffice to protect confidentiality. If the devices do not provide confidentiality, malware could still retrieve the data at the endpoint.

Integrity. A breach of integrity is detected by the *data integrity* services. Te detection itself would still allow data to continue. Hence to protect against or overcome the breach also a reaction is mandatory. It is not necessarily the case that this action will directly repair the fixed data.

When used with an action the *data integrity* services provide enough methods to protect against a breach of integrity.

Availability. None of the proposed services will help with the availability of a resource. In fact, services such as access control are designed to limit the availability of resources for specific peers. Methods which improve availability are based on network design and hardware stability.

Authenticity. The *authentication* services provide a mechanisms to verify the identity of users. Using the *nonrepudiation* services we can verify that the source of data is the identified user. Together these services provide mechanisms to protect against a breach of the authenticity of the data.
 CHAPTER 3. HOME NETWORK

Table 3.1: Objectives provided by services.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Access control</th>
<th>Authentication</th>
<th>Service</th>
<th>Data confidentiality</th>
<th>Data integrity</th>
<th>Nonrepudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Authenticity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accountability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>

Accountability. The last security objective is accountability. Both Nonrepudiation services are needed to provide some form of accountability. Proof of origin proofs that the data is received from a particular peer while proof of delivery proofs that data actually arrived at the intended recipient. When confidentiality is working properly this covers all forms of communication in a secure local network. Otherwise, another party could also receive the data without accountability for receiving the data.

3.2.2 Mechanisms

Next to the services the X.800 recommendation also recommends various mechanisms to provide those services. Similar to the services the mechanisms rely on the existence of each other for better security. For each mechanism, we provide current solutions which provide that mechanism. We look at both home and enterprise solutions as enterprise networks are concerned with security for many years already. An overview of the mechanisms and the services they provide is shown in Table 3.2.

3.2.2.1 Encipherment

Encipherment can provide confidentiality of data and traffic flow. There can be reversible and irreversible encipherment. Reversible encipherment, encryption, uses either a single key (symmetric) or a private and public key (asymmetric). In the case encryption is symmetric, one key is used to encrypt and decrypt messages. When asymmetric encryption is used, one key encrypts data while the other key decrypts data. In irreversible encipherment data is encoded to a form which cannot be decoded back, this is often referred to as a hash.

There already exists a couple of methods which provide encryption in enterprise networks, but they are all based on the same principle. The first option is a VPN connection to create a separate network. In such a setup, all the computers connect to a VPN service and have encrypted connections between the endpoints and the VPN [72]. The VPN server itself requires powerful hardware as it needs to be able to handle all data that passes through the network.

A less aggressive form are point to point IPSec encrypted connections. This decentralized encryption to single connections [27]. It removes the need for a single powerful server. Also, IPSec provides implementations for many other security services [40]. Still a few challenges remain compared with the VPN setup.

The first challenge is key exchange. This is part of authentication which we will discuss later. The second challenge is configuration. All participating clients need to have support for IPSec. This is not always the case. Also when devices do support IPSec use must be enforced while unencrypted communication needs to be blocked.

Microsoft has centralized the configuration of IPSec in enterprise networks via their domain controllers. But this option is too user-unfriendly to run in home networks because the configuration is still very hard.
Table 3.2: Services provided by mechanisms.

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Encipherment</th>
<th>Digital signature</th>
<th>Access control</th>
<th>Data integrity</th>
<th>Authentication exchange</th>
<th>Traffic padding</th>
<th>Routing control</th>
<th>Notarization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Services provided by mechanisms.
CHAPTER 3. HOME NETWORK

Encipherment in home networks is less common. Wireless LAN and Powerline are technologies which encipher the communications passing through them. However, it is still possible to decrypt the data send within these networks is you are part of the network. Also, these connections are converted back to decrypted communication over Ethernet cable.

There exist protocol specific encipherment. Examples of this are HTTPS, which encrypts normal HTTP web traffic, and SFTP, which secures FTP traffic. These protocols rely on the SSL or TLS security protocols. Both these protocols use a public/private key encipherment.

Generally we see that for encipherment, keys need to be exchanged. This is often done as part of the authentication step. Hence encryption relies on the authentication mechanisms (Section 3.2.2.5).

3.2.2.2 Digital Signature

A digital signature mechanism is a special variant of encipherment. It consists of two procedures. The first procedure is to sign data using private data (a private key). This could be done by either a cryptographic check value of the data or by encipherment of the data. The second procedure uses public known information (a public key) of the signer to validate if the sender was, in fact, the sender. These mechanisms provide data integrity and repudiation (with proof of origin).

These mechanisms are mainly used in cases where the validity and origin of data have to be assured. An example of such a situation is a firmware update for a device. Using the digital signature of the file we can verify that the contents of the file is as intended by the original sender.

3.2.2.3 Access Control

An access control mechanisms controls the access various entities have with each other. Using established identities and a set of policies, it decides whether an action is allowed or not. Access control mechanisms can be applied at either end of a communications association and/or at any intermediate point.

One of the locations of access control controls access at the entrance of the network itself. Both for wireless and wired connections this can be done via IEEE 802.1X port authentication [8]. This protocol allows computers to authenticate themselves before getting access to the network. Authentication can be done via any of the three authentication exchange mechanisms discussed in Section 3.2.2.5. It requires a central authentication server and support by the network infrastructure.

The second location where access control is possible is the firewall in the internet gateway. This system is used to protect the network against threats from the internet [33]. The firewall uses a set of rules to which traffic is matched. These rules are matched against traffic by various filters such as the identity of the involved devices and the type of traffic.

The last location is access control at endpoints. This is achieved with a combination of a personal firewall, file access rights, and configuration. The policies work similar to the policies found in the central gateway. The configuration of these policies is very difficult. In enterprise environments, these configurations can be centralized via a centralized server. In home networks, such a server is missing and each client has to be configured manually.

An example of a centralized environment is a Microsoft domain environment. In this environment, all computers are registered to a central domain server. This server can push configurations towards the endpoints. It can also control the other security measures such as the gateway firewall. Furthermore, this system is the authentication server required for the IEEE 802.1 port authentication. This server still needs to be able to authorize itself towards the endpoints.

3.2.2.4 Data Integrity

Data integrity can be checked by creating a checksum of the data and compare the checksum between the sender and receiver. This checksum may be a hash of the data. Adding a sequence numbering to the packets can protect against disordering, losing, and replaying data. These
mechanisms are often included within the protocol or within the encipherment mechanisms and are part of the digital signature of the data.

3.2.2.5 Authentication Exchange

There are three main techniques for authentication. The first technique is with the use of authentication information such as passwords. The sender sends the password and the receiver verifies it. This technique is most commonly used in networks to authenticate for individual services and on websites. The receiver could also forward the details towards a trusted party which checks the validity of the authentication details.

The second technique is cryptography. This authentication technique is an addition to the other types and counters spoofing and identity theft. This technique can use encryption keys to authenticate peers using a challenge-response handshake. These encryption keys are pre-shared or retrieved via on a public/private key infrastructure. For the latter option, a trusted infrastructure is required.

The last technique is the use of characteristics and/or possessions (fingerprints) of the entity. Often this technique is combined with the other techniques to maintain a session. For websites, this is achieved by creating a cookie with a unique identifier. Some systems solely rely on fingerprints to authenticate peers. An example of such a system is the firewalls. The identity of peers is often based on the IP address. Only relying on the fingerprints allows other parties to spoof the fingerprint and to take over the identity. Hence, the fingerprint needs to be secret or other devices must not be able to spoof the fingerprint.

3.2.2.6 Traffic Padding

Traffic padding makes sure that every packet has an equal size, be appending extra bits to fill a packet. Often this is done in the first step of encryption. This protects against analyzing packet sizes to guess the content. Padding could provide some loss in throughput because it requires fixed packet sizes. The loss in flexibility of the size of packets has a great impact on throughput of a network [55].

3.2.2.7 Routing Control

A routing control mechanism controls the route packets take in the network. Routing control is especially useful when multiple routes to the same destination exist. Based on various properties of the data a specific route could be chosen. An example is data which needs better security. This data will be transferred using a more secure route, or using another strategy to avoid less secure connections.

In a typical home network setup, there often exists only one path from one device to another device in the home network. This means that no alternatives are available to route the traffic. Still routing control is useful in home networks. It can be used to provide access control. Using routing control the network can isolate different sub-networks such as the guest network and your own. In home networks, this option is available on various access points via a peer isolation setting.

3.2.2.8 Notarization

A notarization mechanism provides assurance about properties of data communicated between two parties. The assurance is provided by a third party trusted by the other parties. The third party holds the necessary information required to provide the required assurance. Notarization is used to sign and verify keys used for encryption and communication [75]. This is used broadly on the internet and in enterprise networks to proof the origin of data.
3.3 Challenges with Current Security

In the previous section, we have discussed various kinds of technologies which can be used to protect networks. However, not all can or are applied in home networks. In this section, we provide an overview of mechanisms which lack a proper implementation.

3.3.1 The Mechanisms in Home Networks

Several of the mechanisms required for network security are available in home networks. This, however, does not mean that they are actually used.

Encipherment. The first mechanism we discuss is encipherment. Support for encipherment needs to be provided by the endpoints. The network itself cannot provide this support. An overlay network such as a VPN network is required to provide reversible encipherment. Hence solving this limitation cannot directly be solved without changing the configuration settings of every device participating in the network.

The most difficult part for the configuration is the authentication step required for encipherment. To encrypt the communication, keys have to be exchanged in during authentication. These keys have to be unique between every pair of devices otherwise other devices could still view the data. An authentication service has to be used to securely share the keys.

Digital Signature. Use of digital signatures is implicit when asymmetric encipherment is used. However in plain communication digital signatures are not widely used. In some cases, it is used at the device to check the validity of specific files. To provide support within a normal network for all communication also an overlay network has to be used. This overlay network should either provide encryption or is based on an alternative protocol which signs all data in a packet.

Access Control. As discussed earlier in Section 3.2.2.3 many locations exists where access control can be maintained. Although the various locations have a different reach rules may often be checked and filtered in multiple locations in your network. Typically to set up a service which needs to be available to the internet, you have to allow it at three locations. The first is the firewall at the device itself, the second is the firewall located in the modem/gateway, and the last is the NAT table where the shared public address is translated to the address of the local computer.

Quite a lot of steps are required to define a rule. Together with the non-uniform complexity of the configuration tools at the different locations it is very hard to define new rules. As a result firewalls get turned off. Hence even though filtering support exists in many devices, access control on the network is hardly used. To solve this insecurity the user experience needs to be improved and simplified.

Next to usability, there also exists a technical challenge for access control. Filters are based on the IP-address. In current networks it is very easy to spoof an IP address, hence it is very plausible to bypass rules. Hence, we also have to overcome the weak identity used to identify devices.

Data Integrity. Data integrity mechanisms suffer similar challenges as the digital signature mechanisms. However, contrary to Digital signatures, this mechanisms is available within plain connections. The usefulness of such a mechanism is very limited within a plain connection. It is only based on a hash which can be recomputed by everyone, hence modification of data is still possible. It only helps to detect unintended modifications. Solving the problems requires a change of protocols or use of encryption protocols.

Authentication Exchange. There are three classic technologies to authenticate, and all three have their weaknesses. The first technique requires authentication details. Various methods are available and used to compromise the details. Alternatively authentication can be based on cryptography.
Most common situations for authentication with cryptography use public and private keys. The private key is a key only available to one party and is not shared with anybody. The paired public key is publicly available. A third party is required to verify the owner of the public key. This verification part is often done within a Public Key Infrastructure which has a trusted third party. There is no user-friendly method for enrollment into a PKI. Also, it is not available within home networks.

Finally, a mechanism based on public information or possession of other entities is available. This option is the least secure option as everyone can view these details and mimic them. Unfortunately, this is also the mechanism used in home networks to identify and authenticate devices.

Newer technologies do exist which combines various types of authentication, namely single time access codes [54]. For this technology, a user requests an access code which is only valid for one time and often is limited to a brief moment. To support this technology, the user must have a trusted medium to receive the single time password. Often a text message or mail is used as trusted medium to receive these codes, however also more secure options use specialized devices. This mechanism is a mix between authentication with authentication details (the access code) and cryptography (generation of the code).

The major problem within this mechanism is that changing in authentication method is very hard if not impossible. Every device which needs to authenticate them self with the changing authentication needs to support the new authentication. Often this is not the case.

Traffic Padding. Traffic padding is used to make encryption stronger. Typically, it is used together with encipherment to make the size of data not and factor for analysis. Using it in clear channels does not add security as data is send in clear, however due to optimizations it still can be used. The use of padding largely depends on the encipherment method. Still most protocols and methods do use padding to strengthen the encipherment. A real challenge within home networks for traffic padding is nonexistent.

3.3.2 General Challenges

When classifying the challenges we have identified three main classes of challenges. The first class is the missing root of trust. This ranges from a thirsted third party to an identifier. This root of trust plays an essential role in verifying identities and data. Due to the lack of such a party a public-private key infrastructure cannot be made on local networks. This infrastructure is required for negotiating encryption keys in most encryption protocols. Mechanisms relying on encipherment are also crippled by this flaw.

Another major class is the configuration of mechanisms such as access control and routing control. There are many options in the rules and policies for these systems. This has proven to be a challenge to configure as even specialists make mistakes [60]. Because of difficult configuration setups, users often leave these policies to their defaults which are not secure enough. Hence usability is a huge problem.

The last class of challenges is the missing support from protocols. Many mechanisms lack a proper implementation due to a missing element within the network protocols. A solution to this problem has already been developed and is called IPsec. IPsec provides mechanisms for encipherment, digital signatures, data integrity, and traffic padding. Unfortunately, it still requires configuration on the devices which currently is difficult. Also, IPsec requires some form of authentication which also provides some configuration and trust challenges.

3.4 Security Performance for the Home Network

In Section 2.5 we have discussed a method to measure the security performance for different network setups. In this section, we will use this method to analyze the security performance of current home networks regarding the identified problems. First we will discuss the mechanisms followed by the scenarios.
We have discussed the various mechanisms in Section 3.3.1. We have discussed the availability of the various mechanisms. We have identified a problem with authentication of devices which is based on a weak authentication scheme. Also, we have identified that access control mechanisms such as firewalls are often turned off because of usability problems. These two elements form the core of security. Without access control, anyone can just access the data and without proper identities anyone can pose as somebody who has access. Therefore, the rest of the mechanisms do not provide security.

To analyze the scenarios we first have to provide an overview with the default behavior of the network and the devices. We assume a classic, single subnet, home network with a shared public IP address. The firewall in the gateway/modem disabled, and firewalls in endpoint devices left to default or default action. Note that outgoing access (and their replies) are always allowed for all devices. Access from the internet is allowed by most devices, however, the gateway does not have Network Address Translating rules, which means requests to the public IP address will not be transferred to a local address. We formalize these settings in the next overview.

**Windows endpoints:**

**Access to**

- Network and the internet.

**Access from**

- Network; warning with default action accept when traffic is encountered for the first time.
- Internet; warning with default action accept when traffic is encountered for the first time, but no NAT which makes entry unlikely.

**No access to**

- Internet; No NAT rules in GW, but can be bypassed.

**No access from**

**nix endpoints (Linux, Mac OS X, Unix):**

**Access to**

- Network and the internet.

**Access from**

- Network
- Internet; No NAT which makes entry unlikely.

**No access to**

- Internet; No NAT rules in the gateway, but can be bypassed.
CHAPTER 3. HOME NETWORK

Gateway / modem:

**Access to**
- Network and the internet.

**Access from**
- Network
- Internet; limited to specific services.

**No access to**
- Internet; Other services.

**No access from**

Scenario 1: To achieve scenario 1 we need to be able to identify guests. For wireless, this can be achieved using a special guest network. For wired computers, this is more difficult, however, most guests will use the wireless connection in current networks. Once we have identified guests we have to isolate them. Wireless access points often have the option to isolate clients. We could also apply a firewall rule in the access point to block communication. For wired devices, this is much more challenging as there may be paths where filtering is only possible between the endpoint devices. Last we need to modify the firewall rules to block the device from reaching other services. This setting has to be made within the firewall of the gateway. This is a difficult step as normal gateways have no options to create firewall rules for groups of devices. This scenario is plausible in the right conditions but requires complex configuration to block certain internet services.

Scenario 2: Blocking internet for specific devices is possible using the firewall inside the gateway. Hence, this scenario is possible when the gateway firewall is used.

Scenario 3: To achieve this behavior we have to block access from the computer to the IP camera. This rule cannot be enforced by the computer because it could simply disable it. Also, the main gateway cannot necessarily filter this behavior because there may exist a path on the network which does not pass the firewall of the gateway. Hence filtering is only possible when the IP camera supports filtering of devices. Most cameras actually do have an interface for (dis-)allowing certain devices, hence it is possible in most cases.

Scenario 4: This scenario describes no access from the internet to devices. The default behavior might be interpreted as already matching this scenario, but only because no routes exist. Unfortunately, technical capabilities are available to create the routes (temporarily), hence it requires active blocking. This behavior can be blocked at the central gateway and at the device in question, using their respective firewalls. Hence when the user decides to change the non-default behavior of either of those this scenario can be achieved.

Scenario 5: In this scenario the inverse is required; a device needs to be accessible from the internet for a specific service. From the default behavior (disabled firewalls) this only requires setting up the NAT rule in the gateway. However when a firewall does block it, which might be the case in a windows environment, the firewall has to be instructed to allow the traffic.

Generally we see that most behavior can already be set within the security possibilities of available in the home network. Most scenarios, however, do require changes in the difficult-to-configure firewalls present in the network. For scenarios 1 and 2 we do have requirements (isolation options and an enabled gateway firewall) on the network setup to achieve this behavior. Even though the firewall is typically present there might be a situation where it is disabled or not present. Similar the isolation option is only available for wireless connections but for not for wired guest devices. Hence overall network security can be decent en terms of possibilities to set the policies, but practically it is not achieved due to configuration difficulties.
Chapter 4

Solutions

In the previous chapter, we have provided an overview of mechanisms which are needed to provide security. For most mechanisms, some sort of support is already available, however due to some challenges and constraints, not all the methods where applicable. Especially for authentication and access control, the core for security systems, the current technologies cannot provide sufficient protection. In this chapter, we discuss various technical solutions to tackle the challenges we currently see in the core of network security.

4.1 Missing Root of Trust

One of the first challenges discussed in Section 3.3 is the missing root of trust. This is especially a challenge for authentication, since most other mechanisms require authentication. In this section, we give an overview of solutions to problems of current technologies. We focus on authentication and identification of devices, however, some of the proposed solutions can also be used for other purposes.

For authentication, we have three classic technologies and one newer technology for authenticating. Each of these technologies had their own challenge which is based on the missing root of trust. For authentication using authentication details, the problem is unwanted disclosure of the details. For cryptography, a missing trusted third party is the major challenge. Spoofing of entities is the biggest drawback for possession of entities. And for single time passwords, the trustworthiness of the receiving medium is an issue. All of these problems require a different approach in solving.

4.1.1 Authentication Details

Authentication with authentication details generally is used to authenticate a user to a service. Within the scope of this project, it is applicable when users need to login onto devices to configure them. Usually, a web interface is used for the user interface for setting configuration options. Hence solution which solve the problem needs to take the limitations of this interface into account.

4.1.1.1 Solution 1: Stronger policies

The first option to protect authentication details is to make them harder to retrieve. This can be achieved with setting and enforcing password policies. Password policies define everything about passwords. Adequate password policies regulate how the password looks like. Also, policies dictate how they are stored, transferred, and otherwise handled. Policies, however, do need to be enforced otherwise they are still useless.

Basic policies should at least require a password with sufficient entropy to withstand brute force attacks. To limit the possibilities of sniffing passwords storage and transport of passwords
need to be encrypted. Furthermore, policies should define an expiration date on passwords to enforce frequent changing of passwords.

### 4.1.1.2 Solution 2: Time-based One-time Passwords

An extreme form of password expiration is time-based one-time login tokens. There are various methods to deploy this technique. It is possible for a service to generate passwords (or login links) on request by the user. The password is then sent to the user using a user trusted environment such as the email address or phone. Also, it is possible to use a time-based one-time passwords algorithm [54]. The main concern for this method is the trustworthiness of the device which receives the access token. We discuss this option in Section 4.1.4.

### 4.1.1.3 Solution 3: Prevent sniffing

In the first solution, we have argued that we can protect against sniffing by using encryption. However, it is also possible to block sniffing techniques.

In order to sniff data in home networks, the data needs to pass the sniffing device. To put your device in between the two communicating devices routing of packets needs to be altered. Routing within local networks is based on the MAC address of devices. To alter the routing we have two options.

The first option is to change the destination MAC address of a packet. This can be done by poisoning the ARP tables on a device using special messages. The ARP table converts an IP address into the MAC address. To prevent this type of poisoning the ARP tables can be made static.

Another method to alter the routing is via CAM spoofing. CAM tables hold the mapping information which maps traffic towards a certain MAC address to a physical port on the switch [70]. To solve this problem, we also need to make the CAM tables static. This often is achieved using port security. Port security is available in most enterprise switches and is based on 802.1x [8]. Switches for domestic use do not have this feature.

### 4.1.1.4 Solution 4: Proof without sending the credentials

A fourth option to protect disclosure of the authentication details is to ensure that they are not sent over the network. Rather proof of knowledge of the credentials is used to authenticate each other. This is done using the challenge handshake authentication protocol [65]. Using this protocol First A provides B with a challenge X. B hashes X with the secret credentials and sends it back to A. A also computes the hash and verifies the result of B. This authenticates the identity of B to A. The procedure can be done again to authenticate the identity of A to B. A similar solution is actually used for cryptography.

### 4.1.2 Cryptography

Cryptography is used in many different authentication scenarios. The usages are very wide, therefore, many protocols have been standardized. The standards discuss various mechanisms which should provide a root of trust.

#### 4.1.2.1 Solution 1: PKI

The most popular root of trust is a Public Key Infrastructure (PKI) [25]. This infrastructure uses a certificate authority (CA) as trusted third party [34]. The CA signs the public key and the identifier of the party with its own key [57]. Now when the party shares its signed key as his identity, people can verify this claim by checking the signature. However, there are some challenges and flaws within this standard which makes deployment within home networks hard.
Reachability. The first challenge is the reachability of endpoints from the internet. The CA/Browser forum is a group of companies which set requirements for the signing process of CA. The purpose of this forum is to maintain the trustworthiness of the signatures. One of the requirements set by the CA/Browser forum is that the identifier of the party must be publicly reachable [7]. This can either be a name or an IP address itself. Unfortunately in home networks both the hostname and IP are private. To bypass this limitation, we can make the hostname of the clients publicly accessible through a reverse proxy. However, a superior solution is to use IPv6 as this solution introduces a publicly reachable IP address to every device.

Identification. The second challenge is that the identity attached to the public key needs to be verified. Currently, this is a manual process. Depending on the verification level, the process varies from using verification email to performing a full identity audit in person. Once the identity is validated the signing process itself is initiated which consists of manually copying and installing various keys from and to the CA. Even though it is hard to automate the process it can be done.

In Windows-based enterprise networks, the network controller also functions as a CA. During the (manual) registration of clients to the network, the controller is registered as trusted CA on the client. Also, a public key together with the clients hostname is signed by the controller. This setup even removes the requirement for public reachable identifiers. In home networks, this setup is infeasible because there is no secure method of registering.

An effort to automate the registration process of clients to CA is made by the Internet Security Research Group. This group is developing a CA called Let’s Encrypt [11]. The registration procedure is completely automated. The biggest problem for classical setups is proof of ownership of the device. Let’s Encrypt solves this by setting up a web server which serves a unique file. This, however, does require a publicly reachable host.

Trustworthiness. The last flaw is in the technology itself. It relies on a predefined set of root certificates which are the trusted parties [29]. History has proven that some of these parties cannot be trusted [17][46]. Fraudulent certificates have been signed by trusted CAs, which endangers the level of security. The cause of this flaw is the human presence and the oversimplification of the registration process. When the strict regulations set by the CA/Browser forum are followed, this flaw should not happen.

4.1.2.2 Solution 2: Trusted communication

The second solution is to put the trust in communication. If the communication channel can be trusted we can trust the validity of the data. The easiest method for trusted communication is to physically transfer the data using a storage media, like a USB thumb drive or a CD. Also, proximity wireless technologies can be used but they are less secure.

One-time Passwords require this solution in order to provide the root of trust. When the communication of the token is compromised, other people can acquire the token. A simple solution is to provide the token visually on the device which generates the token. In this case, the communication is based on a display.

4.1.3 Possession of Entities

In home networks, possession on entities is the most common method to identify itself. Unfortunately, the entities used are easily “spoofable”. When we cannot change the type of identification we need to make it harder to spoof the identity. The first solution we propose tries to achieve this.

4.1.3.1 Solution 1: Advanced fingerprints

Currently identity is based on MAC addresses we can improve this identity by including more details about the device itself. This can consist of properties such as the operating system version. This, however, does not add extra protection as these properties can still be spoofed easily.
A less spoofable method is to analyze typical traffic behavior of a device. This includes its connection behavior, like response times towards requests. When a device behaves different from the profile, it can be viewed as a different device. Next to the behavior of a device it is also possible to profile the connection itself. This profile can contain the network path, signal strength, and latency. Unfortunately, the more advanced profiles cannot be interchanged easily. Therefore, these profiles have a better use in detection of spoofing rather than to be used for identification.

4.1.3.2 Solution 2: Detect & Evade

In the previous solution, we argued that it is possible to improve the fingerprints of devices towards a level where spoofing becomes much harder. In reality, these profiles cannot be used because of missing support from devices. We are able to use the improved fingerprints to detect spoofing. In the detection of spoofing, we rename these advanced fingerprints to (device) profiles.

To detect spoofing, we use an older profile of a device and a current profile. When the two profiles differ too much it can be assumed that a device is being spoofed. Once spoofing is detected we need to take action in order to prevent misuses of the basic fingerprint. There are various options for the actions ranging from friendly to aggressive.

Notify user. The friendliest action is to inform a user or administrator about the spoofing attack. This does not protect the basic fingerprint / identity of a device. A real action has to be taken by the user to protect the identity of that device. This action has no consequences for devices connected to the network, but it also does not add extra security.

Block access via access control. A strong measure against IP spoofing is to block access to services from the spoofed IP. Locally this can be done fairly easy. To also block traffic on other devices a central access control system is needed. The access control system can block requests from the compromised identity. The spoofed device is required to obtain a different IP address in order to still connect with the network.

Cripple fingerprint. Stronger attacks also spoof the mac address of the basic fingerprint. When this is detected it might not be sufficient to block the IP address. In this case, it is also possible to also perform the spoof attack. This cripples the attack, but also cripples the spoofed device. This method does not guarantee blocking the attack, but it makes it very unreliable. It is for certain that the spoofed device is not able to connect to the network normally. Less intrusive is to prevent MAC spoofing.

4.1.3.3 Solution 3: Prevent

There is no solution which prevent an attacker to try to spoof data. However, there are solutions which can block such behavior at the gate. This is done with the aid of port authentication [8]. Port authentication requires a specialized network infrastructure. This infrastructure is able to block other than the original registered MAC and IP address on a single physical connection. Unfortunately, this type of hardware is currently out of the reach of the consumer market.

4.1.4 One-time Passwords

The last authentication technology is one-time passwords. There are different flavors of this technology. They all, however, have the same challenges: trustworthiness of the devices receiving and generating the access tokens, and the communication between the devices. To bypass the trustworthiness of communication, we can use special algorithms which are based on a pre-shared key and time to generate the access tokens. Now the trustworthiness of the access token depends on the algorithm used to generate the tokens and the device generating them.

To further improve remove the trustworthiness of the device special devices can be used which can only generate these codes, however, this might not be very usable. Alternatively application
generating and showing the access codes can run within a secure platform on mobile devices. An example of such an application is Google Authenticator\(^1\)

### 4.1.5 Applying Authentication to the Home Network

As a first step to improving the home network we need to better protect the identities of devices. Since these identities are based on the possession of entities we need to improve this system. The most fitting solution is Solution 2. This solution does not require modification of identities within the network, hence we do not have to alter the software on devices. Furthermore, we can provide control to the security level to the user by allowing different kinds of actions. For some actions, we do need some support to configure access control mechanisms. In Section 4.2 we discuss a mechanism which allows remote configuration. Solution 1 and 3 do not fit the requirements because they either require a modification at the software of all clients, or a modification of the network infrastructure.

### 4.2 Configuration

The second major challenge discussed in Section 3.3 is the inability to configure security settings within home networks. The main cause is the user friendliness of the interfaces. In this section, we briefly discuss user-friendly design. In the remaining part of the section, the focus is on the configuration of access control in home networks. We examine the technologies, policies and deployment models available to provide access control. Using the constraints stated in the problem description (Section 1.2.1) and the usability lessons discussed in Section 4.2.1, we propose the best model for deployment to improve access control in networks.

#### 4.2.1 Usability

Ease of use of products is very important. One of the main reasons that the configuration of access control systems is set incorrectly is the usability. Using these products is too complex. Therefore, it is simply ignored or turned off [59].

Making a product usable is challenging, especially for security products. Making the product very simple to use often limits the functionality thus methods to describe (dis)allowed behavior. Since the functionality of a security product is security it is very important that a correct balance is found between functionality and ease of use. Hence, functionality, and security need to be considered simultaneously [60, 74].

In an effort to create a usable product, we analyzed various studies on usability within access control. From those studies, we gained a list of requirements and lessons.

**Requirements:**

- The mental and physical load of the security application has to be tolerable [38].
- The user must understand the meaning of actions [38] and the representation of the actions need to match with the users concepts [18] and be consistent [61].
- The system needs to provide feedback on the chosen configuration which is understandable by the user [74][18][38][61].
- The system should only present options which are relevant for the user. Tasks which can be achieved by the system should be automated [38]. The goal is to make the secure path the easiest path for the user [74].

---

\(^1\)https://support.google.com/accounts/answer/1066447?hl=en&rd=1
CHAPTER 4. SOLUTIONS

• Identical objects should be represented the same while different object should be distinct [74][61]. The representation of the objects should match the user’s conceptual model of those objects [18].

• An overview needs to be available which allows to review and modification policies [74][61].

• The configuration should provide sufficient expressiveness to design policies without limitations to the user [74] while the user is still able to make a correct configuration or decision [38].

• The user needs to be able to group objects [61].

Lessons:

• You can’t retrofit security or usability. Both have to be integrated from the start [18][15].

• Many mechanisms exist which provide security, but they are not the solution. Higher level building blocks are needed when constructing secure applications [15].

• Put users’ needs first. They use the product, not the developers [15].

• The interface of the security application needs to be present locally. This can significantly simplify the user interaction [15].

4.2.2 Restricting Access

Various levels of separation exist, first we can separate devices and networks from each other. This completely blocks access between the devices. Using routing technologies we can “repair” a selected set of the connection, however in most cases we want more specific control. Another mechanism focuses on individual communication paths between services. This technology allows you to block and allow different services between the same pair of devices. In the next sections, we discuss the various mechanisms required for restricting access within home networks. The first mechanism we discuss is to separate the networks.

4.2.2.1 Separating networks

Virtually separating networks is an interesting method to use a single infrastructure for multiple networks. There are different techniques which lead to a different better separation of the networks, but this is always at the cost of flexibility.

Routing control. An important mechanism in networks is routing tables [44]. Without these tables, devices cannot communicate within networks. The tables allow devices to identify the next destination of a packet based on the target IP address of the packet. This is done according to various rules present in the routing tables.

A rule consists of a metric, a target (subnet), the next hop (router), and the interface which is used to communicate with. In home networks two, rules are present. In Table 4.1 we show these rules and and extra rule. The first rule is the default action, this rule provides a next hop which may have a route for the required packet. The second rule describes the local network. The On-link flag at the gateway field tells the routing mechanism that devices within that subnet can be found in the local network. The third rule is a null route. In this case, no next hop is known for traffic towards that device and thus is dropped.

Rules are prioritized based on fit. Hence, a rule only specifying a specific IP is prioritized over a rule describing an entire set of IP addresses. When there are two rules with the same fit, the metric is used to select the route.

Using routing control mechanisms, we can influence the contents of the routing tables. By creating null routes, we can disable communication between two devices. However to create two
Table 4.1: Basic routing table for a device with IP 192.168.0.2.

<table>
<thead>
<tr>
<th>Target</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
<td>10</td>
</tr>
<tr>
<td>192.168.0.0/24</td>
<td>On-link</td>
<td>192.168.0.2</td>
<td>20</td>
</tr>
<tr>
<td>192.168.0.4</td>
<td>null</td>
<td>null</td>
<td>20</td>
</tr>
</tbody>
</table>

networks out of one classic network (using one subnet) we have to instrument all devices to remove routes to devices from the other network. To solve this, we need some sort of grouping.

**Subnetting.** Creating various subnets is a great tool to group devices together. Using the default routes present in networks, devices within the same subnet can communicate. Devices within the different subnets can only communicate when the gateways of bots subnets now the route to the other subnet. Figure 4.1 shows a small example of a virtually separated network. The dashed lines show the real infrastructure and the solid lines show the virtual network.

![Virtual separated networks through subnets.](image)

Unfortunately, bypassing this separation is still possible. Using static routing rules, PC A could create a route which states that PC B is available on the local link, which actually is the case. Hence, PC A could still send targeted commands to PC B if it desires to do so. To block this possibilities and technology called VLAN can be used.

**VLAN.** VLAN, short for Virtual Local Area Network, is a technology standard [6] that let you use a single infrastructure for multiple networks. It does this by tagging packets with network identifiers. Usually tagging happens as soon as the packet enters the infrastructure. In Figure 4.1 this would be the switch. The tag attached to the packets is based on the port it arrives from. Hence even when a device mimics to be someone else it cannot enter the other network.

To successfully use this technology, the router and the switch both have to support VLAN tagging. Unfortunately, this technology is hardly found in devices targeted for home use. Hence,
we cannot rely on the availability of this technology within the network.

### 4.2.2.2 Firewall

It is not always the case that we want to block communication between devices entirely. Often some services are allowed. To achieve this behavior we can use firewalls. A firewall can be a very powerful tool when proper filter options are used. Over the years, various flavors of firewalls have been introduced. Each new flavor brought a new filtering concept to the filtering options. But before we discuss the different flavors, we first discuss NAT, an option often bundled with firewalls.

**NAT.** NAT stands for Network Address Translation which exactly describes its function. Using NAT network destined for a specific address (IP and port number) can be forwarded to another address. The major usage of this technology is to share a public IP among multiple private IP addresses. Forwarding itself, however, is not sufficient, the reply to requests will be targeted at the NAT, hence it also needs to redirect the replies.

In order to handle replies and incoming connections, a NAT table is used. The rules inside the table describe what data needs to be forwarded. This is based on the IP addresses of the two communicating devices and the used port numbers on either side. Wildcards on the public IP side are allowed for incoming connections.

To maintain this table the NAT server monitors all outgoing connections. As soon as a new connection is identified it is added to the NAT table. Also, static entries can be added to the table. These static entries allow remote parties to initiate a connection with a local computer. Especially due to this feature NAT is often confused with firewalls.

**Packet Filters Firewalls.** The most simple and first generation firewall is a packet filter firewall. This firewall looks at individual packets and based on static rules allows or disallows the traffic. The filter rules consists of: [4]

- the physical network interface that the packet arrives on;
- the address the data is (supposedly) coming from (source IP address);
- the address the data is going to (destination IP address);
- the type of transport layer (TCP, UDP, ICMP);
- the transport layer source port;
- the transport layer destination port.

Next to the filters options the rules also specify an action, deny or allow. The action taken for traffic depends on the first applicable rule. If no rule is applicable a default ruling is made. For the most secure settings, this should be deny, however in many routers with embedded firewalls the policy is to allow traffic [4].

Packet filters generally do not understand protocols. However, they are still suitable to deny access to certain services. By blocking or allowing the communication ports used for a service packet filters control access to the services.

**Circuit-Level Firewalls.** The second generation firewalls the focus is on connections. In order to pass the firewall, a valid connection is required. To check for valid connections a list of connections is maintained. The only supported connections are TCP connections.

To create a TCP connection a TCP handshake needs to be completed. After the handshake, the connection can be identified as **established**. While the handshake is not completed the state of the connection is **handshake**. If the connection is closing the state is changed to **closing**.

Data packets are not allowed to pass the firewall unless a connection is **established**. If a data packet arrives and a connection does not exist, it is dropped. When a connection does exist the packet is allowed through the firewall.
CHAPTER 4. SOLUTIONS

Application Layer Firewalls. The third type of firewalls verifies data at the application layer. In the application, much more information is present such as usernames and passwords. Filtering at this level is required knowledge about the structure of the protocol and data transferred with the protocol.

To support this knowledge for all possible services and applications is impossible. Hence, application-specific firewalls are required. Often reverse proxies are made which support filtering on application specific parameters.

The reverse proxy services relay requests from the clients to the correct application. Before it relays the requests it goes through the filtering rules for that protocol and/or application. The response to the request also passes the reverse proxy, hence also this information flow can be filtered by the reverse proxy.

Dynamic Packet Filters. The last type of firewalls is Dynamic Packet Filters. This type of firewalls allows to create dynamic packet filter rules. The filter options used in the rules is similar the first generation packet filtering firewalls. The difference is that rules change dynamically over time. Software in the application layer checks traffic for the need for extra rules [26]. This technology is used to support virtual connections between devices while a connectionless protocol, such as UDP is used.

Performance and Security. Performance is very important for firewalls. Firewalls may not slow down the network too much as this frustrates the user. Also, the firewall should provide a decent level of security to protect the same user.

Rules defined in the Application Layer Firewalls require a lot of information which is only available when a rule has passed the complete network stack. This requires a substantial amount of computational power. The rules for the Packet Filters Firewalls are very basic and can be checked within the first few layers of the network stack. The required processing time is much smaller than for the rules of the Application Layer Firewalls.

In Figure 4.2 an overview is given which compares performance and security. Normally packet filter firewalls outperform circuit-level firewalls, however when a packet filter has many rules it becomes slower compared to the circuit-level firewall. This is because the circuit-level firewall does not have rules which are checked. Also, the level of security is close because different concepts are targeted (service, connection state).

4.2.2.3 Deployment

Firewalls and other access control systems can only filter traffic passing through them. Therefore, the deployment location of the access control system is very important. In current home networks there are two locations where access control systems, such as a firewall, are located: at the
CHAPTER 4. SOLUTIONS

endpoint device and in the main gateway. Another location which is used in enterprise networks is at the edge of the infrastructure. For each of the basic locations, we provide an overview in how we can achieve a network in which all traffic can be filtered. We will respect the requirements (Section 1.2.1) of the product design as much as possible. According to the requirements, we are allowed to implement software at a single device, the main gateway. Various ideas for implementing the different network security technologies will require special support from the gateway normally not found on gateways. In the setup, we refer to this device as the Operator Network Unit (ONU).

Central Node. When relying on the central node to provide access control, we must ensure that all traffic passes that central node. This can be achieved by making an overlay network. In this overlay network, every endpoint is directly connected to a central node like the ONU. An overview of such a network can be seen in Figure 4.3. These direct connections are often encrypted, however, unencrypted solutions are also available. Communication on such a network would put a heavy workload on the central node.

To achieve such an overlay network we can use various options. The first option is to use VPN services to connect each endpoint with the ONU. A second option is to use the network separating mechanisms discussed in Section 4.2.2.1. The networks created with the separation consist of isolated devices. All these options have some drawbacks.

For VPN services, we need support on the devices. Also, rogue devices could simply just connect to the physical network and not to the overlay network. In this case, the endpoints need to actively block connections established on the physical network. VPN services do however provide a means to authenticate the different endpoints using credentials. For the Network separating mechanisms, we require hardware not found in homes or we are not able to completely block bypassing. However, in some cases it is possible to detect a bypass and take action.

Edge of network. Using inline security we provide security at the path between communicating devices. In networks, this can be achieved by providing security at every infrastructure device. A main advantage of providing security on every infrastructure device is that we can apply security at every entrance point of the network. A typical technique used to achieve access control at the entrance ports is port authentication.

Figure 4.3: Central access control.
Port authentication can be used to authenticate the device connected to one single entry port. A popular mechanism is EAP authentication which is standardized in 802.1x. As this authentication is performed at the infrastructure itself, it would require support of the infrastructure itself and the endpoints. Most endpoints do have support for this technology. However, depending on the support of the infrastructure, authentication could also be based on properties of a device. Support at the infrastructure level is very limited. Currently, only the hardware found in expensive enterprise networks have support for this protocol. On itself, this mechanism only protects access to the network, not within the network. Therefore, it needs to be combined with other technologies.

A second technology is to apply a firewall at every infrastructure device. This can be done concurrently with the previous technology or on its own. This technique also requires support of the infrastructure itself and generally is even more expensive than the previous setup. Also, because the execution of the access policies is decentralized, access policies need to be distributed. This requires other mechanisms, or it needs to be done manually.

Figure 4.4: Inline access control.

Endpoints. The third location we implement access control in every endpoint and main router. This situation is shown in Figure 4.5. In this situation, every device is responsible for its own access control. Access control deployed on the main router is to protect the network from the internet. Also, it is used to protect access to the main router itself from the network. This technique requires an access control mechanism on every endpoint in your network. Typically single-purpose network devices such as a doorbell or an IP phone only implement simple authentication. They do not limit the devices that can even try to do the authentication.

Similar to the distributed access control in inline access control, the policies are distributed. Also in this situation, we need to distribute the policies. Mechanisms providing the distribution are much harder to implement for this situation. This is because there are numerous of different applications available which implement access control.

Centrally Managed Endpoints. Alternatively to only picking one of the schemes, we can also combine the various schemes. The goal of combining the different schemes is to ensure that all
traffic on the network passes at least one firewall. We call this setup a centrally managed endpoints deployment. A similar setup can be found in enterprise networks, where the firewalls of various endpoints are controlled via a central node. However, in enterprise networks, devices which do not have a firewall (which can be configured from the central node) can still freely communicate. In this deployment scheme, this is solved by isolating the devices which we do not support. By isolating devices, traffic from and to the devices need to go through the central hub, the ONU. This allows the ONU to enforce the firewall policies. For the supported devices, the ONU needs a mechanism to remotely configure the firewalls.

4.2.2.4 Remote Management
Remote management of devices is technically not part of access control. Yet, this service is required in some of the deployment scenarios. There are various different access control systems which do support remote management of the system. Unfortunately, in most cases, this is only limited to their own software and is based on proprietary communication protocols. In general there is no direct support for remote management in the standard access control mechanisms present on devices. However, the mechanisms can be controlled over a command line, shell, which often is accessible remotely.

The most known method for remote shell access is through telnet. For simple network devices, this is the go-to mechanism for advanced control of those devices. The protocol does have a downside though. Communication is in plain text. This means that it is very easy to view all the data transferred. A more secure protocol is SSH. This protocol allows encryption, hence other devices can only inspect the encrypted data. Unfortunately, this protocol is not available within windows environments. Fortunately for windows other tools exists.

4.2.2.5 Availability
Table 4.2 shows an overview of the supported technologies on different platforms found in home networks. For each technology, we state the tool which is used in the specific operating system to provide the functionality. Most operating systems do provide routing control, static NAT rules and a form of a firewall. Within mobile operating systems, all technologies are not available without first hacking the device.
Table 4.2: Availability of technologies for different operating systems.

<table>
<thead>
<tr>
<th>OS / Platform</th>
<th>Routing Control</th>
<th>Static NAT</th>
<th>Firewall Rules</th>
<th>Remote Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>route [24]</td>
<td>PortProxy</td>
<td>Windows</td>
<td>WinRS [51],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Advanced</td>
<td>PowerShell [50],</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Firewall [53]</td>
<td>PsExec [62]</td>
</tr>
<tr>
<td>Windows Phone</td>
<td>route [24]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* Not always included.

For most workstation environments, we have a supported technique for all mechanisms. We do not have this support on personal devices which use mobile variants of the operating systems. The firewalls available on the platforms all support at least the functionality required for packet filter Firewalls. Hence for access control we limit us to the options of this firewall type. We do however have the ability to change the rules dynamically is we remotely instrument the firewalls.

We have also identified that the mobile platforms do not support all the technologies. To still allow individual filtering the devices have to be isolated. Subnetting is the best solution to achieve this as it does not make any assumptions on the devices. Using route control on the ONU, communication between isolated devices and the rest of the network is still allowed.

4.2.3 Policies

In the previous section, Section 4.2.2 we have discussed various technologies which can be used to restrict the access devices have in networks. The best combination of technologies which can be used to provide access control are firewalls for the actual access control combined with subnetting and routing control for isolation. We have already discussed what kind of information is required for the policies of different technologies. However from a users perspective this configuration is difficult. Hence, we need a method to answer the informational need for the policies with the information users can provide.

4.2.3.1 Information need

The different mechanisms have a different Informational need. Some of the informational needs can be answered using heuristics while others require more information from a user. We provide an overview of the informational need together with methods to supply the information in this section. In Section 4.2.3.2 we discuss the required user interaction to retrieve the information which needs to be provided by the user.

**Subnetting.** For subnetting, a mapping is needed which maps devices to the subnet they need to use. There are multiple methods to create the mapping. The first method is to distinct based on support. When devices do not support all required technologies, isolation of the devices may assist in protecting the devices and the network. Isolating the device can be done using a subnet which only contains that particular device and the gateway of the subnet.

A second method is to combine devices with a similar function together. Also, this can be done automatically. However, the drawback of such a grouping is that many local area networks will need to pass the gateway.

**Routing control.** Routing control requires a table with routing rules. This table is different for each device present on the network. The basic routing tables are derived from the subnets
CHAPTER 4. SOLUTIONS

assigned to devices. More specific rules can be added for devices which communicate a lot and are not within the same subnet. This will bypass security measures taken on the central gateway between the subnets. Also, endpoints need to have support to configure more specific routes. Scenarios, where such a rule would be necessary, exist when performance has a higher priority than security. This is a compromise which we do not offer as it both add extra complexity and degrades security.

**Firewalls.** The last technique which can secure the home network is the firewall. We have discussed various types of firewalls. In the practical implementation however we have only seen packet filter firewalls which are usable on many systems.

A packet filter rule consists of 5 elements: the source from where (and which ports) traffic originate; the destination to where (and which ports) does traffic goes; the protocol used to send the traffic (TCP, UDP, ICMP, other, all); the direction traffic going, inbound, outbound, or passing through; and the action taken (allow, block).

Next to the packet filter often NAT is available. NAT allows rewriting the destination of the traffic. This is useful when sharing a common IP with the internet. However, it could also be used to forward traffic to other destinations. The rules to identify traffic are similar to the filter options available for the firewall. The action, however, is forwarding data to a new destination. Hence, an extra field is required to support setting the destination for forwarding.

For the NAT and firewall rules the following general fields are required: Which devices are involved; the communication type used; and the action required for the communication happening between the involved parties. This information needs to be provided by the user.

**Remote Management.** All remote management options require a form or authentication to connect to the remote shell. Usually, this is achieved via a username and password. A more advanced setup may require the distribution of public keys. This information requires a registration step of the network devices to the central provider. During this registration step, we can also register more information such as a user-friendly name of the device and the location of a device.

4.2.3.2 Availability of information

People are unaware of the benefit of security until a security failure hits. At the same time users view security as an impediment to efficient and user-friendly operation [68]. Hence, security needs to be simple, straightforward and must demonstrate the benefit. This, however, does not explain what information on security users can provide.

Poole et al. performed a study [58] on the users view on networks and security on networks. They identified that usage is typically described in (descriptive) usage scenarios. An example of a participant describing a computer states 9-

yr-old girl bedroom desktop wired to network for printer access, no internet access (?), software block. This statement shows why a device is connected to the network while also discussing what is blocked and how. In this example, we also see one of the problems with the users view. First of all the statement does not cover all aspects. Access to other resources available on the network is not described. Furthermore, the requirement no internet access might be too strong. In most cases, people do want their virus scanners and operating system to receive updates.

The study showed that users can provide usage scenarios, however, they are not able act and only allow the required behavior. The access model the users have for devices often does not suffice. Also, the study shows that most allowed actions are descriptive. Remote printer access means that the user allows a device to communicate with the printer using the required printing protocol. The application of these findings can already be found in many port forwarding solutions and firewalls. In many cases, a profile such as network printing or web server can be used to identify allowed / forwarded data.

The study by Poole et al. also showed that identifying devices also is descriptive. Devices are often referred to using the device type (laptop, computer, television, etc.), the location (kitchen, living room), or the ownership of the device. This concept is already used in many operating
systems where the hostname is based on the device type and the owner. For Windows, the default hostname is \textit{<YourName>-Computer}. Also in firewall and port forwarding configuration it becomes more common to select devices based on their (descriptive) hostname.

Hence for the user policies are a descriptive language of allowed actions. References to objects involved and the actions itself are also descriptive rather than formal which is required by the machine. We refer to this description of allowed behavior as a scenario.

4.2.3.3 Translating the Information

To transform the usage scenarios provided by users to formal rules we need to know how the information is provided by the user. This is done using three steps. The first step identifies the devices for which a usage scenario is applicable. The second step identifies the usage scenario. The last step identifies other involved parties.

The first action users need to do is to select the local devices for which a usage scenario is set. This can either be from a custom selection of from (predefined) groups. Identities of the devices used in this representation are matched to the machine identities. These are based on the advanced fingerprint / profile of the device. The coupling is provided by initial registration.

After the selection of the target devices, a usage scenario is picked for the devices. The scenario describes the behavior allowed by a device. The scenario transfers to the formal filtering rules that define the communication and direction. Elements of the firewall / NAT rules stored in scenarios are: the ports (source and destination) used to communicate; a reference to which selection of devices is the source, and which selection of devices is the destination; what protocol is used; and last which direction traffic is going. Also, some flags are added.

One flag states if the rule can only be applied to a single device. This is useful for scenarios where the device becomes a public reachable server. Another flag is used to limit the possibilities for the second selection of involved parties.

The last step is to select other parties involved. This is done with a similar option set as the initial selection of devices, however now also the Internet is part of the possibilities. When only a single option is available (due to scenario constraints), it is automatically selected.

When the user has provided all the input the first and the last step are merged with the second step to create firewall rules. A second processing step analyzes the scenario to identify the need for NAT rules which make a device remotely accessible.

Feedback of the configuration is important. As shown in the example presented by Poole et al. people tend to underspecify the allowed behavior or the security measures. Using a feedback step with the resulting consequences of the specified behavior allows verification of the rules.

4.2.4 Applying Access Control to the Home Network

The most important and flexible technology required for access control is the firewall. Using this technology we can make the very specific rules necessary to block and allow specific services. As deployment scheme for the firewall, the centrally managed endpoints deployment needs to be used. Using this scheme we can guarantee at least one firewall between every possible path on the network. This, however, assumes that we can securely isolate devices.

The only feasible option to isolate devices which we cannot configure is through subnetting. Hence, we are bound to subnetting for isolation. To make subnetting more secure, we need to actively search for unregistered devices. Routing control is only used to link the subnets together. The firewall at the ONU provides the actual access control filtered in the case that isolated devices are involved.
Chapter 5

System Design

In Chapter 4, we have discussed various solutions for the challenges in home network security discussed in Chapter 3. In this chapter, we discuss a design of a system, the Operator Network Unit, which merges several of these solutions, as components, into one product which can be placed as main gateway inside the home network.

The solutions discussed in Chapter 4 have two major themes. The first theme is identity management. In Section 5.1 the design of the Identity Management component is discussed. The major tasks of this component consist of a list of known devices and protecting the identities of those devices.

The DHCP server provisions the identities of the devices in a network. In the proposed system, we also include a DHCP server. The design of this DHCP server is discussed in Section 5.2. The DHCP server has a special task in the system. Alongside provisioning the identities to devices, it also provides the isolation mechanism through subnetting. The latter option is required for access control in the network which is the second theme of the solutions discussed in Chapter 4.

For access control, we rely on the centrally managed distributed firewall. As a fallback for unsupported devices, we use the centrally deployed firewall scheme together with the isolation provided by the DHCP server. Section 5.3 discussed the design of the access control component for our system. The access control system maintains and delegate (or provides) enforcement of the scenarios and policies set for access control.

The last component is the event manager which is discussed in Section 5.4. This component is added to the system to initiate communications with other systems and client applications. Also, internally this system is used to create event-based triggers for the behavior of the system.

An overview of the components is presented in Figure 5.1. In Appendix A detailed class diagrams can be found for the various components.
5.1 Identity Management

The identity manager component has two main tasks. The first task is maintaining the list of known devices on the network. As a second task, it needs to identify spoofed and unknown devices. Both these tasks require a different approach. Figure 5.2 shows an overview of the components.

![Identity Management Diagram]

Figure 5.2: Component overview for the identity manager.

The first task is primarily an administrative task. It requires the creation and storage of device profiles. To this end, we have introduced the sub-component device profile.

5.1.1 Device Profile

An important part of device management is the profile of devices. The profile has various parts. The first part focuses on the classic identities of the device in the network. This part consists of three attributes. The MAC address, the hostname, and the IP address / lease. These attributes can be retrieved during the DHCP handshake when the device acquires its IP address. For the API access also a unique identifier is added as an identifier for the device.

For presentation to the user, extra fields are introduced which also describe the identity. The fields consist of a name, location, and description. The user can set the values of these fields upon registration of the device. Next to this information the user is also able to assign tags to a device. Tags can be used in for grouping or to further identify the device. When selecting devices based on tags, other properties and attributes are also added to the tag list.

Next to the extra fields which identify the device also authentication details can be provided to remotely execute commands. The required fields depend on the supported technologies which, in turn, are derived from the device profile. We combined the fields for authentication to the client in three attributes. A username and password for traditional login and an authentication string for the other situations. A flag identifies which remote command execution mechanism is used.

The last part of the profile is the actual profile. The profile consists of a set of services running
on the device, the response time of the device, and the device type. Other properties can be added
as a set of tags. The values of these parts are based on a scan performed by the device profiler.

5.1.2 Device Profiler

The device profiler is based on Nmap[47]. Using the command \texttt{NMAP -T5 -A -v <IP ADDRESS>}
we create a detailed profile of the device. By default, Nmap scans the most commonly used ports
and performs a ping test. The flag \texttt{-T5} tells Nmap to use many network resources to do testing.
This speeds up the test. The flag \texttt{-A} tells Nmap to identify the operating system and versions of
the software.

To process the results of Nmap we use a wrapper nmap4j[39]. This wrapper allows parsing of
the output of Nmap by java. Using the wrapper, we process the output to the required profile
fields. The unprocessed (XML) result of Nmap itself is also stored in the device profile.

The device profiler runs once every hour to update the profile of the device. When too much
change is noticed between the old and the new scan, the identity is regarded as spoofed. This
event is published using the event manager. If the difference is not too big the profile is updated
with the new profile details. An event is created for the updated profile and is pushed to the event
manager.

The device profiler also detects changes in available services on a device. This knowledge is
important for the access control component. That component can issue new scenarios on the
network based on the new service. Using a special event for new services, the detection of a new
service is communicated with the access control component and other subscribed parties.

The device profiler only scans the known devices in the network. Unfortunately, devices can
also enter the network without announcing themselves to the ONU. To identify these devices we
use a network profiler.

5.1.3 Network Profiler

The network profiler quickly scans the network for all devices existing on the network. It does
so by pinging every address available on the network. If a device replies to the request we verify
if we have a matching device within the registered set of devices. If not an unknown device is
encountered. This event is published using the event manager. Similar to the device profiler this
scan is executed using Nmap and the results processed with namp4j.

5.1.4 Device Groups

One of the usability requirements (Section 4.2.1) is the possibility to create groups for which we
want to specify policies. Hence, we need a mechanism which supports grouping of the devices.
We have two types of groups, manually created groups and dynamic groups. Manual groups are
groups of devices defined by the user. Dynamic groups are groups based on tags. The initial set of
standard groups consists of all local devices (no tags), all trusted devices, and all isolated devices.
However, also other groups can be added such as all smart TV’s if the user assigns them with such
a tag. In the future automatically assigned tags can be used to create more groups.

Groups, generally, can be assigned to scenarios. In this case, the policies attached to the
scenario will be enforced for all members of the group. When the group layout changes also
the access control scenario will change. In case an access control scenario changes, we have to
reevaluate the policies and push them to the clients.

5.1.5 Use cases and Usage Scenarios

Various use cases exists which require or result in user interaction. Generally user interaction is
invoked by a client app registered to the event notifier and handles the event. The client may also
choose to ignore the event. This, however, would block the device from the network in case of a
new device event.
5.1.5.1 Registration Process
The registration process starts as soon as a new device is provided with an IP address. During the assignment, the device has been identified as a new device and once the device received its address an event is published. This event requests the user to register the new device. Simultaneously the device profiler starts to profile the device. Figure 5.3 contains the use case diagram for this use case.

When the users visit the registration page it first enters the name, location and description of the device. The location can be chosen from existing locations or a new location can be used, this is used to limit the number of locations used by users. When the details are provided the setup waits for the device profiler to finish. Once the profiler has finished profiling the device the client can request which authentication details are required. This is based on the supported mechanisms identified by the device profiler. The user provides these authentication details or skips this part. Once the authentication details are accepted or skipped the registration process itself is finished. What follows is a process which allows the configuration of access control scenarios.

5.1.5.2 Spoofed Identity
When a spoofed identity is encountered by the device profiler, the user gets a notification of a security threat. Currently no actions are programmed and/or subscribed to the event notifier. In future work, however, this can be done.

5.1.5.3 New Service Discovered
Every once in a while a user installs or opens a new application which performs a service to the network. This server application is discovered by the device profiler which published the new service event. The service is linked to a known scenario. If no known scenario exists a new scenario is created and the user can assign the new scenario to devices. In Section 5.3.5.1 we discuss this flow in more detail.

5.1.5.4 Unknown Device event
Similar to the spoofed identity flow we currently do not have an implementation of an action which can be performed by the system. The user can get a notification is the client used by the user to control the system produces the notification on this event.
Device registration

User

Client App

Identity Manager (API)

EventManager

Device Profiler

Figure 5.3: Use case diagram for the registration process.
CHAPTER 5. SYSTEM DESIGN

5.2 DHCP Server

The DHCP server is a very important component of the system. It provides the devices with their identities. In the DHCP server, these identities are called leases. The implementation of this service in the current project is based on rogued[43], a java implementation for a DHCP server. We use the core of this service which handles parsing, creating, receiving, and sending DHCP request. We removed anything related to network leases, as we want to achieve device isolation using the DHCP server. The basic sub-components of the DHCP server are the subnets, the leases, and the lease provider. An overview of these sub-components is shown in Figure 5.4.

![Component overview for the DHCP server.](image)

5.2.1 Subnet

This component is the software representation of a subnet. It is created based on the IP identifying the subnet and the subnet mask. Each subnet contains a set of leases relevant for that subnet. To maintain this list, it is possible to add a lease to the subnet. Leases automatically are removed when they are expired.

Next to maintaining the leases an instance of the subnet component also has functions to query the subnet. It can be queried to identify if the subnet can contain a given IP address, has a free IP address, and to get a free IP address. Furthermore, the subnet has the option to register the gateway of the subnet to the LAN interface on the device running the software.

5.2.2 Lease

Devices are assigned a lease when performing a DHCP request. This software representation of the device contains a link to the device it is used for and the subnet it is in contained. Furthermore, it contains the information required for the lease itself. This consists of the start time, duration, and the attached IP address of the lease. Last we have a state which makes sure that we can keep a history of leases while we can also reuse them. This state also supports reserving IP addresses during the registration process. This avoids proposing two devices the same IP address. The leases are generated from within a subnet. The lease provider selects the correct subnet for a device.

48 Securing the Home Network
5.2.3 Lease Provider

For the new lease setup, we identify if the device can be assigned an IP address in the trusted subnet or if the device needs to be in an isolated subnet. This decision is based on the possibility to remotely execute commands, the registration status, and the number of services running. In Figure 5.5 we present the decision tree for this decision. If a device is not registered it is always isolated, the remote configuration step may be skipped. If a device is registered and is not running any services it is assigned to the trusted subnet. When a device is registered and is running services the decision is based on the possibility to remotely configure the firewall. If this is possible the device is positioned in the trusted subnet, otherwise it is isolated.

![Decision Tree](image)

Leases within the trusted subnet are provided with an IP in the subnet 192.168.0.0/16 which ranges from 192.168.0.1 to 192.168.255.255. The first address is reserved for the ONU and the last address is the broadcast address. Hence in the trusted subnet we have room for around 65,000 devices. For the isolated devices we create very small subnets in the form 10.x.y.n/30 where x, y, and n range between 0 and 255, and n is a multiple of 4. In this case the subnet range from 10.x.y.n+1 to 10.x.y.n+3. Similar to the trusted subnet the first and the last address are reserved for the infrastructure, hence we can fit exactly one device in this subnet. We have room for around 4,200,000 subnets, hence we can isolate 4.2 million devices using this setup. These subnets are created and registered on demand to limit the number of IP addresses assigned to the ONU itself.

The lease providing the IP address to the device does expire, to prevent this, devices have to renew their lease once in a while. There is no other mechanism present in home networks to force the client to renew its address. Hence only during the renew process our system can also reevaluate the lease type which needs to be assigned to a device. To speed up movement we can select a short renew time for the lease, however, this will also strain the network. To limit the requirements on the network infrastructure, we only provide unregistered devices with a short lease time while registered devices receive a longer lease time. Unregistered devices are provided with a lease time of 10 minutes and a renew time of 5 minutes while registered devices have a lease time of 2 hours and a renew time of 1 hour. This means that for unregistered devices we can switch the lease type every 5 minutes, while the registered devices can swap every 60 minutes.

5.2.4 Use cases and Usage Scenarios

DHCP is mainly hidden from the user. The user does not interact with the DHCP server. However, there are a few scenarios where the lease type changes and this might be noticeable or caused by a user. The first moment is when the device connects to the network for the first time. The DHCP
server provides the device with a temporary lease and asks the identity manager to register this device. The identity manager will process this request and asks the user for extra details. This use case is discussed in Section 5.1.5.1. In this example the lease type may change from isolated to trusted. Appendix B shows an overview of all the scenarios where the lease type may change.

5.3 Access Control Mechanism

The sub-components of the access control mechanisms are the policies, the scenarios, and the translator/pusher. Figure 5.6 show these components in an overview. Scenarios are sets of policies which are translated and pushed to the clients using the translator/pusher component. Scenario’s and policies are pure administrative components which store the access control policies. The translator/pusher is a functional component which starts enforcement of the policies.

![Component overview for the access control mechanism.](image)

5.3.1 Policy

A policy is designed to match network traffic and state what should happen with the traffic. It contains three parts of attributes. The first part describes the source of the traffic, the second part the target of the traffic, and a general third part. Both the source and the target are derived from devices participating in the scenario in which a policy is contained. By referring to SideA or SideB of the scenario set which devices present in the scenario are the source and the target.

The general part consists of the service (e.g. web server or mail) matched by the policy, and the action (block or allow). The service is described with a name and a description while it is defined by the communications protocol (TCP or UDP) and source + target ports. Generally, policies describe a source requesting a service from the target.

5.3.2 Scenario

The scenario is the component used to create the user-friendly overlay on top of the policies. A scenario consists of the attributes: name, description, side A, side B, and a set of policies. The
name and description of a scenario are visible to the user while side A and side B are set by the user.

In the user interface, the user can assign multiple objects to side A and side B. If no object is provided to side A or side B, it will match any device. The system supports a combination the following elements in either of the two:

- device profile,
- device group,
- software representation of a subnet,
- an IP address,
- an IP range,
- and a string representation of a subnet.

The scenario component has one function. This function identifies which policies need to be deployed to a specific device. The function checks if the provided device is the device providing the service for the policy if this is the case it is returned. When the selected device is the ONU all policies where at least one party is not trusted are returned.

5.3.3 Scenario Templates

A special form of a scenario is the scenario template. The difference between the scenario and the scenario template is that a scenario is supposed to be enforced while the template serves as a template to generate a new scenario. The only difference between the two types of scenarios is that the template has two extra flags which state the need for the two parties. In some cases, a scenario does not require a second party, or the second party is fixed. Using these flags we can identify which information is needed.

Initially, various default scenarios are present in the system. Attachment C shows an overview of these scenarios. Advanced users can create new templates. In future work, we can discuss the possibility to invoke the cloud to retrieve new scenario templates.

5.3.4 Translator/Pusher

The translator/pusher component is one of the core components of the system. The translator translates the policies attached to the scenarios to a device specific representation which can be used for the configuration. Usually, this representation is a command which needs to be executed to activate the policy for that specific device. A variant is of the translator/pusher is needed for each device type supported by this application. One of these variants is the variant for the ONU. This variant will translate and push the settings to the ONU itself.

The set of policies generated for a device is based on policy search function inside a scenario. Using this function, we select all relevant policies and translate them. Once the policies are translated, they are pushed to the selected device by the pusher.

The pusher prepares a configuration file which is pushed to the device. When the file is stored it will be executed. The first commands clear the firewall to remove previous configuration. Then a default policy is enabled which allows communication from and to the ONU. After that the translated policies are enabled. Last are is the default policy which blocks every incoming connection. Priorities to the rules are assigned such that the global above does not change. A potential order in priorities for the user-defined scenarios is respected, but only within that range.

An event triggered function watches for events which require the ONU to update the policies. In most cases, only the devices involved in that change are updated. Still, in some cases the set of involved devices might not be clear. In this case, the firewall of all devices is updated.
5.3.5 Use Cases and Usage Scenarios

There are many scenarios where the user can invoke an update of the access control system. However, only a few base cases exist. A scenario is added, changed or removed. A change in a scenario is also invoked when a change in its dependencies occurs. We will only discuss the use case where a scenario is added to the system.

5.3.5.1 New Service Discovered

This use case focuses on adding a scenario. A graphical representation can be found in Figure 5.7. The device profiler (Section 5.1.2) has identified a new service running on a device (dev A) and this is broadcasted using the event manager. The Access control system receives this event and identifies or creates a scenario template which fits the service. The user is notified on the availability of a new scenario for dev A. The user registers this scenario and selects a group which is allowed to use this service of dev A. This triggers the event that a new scenario is added. The translator/pusher registers this event and starts updating the policies for dev A, the selected group, and the ONU.
New service detected

User

Client App

Access Control Mechanism (API)

Event Manager

Translator / Pusher (Device A)

Device Profiler (Device A)

New service discovered

request info

details of ST

apply ST, source=A, target=T

new service for device A

new template ST for device A

new scenario S to enforce

yes, with target T

apply ST to device A

new service to enforce

new template ST for device A

new scenario S to enforce

policies for device updated

push S

Figure 5.7: Use case diagram for applying a newly detected service.
5.4 Event Manager

Events are an important communication mechanism for the system. Through events, the entire system can be monitored and individual parts are activated. Components within the system can register to the events by creating an event handler. Components from outside the system (e.g. the configuration client) can access the events through the API. We have different events, but all the events rely on a basic event.

5.4.1 Basic Event

The basic event provides basic functionality. The basic event consists of the following fields:

- id: long; The identifier of the event, new events have a higher identifier.
- time: DateTime; The time the event was raised/created.
- code: int; The event code.
- name: String; A user-friendly name of the event.
- eventData[ ] : <clazz: class, name: String, data: Object>; Data entities attached to the event identified by their class, name, and the data object itself, the order of data elements is not guaranteed (optional).
- description: String; Extra text for the event (optional).

The code and name fields are preset for a certain event. For the basic event, the code is 000 and the name is BasicEvent.

The event code is based on the component and parts of the system. The first digit identifies the main system component (1: identity management, 2: DHCP, 3: access control). The second digit states which sub-component the event originates from. The last digit is used to identify the individual events within the component.

To support both pushing and pulling of events through the API, we store all the events. The requester is able to retrieve events with a specific event code using a filter. Furthermore, it can provide an ID if its last received event. The API will flush all events after that ID to the requester. In the case the requester wants to set up a push connection, the connection is left open and new events are pushed over that connection.

5.4.2 Specific Events

We have various types of events. The different event types extend the basic event. All the specific events change the event code and name of the basic event. Moreover, the data part of an event is better specified. This allows event listeners to better understand the data part of an event. An overview of all events is attached in Appendix D.

5.5 Review of Design

To compare our security solution with the current networks’ security solution, we use the review system discussed in Section 2.5. We used the same approach in Section 3.4 to discuss the security performance of current systems. We first discuss the mechanisms we have introduced to improve security to identify how overall security is improved. Next to the we analyze the setup of the different scenarios discussed in Section 2.5.1 to review the practical security and usability.
5.5.1 Mechanisms

We have discussed eight mechanisms in Section 3.2.2 which according to our analyzes and the ITU Recommendation [2] should provide security within networks. Five of those mechanisms, encryption, digital signature, data integrity, traffic padding, and notarization, require the application of specific protocols. Even though enabling those protocols is within the scope (Section 1.2) of this project, there still is a lack of support for many protocols across all devices found in homes. This means that currently it is not possible to select a unified protocol set to provide these mechanisms in homes. Solving this problem requires software modification among many devices. This is outside the scope of this project.

The other three mechanisms, access control, authentication exchange, and routing control, are the mechanisms we have focused on for this project. For access control and routing control the mechanisms in place provide good functional support, however, the configuration is the biggest obstacle. To overcome these problems we have introduced a system which allows remote configuration of the firewalls at endpoints to provide access control at the device level. We leveraged the routing control mechanism together with subnetting to isolate unsupported devices and force traffic from and to those devices to pass the ONU, the central router which also provides access to the internet. The ONU, at its turn, can enforce the policies for those devices which cannot enforce them themselves.

For the authentication exchange mechanism, the problem consisted of a weak authentication scheme which is used in local networks to identify devices. The identity of the devices is based on the MAC and IP address of the device. As discussed earlier in Section 3.3.1 this authentication mechanisms can by bypassed fairly easily. Unfortunately, this identity of devices is used by many other systems and mechanisms such as the firewall which we use for access control. Hence, we have to better protect those identities. In the current system design, we have included a detection service which detect if a device is spoofed.

Detection of spoofed devices is based on a larger profile of system properties. This includes the host name, operating system, services running on the device together with version information and timing metrics between the ONU and the device. Currently, the system design has no action other than invoking an event which may be caught by a client. This event does, however, allow an action or another system to register to the event and counter the (alleged) spoofing attack.

5.5.2 Scenarios

In Section 2.5.1 we have introduced five scenarios. We use these scenarios to analyze the functionality of the system and the ease of use. For all scenarios, we assume that our product is deployed and fully functional client software is available.

**Scenario 1.** The first scenario was to provide a guest device with access to the internet and required services on the network for internet access, without having access to the rest of the network. Furthermore, other devices should not have access to resources available on the guest device. By default in the new network design, the ONU assigns an isolated address to the guest. Without registration of the device, this guest only has access to the ONU. During, or at some point after, the registration process a usage scenarios can be assigned to the device. In this case, we only assign the usage scenario “Guest” device to the network. The scenario has no ability for a second party since it is predefined. This scenario allows most known web, mail and chat services to be accessed. Hence, our system can provide this scenario.

**Scenario 2.** The second scenario requires no internet access, but unrestricted network access. Also, access originating from the internet needs to be blocked. By default, the device is blocked for all communication except with the ONU. Hence, we need to apply a scenario which allows unrestricted network access. Using the scenario “Unrestricted Access” we can achieve this goal. We first select the device which we want to give unrestricted access. Followed by the usage scenario “Unrestricted Access to”. Last we will assign the local network as the target of this scenario.
Scenario 3. The next scenario we want to achieve is limited access to local networks. More specifically, we want access to computers and printers (only for printing) within the network while we do not have access to local IP cameras. Similar to the previous scenarios, by default all network connections are blocked. This means that access to local IP cameras is blocked by default.

For access to computers, we can use the similar usage scenario as in the previous scenario, namely “Unrestricted Access To”. This time, however, we apply a different group of devices in the last step (compared to scenario 2). This can be done using two mechanisms: the user can create a group of devices which are computers or the user can select all the devices which are a computer. The first method is preferred as the group is more easily maintainable and can be reused.

For allowing access to a printer, we use the usage scenario “Printing” as usage scenario. As the target, we select a set or existing group of printers. This scenario only allows connection over ports often used for print servers. Other services such as file sharing which use a different connection (and port) are still not allowed.

Scenario 4. In scenario 4 we describe the situation in which we do not want access from the internet to devices. By default, this is not entirely the case. When a device is allowed to communicate to the internet we allow replies to the requests from a device to pass the firewalls. In most situations, this behavior is wanted; however, in some situations it is not wanted. In this scenario, we will assume the latter.

To achieve this setup, we will again use a predefined usage scenario named “Block internet responses”. This is the only scenario available with an actual blocking rule. Similar to the guest scenario this scenario does not require the selection of a second group of devices as this is predefined in the scenario.

Scenario 5. In the last scenario we discuss we want to achieve running a service from within the network available to the internet. For the scenario, we have chosen a web service. Deployment of this scenario can have two starts. Either the device is still in registering phase or the device is done registering and the new service has been detected. In both cases, the user only gets presented with this option if the service is detected. The detected service is automatically linked to a scenario (existing or created while scanning) and the user asked if he wants to allow or deny the scenario to be deployed. After that he can select to which devices it is available using the groups or a custom selection. In this case, the predefined group internet is used.

5.5.3 Assessment

Generally we have improved the access control and the authentication exchange mechanisms. These are direct improvements over the traditional network setup we have discussed in Section 3.4. In the traditional network setup, network identities are not protected and access control is not enabled for traffic on the local network. In the proposed system, we provide a means to detect attacks on the identities and have enabled access control.

Using improvements we have created one location, the ONU, where policies are maintained. Also using the new access control mechanism we can create rules which are very dynamic in selecting devices and have descriptions logical to the user. This improvement is visible when comparing the feasibility of the different scenarios. In the traditional setup, most scenarios require much configuration on many different devices while others are not able without added assumptions. In the proposed system, all scenarios are feasible and the configuration load is very limited. Also the setup allows specific targeted rules which results in less unwanted behavior.
Chapter 6

Conclusion

In this project, we have addressed network security for the standard home network. The goal of the project was to propose a system which can enhance the security of the home network. The motivation of this topic is the security threats introduced to the network by the large number of devices that are, and will be, connected to the home networks. In Section 6.1 we provide a short summary of our findings. The limitations and future work of this project are discussed in Section 6.2. Finally, in Section 6.3 we discuss if we have achieved all the project goals.

6.1 Overview of Our Work

In Chapters 2 and 3 we have analyzed the domain and identified problems with the security in current home networks. First we discussed the definition of network security in Chapter 2. Security generally means protection and precaution taken against threats. Within networks, important properties are confidentiality, integrity, availability, authenticity, and accountability. Hence, for a network, the threats are breaching those properties. Several measures and mechanisms exist to counter these threats. The focus of this project is in establishing trusted identities and access control between devices on the network, the core of those mechanisms.

In Chapter 3 we have discussed the mechanisms required for protecting the home network. We have identified which security technologies are present in home networks. For most devices technologies are available for each mechanism, however, the technologies themselves differ between the devices. Another problem with the technologies is that they are not be used, or in some cases cannot be used, for communication between devices within the home network. Especially for trusted identities this is troublesome. The basic identity used for communication on networks and the internet is the IP and MAC addresses (also names can be used). A technology to verify the ownership of identities exist for devices on the internet, but this mechanism does not support devices on local networks Hence, the current local network lacks the functionality to verify ownership of a network address.

Another problem is the access control technologies deployed within networks. Many devices within the local network are not deployed with an enabled firewall, hence they allow all incoming communication. The only protection these devices have is the potential firewall deployed at the modem/gateway which filters traffic originating from the internet to the local network. Enabling the firewalls on the devices in the local networks often is very difficult, because generally people do not have the knowledge to define correct policies. In environments where firewalls are deployed at the devices in the local network, they often still accept most connections originating from the local network.

Our contribution to home network security is presented in Chapters 4 and 5. In Chapter 4 we have designed and discussed various solutions which address the problems found in home networks. We have discussed various methods to improve authentication for the different authentication scenarios as well as solutions to the configuration problem found for access control. Based on the
CHAPTER 6. CONCLUSION

analysis of the solution we designed a new gateway / router called the Operator Network Unit abbreviated to ONU.

In Chapter 5 we explained the design in detail The ONU provides the new mechanisms required to centrally manage the security settings for the entire network. The system has three core components: the identity management component (Section 5.1), the DHCP server (Section 5.2), and the access control mechanism (Section 5.3). Identity management maintains and protects the identities of the devices using profiles. Using a device and a network profiler, the components scans for threats to the identities. The identities themselves are provided by the DHCP server. Based on the profile of a device the DHCP server will isolate the device or puts the device with all trusted devices. This functionality is required for the access control mechanism. The access control mechanism maintains and enforces the policies derived from the scenarios.

6.2 Limitations and Future Work

We identified several limitations to our design and assessment. The first limitation is that we currently only detect falsified identities. We do not take (automated) action against them. In future research and versions, we can elaborate and implement the possible actions we already have discussed in Section 4.1.3. This will improve the network identities even more.

The second limitation is the lack of support for automatic priorities of firewall rules. Firewall policies may overlap and when the overlapping policies have different actions (i.e. drop and allow), the specified behavior is unclear. Priorities need to be introduced to solve this problem. The problem is with assigning the priorities and providing the user with correct feedback on the consequences of the overlapping rules. Identifying a user intuitive prioritization of the scenarios and policies requires more research.

Another limitation in our design is the extent of improvements. Using the central management technology, it is possible to do much more specific configuration and actually provide an overlay network. The overlay network can provide the remaining mechanisms (encipherment, digital signatures, data integrity, traffic padding, and notarization) which rely on the communication protocols rather than the network infrastructure. Also, an overlay network can improve identity protection by adding a new device identity which can not be spoofed. Similar to the firewall setup the ONU can be used as a gateway between the more secure overlay network and the network used by the devices which do not support the technologies. Also, the ONU can serve as a gateway between multiple technologies which provide the overlay network.

The fourth limitation is in the test we used to analyze the performance. We have tested the existence of security mechanisms and have used simple use cases to discuss the usability and functionality. A more advanced security test is required to test the performance of the security mechanisms. Also, resting the usability of the software requires testing with real users rather than a theoretic analysis. The latter, however, does require the availability of a user interface which can communicate with the API of our solution.

The last limitation is the lack of security mechanisms for the ONU itself. Currently, we have only described how we can use several technologies deployed at the ONU, to more secure the network. Security of the ONU, however, is missing. In further research, we have to analyze where security is required for the ONU and how this can be enforced.

Next to the limitations in our solution design also some opportunities to further improve the solution exist. We can improve the initial detection step with the possibility to detect what device is trying to connect. Based on this property we can already assign a basic scenario set to the device. Further improvement can be reached if we share information among different networks using the cloud.

Based on the configuration information of many networks, default scenarios and device profiles can be generated. These profiles can be used to automate the selection of available scenarios of a device. Also, the set can be used to recommend a specific set of scenarios to a device.

Another opportunity for improving the design is to improve the subnetting mechanisms to create groups of isolated devices. There exist situations with many traffic between isolated devices.
All this traffic needs to pass the ONU which might not be able to keep up. To bypass the ONU, these devices have to be combined into one subnet. This removes security control between those devices but does allow for greater performance among those devices.

6.3 Concluding Remarks

According to the problem statement, we have two goals. The first goal is an overview of different mechanisms and technologies that can improve home networks. In Chapters 4 we have provided this overview. The second goal was to propose a system design which implemented several of these mechanisms to provide better network security.

The design of this system is discussed in Chapter 5. To analyze the performance between the current home network setup and the new setup, we have designed and discussed security performance measures based on the supported security mechanisms and a few configuration scenarios. Section 2.5 describes these measures. For both measures, the new system improved in terms of support and performance.

The new system is designed to tackle the challenges based on the, for the current local network setup, problematic mechanisms. When reviewing the supported mechanisms in the current network with our solution, we have identified that our system will improve at least the most important mechanisms (identities and access control). This improvement is also visible in the scenarios we have used to test usability and functionality. In the classic network setup, most scenarios are achievable, but require expertise to configure the firewall policies. Also, some scenarios require configuration at multiple devices. In the new setup, many advanced firewall options are abstracted and automated. Also the configuration is centralized. Using this mechanism, the configuration of scenarios is much easier. Also, the new system has lesser unwanted behavior because the access control policies can be applied much more specific selection of devices.
Bibliography


BIBLIOGRAPHY


[54] D. M’Raihi, S. Machani, M. Pei, and J. Rydell. TOTP: Time-Based One-Time Password Algorithm. RFC 6238 (Informational), May 2011. 24, 28


Appendix A

Class diagrams
Figure A.1: Class diagram for the identity management component.
Figure A.2: Class diagram for the DHCP server component.
Figure A.3: Class diagram for the access control mechanism component.
BasicEvent
- identifier: long
- creation: DateTime
- code: int
- name: String
- eventData: EventData[]
- description: String

EventManager
- events: List<BasicEvent>
+ addEvent(event: BasicEvent) : boolean
+ getEvents(eventType: Class) : List<Basic...
+ getEvents(lastID: long) : List<BasicEvent>

EventData
- clazz: Class
- name: String
- data: Object

AEvent

BEvent

Figure A.4: Class diagram for the event Manager component.
Appendix B

DHCP Lease Change

Table B.1: Old to new device state and lease type.

<table>
<thead>
<tr>
<th>Old Registered</th>
<th>Old Running services</th>
<th>Old Controllable</th>
<th>New Registered</th>
<th>New Running services</th>
<th>New Controllable</th>
<th>Lease</th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Continued on next page
Table B.1 shows all transitions between the possible device states and the effect on the DHCP lease. The red cross means that a device does not support the technology and a green thick means that a device does support the technology. In the last column the red cross means that a device needs to be isolated, while the green thick means that it can be part of the trusted subnet.
Appendix C

Template Scenarios

In this appendix we provide an overview of the different scenario templates. We did not include the description field for the service.

Name: Guest
side A: - (required)
side B: internet (fixed)
Description: Allows basic internet and mail access
Policies:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Service S. port</th>
<th>T. port</th>
<th>Protocol</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>80</td>
<td>TCP</td>
<td>HTTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>8080</td>
<td>TCP</td>
<td>HTTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>443</td>
<td>TCP</td>
<td>HTTPS</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>53</td>
<td>UDP</td>
<td>DNS</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>110</td>
<td>TCP</td>
<td>POP3</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>995</td>
<td>TCP</td>
<td>POP3</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>25</td>
<td>TCP</td>
<td>SMTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>2525</td>
<td>TCP</td>
<td>SMTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>465</td>
<td>TCP</td>
<td>SMTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>21</td>
<td>TCP</td>
<td>FTP</td>
<td>ALLOW</td>
</tr>
</tbody>
</table>

Name: Unrestricted Access
side A: - (required)
side B: - (required)
Description: Allows complete unrestricted access from side A to side B.
Policies:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Service S. port</th>
<th>T. port</th>
<th>Protocol</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>ANY</td>
<td>ANY</td>
<td>-</td>
<td>ALLOW</td>
</tr>
</tbody>
</table>
APPENDIX C. TEMPLATE SCENARIOS

Name: Printing To
side A: - (required)
side B: - (requires)
Description: Allows devices in side A to print on printers in side B
Policies:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Service</th>
<th>T. port</th>
<th>Protocol</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>515</td>
<td>TCP</td>
<td>LPR</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>721-731</td>
<td>TCP</td>
<td>LPR</td>
<td>ALLOW</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>ANY</td>
<td>911</td>
<td>TCP</td>
<td>ephemeral ports</td>
<td>ALLOW</td>
</tr>
</tbody>
</table>

Name: Block Internet Response
side A: - (required)
side B: internet (fixed)
Description: Block all communication which origin is the internet and has a destination which is part of side A.
Policies:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Service</th>
<th>T. port</th>
<th>Protocol</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>ANY</td>
<td>ANY</td>
<td>ANY</td>
<td>-</td>
<td>BLOCK</td>
</tr>
</tbody>
</table>

Name: Web Service
side A: - (required, single)
side B: internet (fixed)
Description: Allows devices from the internet to connect with a web server running on a local device.
Policies:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Service</th>
<th>T. port</th>
<th>Protocol</th>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>ANY</td>
<td>80</td>
<td>TCP</td>
<td>HTTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>ANY</td>
<td>8080</td>
<td>TCP</td>
<td>HTTP</td>
<td>ALLOW</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>ANY</td>
<td>443</td>
<td>TCP</td>
<td>HTTPS</td>
<td>ALLOW</td>
</tr>
</tbody>
</table>
Appendix D

Events

D.1 Identity Management Service

Device Profiles.

Code: Name: Data:
101 registerRequestEvent DeviceProfile

Description:
This event is invoked when the Identity Management Service requires a user to register a given device. This event is not used internally, but is used at a client which can register a device.

Code: Name: Data:
102 deviceRegisteredEvent DeviceProfile

Description:
This event is invoked when a device is registered to the system. This event is not used internally.

Code: Name: Data:
103 deviceUnRegisteredEvent DeviceProfile

Description:
This event is invoked when a device is removed from the system. This event is not used internally.

Device Profiler.

Code: Name: Data:
111 deviceProfileReadyEvent DeviceProfile

Description:
This event is invoked when the device profiler is finished for the first time for a device. The registration process should wait for this event in order to check which remote commands services are available. Internally this event is not used.

Code: Name: Data:
112 deviceProfileChangedEvent DeviceProfile

Description:
This event is invoked when the profile of a device changed. New services are excluded from this event. This event is not used internally.
### APPENDIX D. EVENTS

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>113</td>
<td>serviceDetectedEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when a new service is discovered. This event is used to generate a new scenario template in the access control mechanism.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>114</td>
<td>SpoofedDeviceDetectedEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when the profile of a device changes too much, i.e. it is potentially spoofed. This event is not used internally, although action might be attached to this event in the future.

**Network Profiler.**

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>unknownDeviceEvent</td>
<td>IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hostname</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when the network profiler identifies an unregistered device. Internally this event is not used, firewall policies already provide protection. Potentially the device is misconfigured, this event can serve as a warning to the user.

**Device Groups.**

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>131</td>
<td>newGroupEvent</td>
<td>DeviceGroup</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when a new device group is created. This event is not used internally.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>132</td>
<td>GroupChangdEvent</td>
<td>DeviceGroup</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when the members of a group change. This event is used by the Access Control System as a trigger to redeploy the firewall policies.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name:</th>
<th>Data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>132</td>
<td>GroupRemovedEvent</td>
<td>DeviceGroup</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when a group is removed. This event is used by the Access Control System as a trigger to redeploy the firewall policies.
D.2 DHCP

General.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>newDHCPDeviceEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

Description:
This event is invoked when a device has successfully acquired a new lease, but is not yet registered. The Identity Management Service registers to this event in order to retrieve new devices which have been connected to the network and start their registration procedure.

Lease.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>newLeaseEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

Description:
This event is invoked when a device has successfully acquired a new lease, thus, has an new IP address. This event is not used internally.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>212</td>
<td>leaseRenewedEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

Description:
This event is invoked when a device has successfully renewed his lease. This event is not used internally.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>leaseExpiredEvent</td>
<td>DeviceProfile</td>
</tr>
</tbody>
</table>

Description:
This event is invoked when the lease of a device is expired. There is no guarantee that this event will occur when the lease expires. This event is not used internally.

Subnets.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>221</td>
<td>newSubnetEvent</td>
<td>Subnet</td>
</tr>
</tbody>
</table>

Description:
This event is invoked when a subnet is created and registered to the system. This event is not used internally.

D.3 Access Control Mechanism

Scenario.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>newScenarioEvent</td>
<td>Scenario</td>
</tr>
</tbody>
</table>

Description:
This event is published when the user has created a new scenario. The translator/pusher registers to this event and starts pushing updates to the devices included in the scenario.
## Code: Name: Data:
### 302 ScenarioChangedEvent
**Description:**
This event is invoked when a scenario is changed. Similar to event 301, the translator/pusher registers to this event and updates the devices accordingly.

### 303 ScenarioRemovedEvent
**Description:**
This event is invoked when a scenario is removed from deployment. The translator/pusher only has to update the devices involved in the scenario.

### Template.

### Code: Name: Data:
#### 311 newTemplateEvent
**Template** (Device)
**Description:**
The event is invoked when the device learns a new template (for a specific device). The user interface can use this event as trigger to start assigning the new template.

#### 312 TemplateRemovedEvent
**Data:**
**Template**
**Description:**
This event is invoked when a template is removed. No service requires this event.

#### 313 TemplateChangedEvent
**Data:**
**Template**
**Description:**
This event is invoked when a template is changes. No service requires this event.

### Policy.

### Code: Name: Data:
#### 321 newPolicyEvent
**Policy**
**Description:**
This event is invoked when a new policy is created. No service requires this event.

#### 322 PolicyRemovedEvent
**Data:**
**Policy**
**Description:**
This event is invoked when a policy is removed. The access control mechanism uses this as trigger to update all devices where this policy is deployed to.

#### 323 PolicyChangedEvent
**Data:**
**Policy**
**Description:**
This event is invoked when a policy is changed. The access control mechanism uses this as trigger to update all devices where this policy is deployed to.
Translator/pusher.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>331</td>
<td>DeviceUpdatedEvent</td>
<td>Policy, Device</td>
</tr>
</tbody>
</table>

**Description:**
This event is invoked when the policies of a device are updated. No service requires this event.