NTNU
Norwegian University of
Science and Technology

Analogies of Information Security

Amund Bauck Sole

Master of Science in Communication Technology
Submission date: June 2016
Supervisor: Maria Bartnes, ITEM
Co-supervisor: Lillian Røstad, ITEM

Norwegian University of Science and Technology
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

Analogies of information security

Amund Bauck Sole

Submission date: June 2016
Responsible professor: Lillian Røstad, NTNU
Supervisor: Maria Bartnes, Sintef/NTNU

Norwegian University of Science and Technology
Department of Telematics

Abstract

Today, computers are used in virtually all aspects of peoples life, and data is collected of you almost everywhere that you go. At the same time, people have started getting more and more concerned about what parts of their lives have been collected and who has access to that data. In a world where we only seem to get more and more dependent on the computers around us and the valuable information they hold, the necessity for understanding the security behind these systems, and how they can be broken, also increases.

Everyone needs a basic understanding of security to be able to stay safe online, not only security experts and computer scientists. Concepts of cyber security and hacking can, however, seem quite intimidating for ordinary people and are usually only for the enthusiasts. How can these subject be introduced to the public in a manner that most people with no background in computer science would be able to understand?

In this thesis it will be tested wither analogies and metaphors would make it easier to teach the fundamental subjects of information security and hacking to people with no previous background in computer science and only basic computer skills. This will be done by conducting interview on people with no background in computer science to see what analogies work the best for different topics in information security. From the analogy getting the best response, a small game will be designed with the purpose of using this analogy to teach beginners in the topic.

From the interviews conducted, it was found that the topics considered complicated and difficult by the interviewees the analogies had the best effect. The most popular analogy was explaining the basics of public key encryption and this was then made into a small game taking inspiration from several gamification elements.

The interviews also reveled that analogies often had little effect for topics that were considered straightforward as they usually became redundant and inferior to ordinary explanations of the topic.

Preface

This project was performed over a duration of 22 weeks in the Spring of 2016 at NTNU. I would like to thank my professor in charge Lillian Røstad from the Department of Telematics, for making this project possible and also my supervisor Maria Bartnes at Sintef for all the help and advice throughout these weeks. I would also like to thank everyone willing to borrow me some of their time for the interviewing process. Your contribution has been priceless for this thesis. Finally a big thank you to everyone who has generously given me some of their time to proofread and comment on this report, it is very appreciated.

It has been a challenging but also very rewarding project to dive into.

Contents

List of Figures

Chapter 1
Introduction

Having basic computer skills and understanding is becoming increasingly important in society today. Everyone needs a basic understanding of security to be able to stay safe online. Most people are only exposed to the term hacking in either movies or the media and might not have the correct view of what hacking implies, how it works, or what can be done to prevent yourself from being affected by malicious hacking in the best way possible. The topic of information security can, however, be an intimidating one for many people. As a computer enthusiast, there is often nothing harder than trying to explain certain concepts and technologies to people without the same experience and knowledge about computers as yourself. It is very hard to find an easy way of explaining complicated subjects you yourself have used years studying, to any layman. This poses as a threat to the security of everyday people using the Internet, as a basic understanding of hacking and information security is important in order to stay safe. A good way of introducing everyday people to the topics of information security and hacking is therefore very much needed.

The goal of this project is to find out whether using analogies to introduce important concepts of information security and hacking, would lead to a better understanding for people without any previous background in computer science. This being anyone from kids in high school to their parents or even grandparents. It is assumed that users already have basic computers skills, but not any knowledge about information security or hacking in general. The use of gamification in the design is intended to be used to make the topic more intriguing and motivating for use.

The primary objective of this task is not necessarily the implementation of the tool, but rather to construct tasks that manage to intrigue beginners with challenging task without making things too complicated. Rather than focusing on completing a tool introducing beginners to information security and hacking, this thesis focuses on how to design tasks for this subject area with the help of analogies and gamification.

Chapter 2
Background

In the background section of this thesis, the main subjects used in the testing will be presented. In Chapter 5 the analogies used to explain the various topics will be introduced and justified.

2.1 Basic security concepts

2.1.1 Key concepts

A few key concepts are forming the foundation for information security that is important to understand and are considered some of the most crucial components of security in computer systems. They are referred to as the CIA triad. Other concepts such as non-repudiation and authentication, however, are also important concepts to have in mind when trying to design and construct secure systems.

Confidentiality

One can look at confidentiality as the privacy of information. The goal of confidentiality is the protection of information or data from disclosure to unauthorized persons. Only authorized users should gain access to information, and it must be protected when it is used, shared, transmitted and stored [9]. In other words to make sure that certain information is only available to the parties it is supposed to be available for, and to no one else. Loss of confidentiality happens when information is obtained by someone who does not get the correct authorization for the information in question. Examples of data that is important to keep confidential can be information such as medical journals, credit card numbers, and passwords.

Integrity

Information integrity refers to the protection of data from being tampered with by an unauthorized source [9]. It is vital to be sure that only authorized parties have modified information as this information is only of any value if it is correct and trusted.

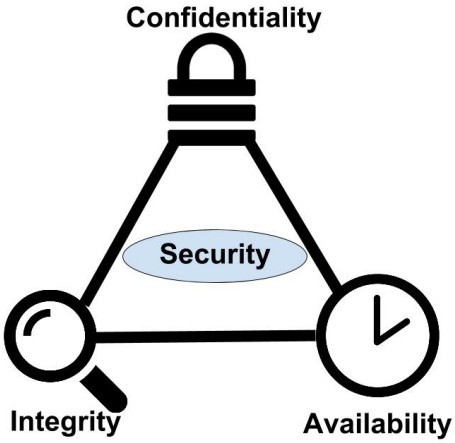

Figure 2.1: The CIA triad

If a business cannot trust the integrity of its data, that business cannot operate properly. At first glance, it might seem like confidentiality also ensures Integrity, but this is not always the case as it is possible to alter information without having the access to it. Another important trade of integrity is that if some information were to be modified, this alteration would be detectable.

Availability

Availability of information refers to the assurance that the data is accessible when authorized users [9] needs it. Any information, no matter how valuable and secure it is, will be of no use to anyone if no one can access it when he or she need it. If you put the keys to your house in a steel box and wield it shut, they will be very secure, but not of much use to anyone, yourself included. One of the most common attacks on the web nowadays is aimed at the availability of a service. A DoS, or Denial of Service, attack shuts down systems by overloading them with traffic so legitimate users will not be able to use the services when they need to.

Authentication & Authorization

Authentication can be defined as "the identification of the person or system seeking access to secure information and/or system" [9]. Authentication aims at proving that an individual or party is whom they are claiming to be. Authorization is the "act of granting users or systems actual access to information resources" [9]. This is done by checking if this person or party has the rights to perform a given action such

as running a program or altering a file. Authentication and authorization are very connected as parties must be authenticated to be able to perform tasks they are authorized to do.

Non-repudiation

The goal of non-repudiation is to "prevent the denial of previous commitments or actions" [1]. Non-repudiation, in general, is an assurance that a party in any form of transaction between two parties can not deny either sending or receiving some data. If a party performs some action, it is impossible for that party to deny later performing the action in question.

Least privilege

Least privilege is defined as follows: "Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Primarily, this principle limits the damage that can result from an accident or error. It also reduces the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur." [22]. In this way, a party's access is limited to the minimal level necessary to perform the task at hand. When applied to people, or employees, this translates to giving them the lowest level of user right possible for them to still do their jobs.

2.1.2 Cryptography

In today's society, it seems that everything you can imagine is just a few mouse-clicks away at any time. There is little information that is not in some way represented digitally and securing sensitive information is no longer as simple as locking it away in a safe. With the introduction of computers, the need for some digital counterpart to the safe and lock was necessary. The answer was found in cryptography.

The word cryptography originates from the Latin word "kryptos", which roughly translates to hidden or secret. The main goal of cryptography is to provide secure communication in situations where an unwanted third party might be listening in, or eavesdropping, on said communication. This is done by encryption and decryption of the communication. There are many ways of doing so, but the easiest way of explaining it would be to imagine putting your private message in a box and lock it with a key only the ones who are supposed to see the message would have the key to. Putting the message in the box and locking it would work as the encryption of the message while unlocking it and taking out the message would be the decryption. Cryptography is when you apply this concept of keys and locked boxes into the nonphysical world where you instead of boxes with locks use ciphers to encrypt the

message you want to keep secret. For someone without the correct key to the cipher, all messages encrypted with this cipher would appear meaningless. Only someone with the right key and cipher would be able to decrypt and read the messages sent. Cryptography also shares tight bonds to the key concepts of information security, confidentiality, integrity, authentication and non-repudiation as cryptography can be a way to achieve these concepts in computer systems.

Ciphers

Ciphers are algorithms used to encrypt and decrypt messages. When you use a cipher to encrypt a message, often called a plaintext, you will output the ciphertext of said plaintext. This ciphertext will without the correct key to the cipher be close to impossible to convert back to the original message depending on the strength of the cipher used. One of the earliest ciphers known is called the Caesar cipher, it is said that it was employed by Julius Caesar to communicate with his generals on the battlefield. In this way, an untrusted messenger delivering the messages to the enemy was not a concern as the message would be unreadable to anyone but the generals of Caesar. His cipher was very simple; it works by substituting a letter in the message with the letter three steps to the right of it in the Latin alphabet. So the message “attack at dusk” would translate into “dwwdfn dw gxvn”. This is called a substitution cipher. Ciphers have changed a lot in the modern days with the introduction of computers, so there is often made a distinction between modern and classical ciphers. Another example of a classical cipher is the transposition cipher where the ordering of the letters is changed to obscure the message.

Asymmetric vs. symmetric encryption

Encryption can either be done symmetrically or asymmetrically. With symmetric encryption, two parties share a secret key that is used for both the encryption and the decryption [2]. This key can either be the same key for both operations, or slightly different, but it is important that the decryption key is easily derivable from the encryption key. An example of this can be the Caesar cipher. At first sight, it might look like it has the same key for both encryption and decryption, but there is a small difference in that you for the decryption need to shit the letters three times to the left instead of to the right. It is not the same operation, but the connection between the encryption and decryption is easily derivable.

Asymmetric encryption does not have the same key for encryption and decryption [2]. Instead, it relies on having a pair of two different keys. One private and one public. Because of this, asymmetric encryption is also known as public key cryptography. The private key is, as the name states, not to be shared with anyone. However, the public key can be given to anyone without compromising the secrecy of any messages you send. An important trait of public key cryptography is the fact that encrypted

communication can be achieved without having first to exchange a secret key. If you encrypt some message with someone's public key, only the private key will be able to decrypt that message. This works by having parties disclose their public key to everyone, hence public, and if someone wishes to communicate privately with this party, he or she will encrypt his or her message with the other party's public key. Now only the party with the private key matching this public key will be able to decrypt the given message as shown in figure 2.2. This makes secret communicating with multiple different parties significantly more practical as a different secret key is not needed for every separate communication line.

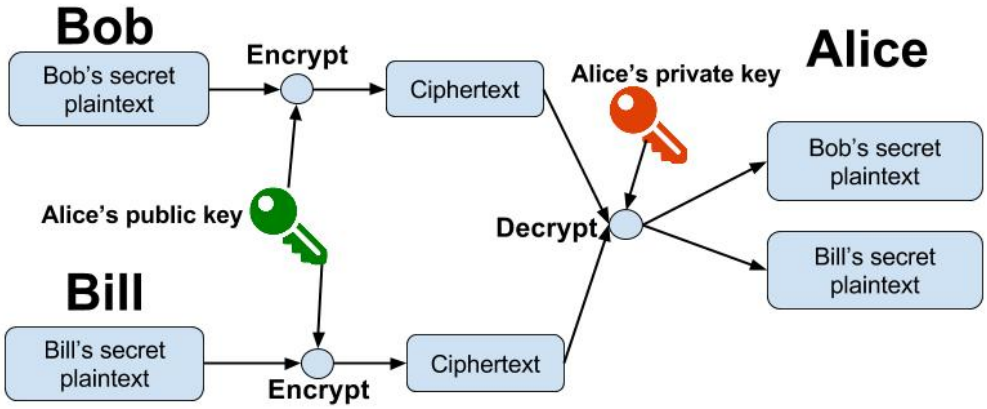

Figure 2.2: Public key cryptography

2.1.3 Code breaking

All ciphers today give away some information about the plaintext it originated from. It is, therefore, possible to break the cipher by using different methods depending on the cipher that you are trying to break.

Brute force attack on a cipher is the simplest way of cracking a cipher and works on all ciphers. However, it is also the less efficient as it involves simply trying all possibilities there is one by one until a match is found. Brute-force can, however, be used in combination with other techniques by applying it after some options are eliminated and the key-space has shrunk enough for brute-force to be feasible. With today's computers, that amount of possible keys feasible for brute-forcing is getting surprisingly big.

Frequency analysis

For some simple ciphers, such as the Caesar cipher, it is possible to use frequency analysis of the letters in the ciphertext to help break the cipher. In figure 2.3 the overall frequency of the English and Norwegian language is presented in a graph. A text does not need to be more than a few sentences long before this pattern becomes visible when looking at the occurrence of the letters in the text. When using simple mono-alphabetic substitution ciphers, looking at this simple statistical property of the text, it is simple to figure out which letters in the ciphertext corresponding to which letters in the plaintext. The letter in the ciphertext occurring most frequently is in the English and Norwegian language most probably representing the letter "E" in the original plaintext. As all other similar statistical properties, this one also follows the law of large numbers, which states that "under certain very general conditions the simultaneous action of random factors leads to a result which is practically non-random." [18]. So the larger the text sample, the more accurate the frequency analysis will be. There are of course exceptions to this as some texts for some reason might use certain characters more or less than the average. An example of this is the novel written originally in French by George Perec, later translated into English by Gilbert Adair, where neither the original, nor the translation contained any instances of the letter "e" throughout the whole novel [14]. This is, of course, an extreme case and almost all English texts of a certain length would show the close to the same graph as the average for the English language. In figure 2.3 the character frequency of this thesis is also shown, and it is very similar to the one of the English language in general. With the use of poly-alphabetical ciphers, however, this statistical property is less visible as the same letters in the plaintext would not always become the same letter in the ciphertext.

Attack models

When trying to break an encryption, the attacker may have different levels of access to the encryption mechanism. These various levels of access are called attack models, each giving the attacker a different amount of information to work with. The more elaborate access the attacker has, the more useful information he can get. Here we will mention the four most common attack models [11]:

- **Chosen plaintext attack:** The attacker can choose any plaintext and get to know the corresponding ciphertext through some kind of encryption oracle which is able to produce the ciphertext for any plaintext it is given without revealing the key to the encryption.
- **Known ciphertext attack:** The attacker is in the possession of some ciphertext but does not know what the corresponding plaintext is. He does not choose the ciphertext obtained and is not able to produce any more of it.

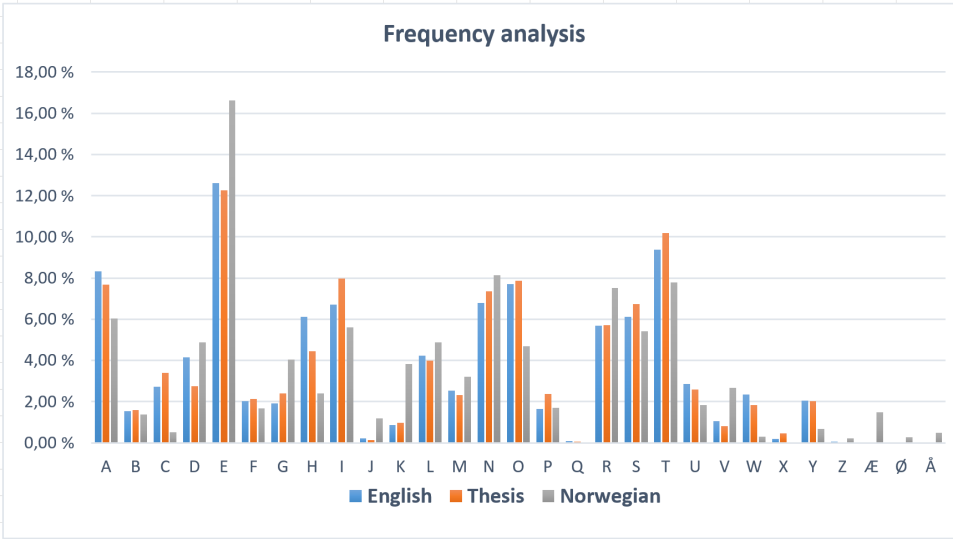

Figure 2.3: Frequency analysis of the English language, the Norwegian language and this thesis

- **Chosen ciphertext attack:** The attacker is able to choose any ciphertext and obtain the corresponding plaintext to this ciphertext.
- **Known plaintext attack:** The attacker is in the possession of some plaintext and its encoded ciphertext. He did not choose any specific text himself and does not know what the text is about. He is also not able to get any more plaintext-ciphertext pairs.

2.1.4 Malwares & countermeasures

Malwares

Malware is a term used to generalize all kinds of software made to cause harm intentionally. There are several different distinctions in malware, and some of the most common ones are described below.

- **Viruses.** A computer virus was first defined back in 1986 as "a program that modifies other programs to contain a possibly altered version of itself" [4]. It is defined by its ability to copy itself and spread to other systems.
- **Worms** are very much like viruses in the way they reproduce and spread, but they differ from viruses by their ability to self-propagate from one computer to

another [12], while viruses require some action to be performed by the user of the computer, to propagate.

- **Trojans** got their name from the ancient story of how the Greek built a huge monument of a horse as a sign of peace and gave it to the Trojans. The Trojans accepted the monument and brought it back to their heavily fortified city of Troy which the Greeks had failed to siege. However, the Trojan horse was filled with Greek soldiers, now on the inside of the city walls, able to open the gates for the Greek army. This, in turn, lead to the conquering of Troy by the Greeks. Trojans work in the same ways on computers. They lure their way onto your secure computer by concealing themselves as “good” software that a user consciously will download on their computer not knowing it is hiding malicious code[12].

- **Rootkits** are a hidden software with the intention of infiltrating an operating system or database without getting detected or removed and perform some specific, malicious operation [3]. They try their best to hide their presence on a computer, so they are not detected while for example collecting or deleting private data from the system it has invaded as well as leaving it vulnerable to other attacks. They are often designed to infiltrate the "root", or kernel, of the program and thus operating without showing their presence in a system to the user.

- **Spyware** loosely describes computer software that in some way tracks a user’s activities online and offline [17]. It can, for instance, be used to provide targeted advertising or other types of activities users usually find very intrusive. A keylogger is an example of spyware that, when installed o a computer, is able to track every keystroke made and send this information an attacker that, for instance, can look for passwords written on the keyboard or other sensitive information.

Countermeasures

As malware exists, methods of protecting against such threats have also been constructed. Below, some categories of malware protection are listed, as well as how they work.

Signature-based detection Signature based detection is a device or software that monitors network or system activities for malicious activities, by looking after code on the machine resembling already known malicious code. If any signatures are found, this code will then be deleted.

Behavior-based detection Behavior based detection is a device or software that monitors network or system activities for malicious activities, looking for activities normally associated to different types of malware or any other activity on the system deviating from what is considered to be normal for that system. this does not look at the code of a program, as signature-based detection systems, but on the behaviour of a program. If a program acts suspiciously, it will be removed form the computer.

Firewalls A firewall is a network security system, either hardware- or software-based, that controls incoming and outgoing network traffic based on a set of rules [13]. Acting as a barrier between a trusted network and other untrusted networks - such as the Internet - a firewall controls access to the resources of a network through a positive control model. This means that the only traffic allowed onto the network is the traffic following the policies defined by the firewall; all other traffic is denied. Firewalls differ from intrusion detection systems in that they only look "outwards" for possible threats, as opposed to behavior based- and signature-based detection systems, that look for threats already in the system.

Sandboxes Sandboxing isolates programs from the rest of a system, preventing malicious or malfunctioning programs from damaging or snooping on the rest of your computer. It provides strictly controlled resources for guest programs on a computer, restricting its privileges, so that potentially malicious code cannot cause any harm to your computer when only run inside the sandbox.

2.1.5 Password strength

The strength of password relates to the number of different possibilities there is for the password as well as its randomness/entropy. If someone wants to break a password, he or she would have to guess what that password is in some way, so the more possibilities there are to guess, the harder it would be to break it. Making a password longer is one way of increasing the amount of possible keys. Also, the use of more than just standard characters will have this effect. Not using well-known words will also strengthen a password as an attacker would have to try combinations of letters that are not in any dictionary, making more guesses required. Passwords are also enhanced by replacing them once in awhile so that an attacker cannot rule out earlier failed guesses when trying to crack a password.

2.1.6 Social Engineering

www.social-engineer.org [10] says that social engineering is a blend of science, psychology, and art, and defines it as "Any act that influences a person to take an action that may or may not be in their best interest." Social engineering is in itself not necessarily a malicious act. It can be used for both good and bad but involves

influencing other people's actions in some way. Social engineering is said to be used in over 66% of all attacks by hackers, hacktivists and nation states [23]. [10] has broken down malicious social engineering into three top methodologies, Phishing, Vishing, and Impersonation.

- **Phishing** is the method of using emails appearing to be from legitimate to gain influence or personal information. Around 50% percent of all emails sent in June 2016 were spam [24]. Phishing represents 77% of all socially based attacks, and 88% of all reported phishing was clicking links within emails [23].
- **Vishing** involves gaining information or trying to influence peoples actions over the phone. The average loss for targeted businesses in 2013 was 42.546$ per account [23].
- **Impersonation** is the practice of presenting oneself as someone else to obtain private information or access to a person, company or computer system.

Real world examples of social engineering are many. For example, the classical pyramid scheme is a type of social engineering relying on gaining the trust of other people by the promise of high returns by doing nothing else than handing over your money and getting others to do the same. Nigerian Advance Fee Scheme scam works in a similar way. The goal of the scam is the delude the recipient of the mail to think that they have been singled out in a very lucrative deal, and for only a small up-front payment they will gain great economic rewards.

2.1.7 SQL injection

Most websites have its data stored in databases which it communicates with to extract certain information from this database. The programming language used to communicate with databases is called SQL. With this language, you have the power to both look up in the database and modify it in any way you see fit. There are commands for looking up and extracting certain information, and there are commands able to alter the database, including adding and deleting data. Web sites use a database for all sorts of different things, one example being to store different users and their passwords for that website. The language is quite similar to the English language when written, so a command can often be understood even if one is not familiar with the language.

SQL injection happens when software accepts data from an untrusted source and fail to validate this data properly. Then it is possible for attackers to construct their own SQL query dynamically to the database backing that software [7]. The impact

of a successful SQL injection can be anything from an extraction of sensitive data to deletion of data or dropping of tables in a database.

Figures 2.4, 2.5 and 2.6 are an example of a application vulnerable to SQL injection and shows how it can be exploited.

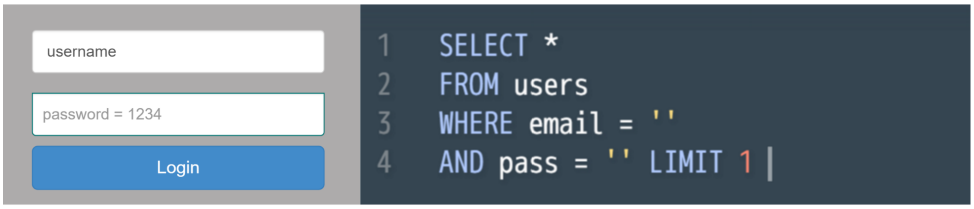

Figure 2.4: An empty login page vulnerable to a simple SQL injection as well as a screen shot of the the application code looks like behind the scenes at this moment.

In figure 2.4 an empty login page vulnerable to a simple SQL injection is shown together with how the application code looks like behind the scene. Figure 2.5 shows an example of how an ordinary input request would be added to the SQL query when a user inputs the values “user@email.com” as the username and “password” in the password field of the login page. The query would then check in its database if the password belonging to the user “user@email.com” would indeed be “password”, if this is not the case, access would be denied.

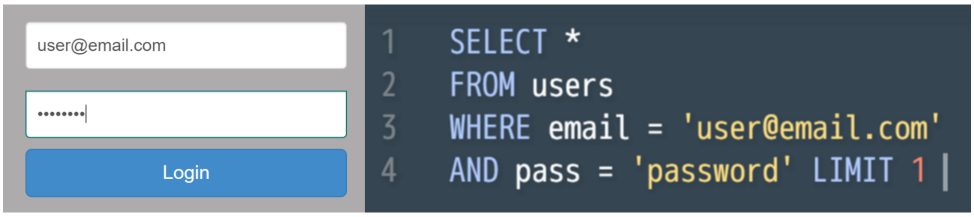

Figure 2.5: Example of how an ordinary input request from a user, named ’user@email.com’ typing in ’password’ as the password would look like.

In figure 2.6, however, an attacker has inserted the password *‘or 1=1–*. The quote is inserted directly into the SQL string, making the application think that the input from the user is now over and is expecting a command to be performed. This command has been crafted by the attacker stating that you will get access to the page of user “user@email.com” if 1 equals 1. The two dashes in the end tell the

application to ignore the rest of the SQL query. An attacker has now managed to gain access to a user's account without having access to the password.

Figure 2.6: Example of how an attacker can manipulate his input in a specific way in order to gain access to an account without knowing the password for said account.

2.1.8 Buffer overflow

A buffer overflow is a result of putting more data into a buffer than the buffer can handle [19]. It can occur when a program, while writing data to a buffer, overflows the memory space set aside for said buffer, causing the data to overwrite adjacent memory locations it is not intended to be writing in, corrupting these memory locations. This anomaly is made possible when bounds checking is not performed on the data intended for the buffer, which is supposed to check whether the data is too big for the buffer space set aside for it. For a full understanding of how buffer overflows are performed, detailed knowledge about the memory layout of computer functions. This is however not relevant to this thesis; it will not be explained in such detail when introducing the topic to beginners.

In figure 2.7 two variables have allocated some memory adjacent to each other in the memory, a 7-byte long string buffer, 'foo', and a two-byte integer, 'bar'.

```
char        foo[7] = "";
int         bar    = 1979;
```

Var name	foo							bar	
value	Empty string							1337	
hex	00	00	00	00	00	00	00	05	39

Figure 2.7: the buffer 'foo' and variable 'bar' lay adjacent to each other in memory, as long as the string is empty, its data only contains of zeros.

If someone tried to store the 8 byte long string ‘overflow’ in the 7 byte long buffer, this would lead to the last letter in the word, ‘w’, being written into the memory space of the integer, ‘bar’, corrupting this integer by changing the ‘05’-byte into the hex-value of the letter ‘w’, 77. This is shown in figure 2.8

Var name	foo							bar	
value	‘o’	‘v’	‘e’	‘r’	‘f’	‘l’	‘o’	30521	
hex	6F	75	65	72	66	6C	6F	77	39

Figure 2.8: The buffer for ’foo’ is overwritten and the value held in ’bar’ is corrupted by the overflow.

This vulnerability in the program can be exploited by an attacker in several ways. It can overwrite local variables near the buffer in memory on the stack to change the way the program behaves. It is also possible to overwrite the return address of the specific stack frame, changing it to an address specified by the attacker, usually pointing to a buffer filled with input from the attacker himself to execute some code.

2.2 Gamification

Gamification can be defined as the use of game elements and game design in non-game contexts. It is about understanding what makes games successful and engaging to the users, and then thoughtfully applying these aspects of the games to other situations that are not themselves games. Examples of such elements are leader boards, badges to award achievements, and different types of point systems. When these elements are introduced into a situation that is not a game, it is called gamification. Gamification can be a powerful tool, and is often used in businesses to in some way increase their revenue. An example of this is Nike+. Nike sells running shoes, and therefore, Nike wants people to move more. Nike+ is an accelerometer that fits in the sole of a shoe and tracks all your movement when running or walking. It then has the possibility to track how far, how fast and how often a person exercises. This information is used in an app to compute everything from fastest run and average running speed to what weather it was when you were running and what days you usually go for a run. This information is then presented in the app with a lot of well-known gaming elements as challenges, high scores, trophies, and progression. The app is meant as a motivator for people to do what Nike wants you to do, go running with their running shoes. Gamification is not to be mistaken with game-based learning [8]. The main

differentiation between the two is that gamification does not involve participants to design their own game or play a commercially produced game, like Minecraft [16], but rather use specific elements of games to evolve the learning experience.

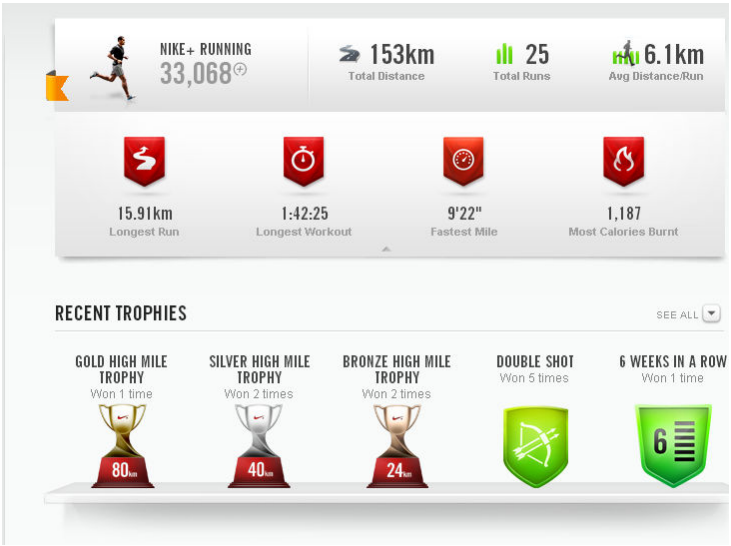

Figure 2.9: Screenshot for the Nike+ app showing how they use gamification as motivation for running [5]

Gamification can not only be used as a business idea but also as a powerful learning tool. Motivation is something needed to learn any material, and using gamification can be a great technique for increasing and engaging participation in the learning process. A study from 2011 also shows that 55% of the respondents that were employed would be interested in working for a company that offers games as a way to increase productivity [21].

2.2.1 Gamification elements

When applying gamification, there are several different elements of gaming that can be used. Oracle [20] has made some gamification guidelines for businesses where they have broken the various elements of gamification into four main categories, each with their own subcategories:

Feedback Mechanisms

Feedback Mechanisms elements are used to give information to the users about their Performance. Some feedback mechanisms include:

- **Points:** Values earned by performing certain actions, can be combined with other elements such as levels and leader-boards.
- **Levels:** Shows users improvement over time.
- **Badges:** Tokens rewarded to users for some specific behavior.
- **Bonuses:** An extra set of rewards for completing a set of actions.
- **Notifications:** A feedback mechanism is alerting users of changes, such as new tokens or levels available or earned.

Indicator Mechanisms

Indicator Mechanisms are elements that in some way defines a user's relative position in the system or compared to other users. Some indicator mechanisms include:

- **Countdown:** Countdown indicators that give the users a sense of urgency, such as encourage increased activity within a certain time frame.
- **Progression:** Progression indicators helps the users understand where they currently are in the system, and what it would take to go a step further. They help people continuing their interactions within the system and can be combined with, for instance, levels and badges.
- **Leader-boards:** List of the top users in different areas. They show users their relative placement in regards to other users of the system. If not careful, leader-boards might demotivate those in the bottom.

Game Design Mechanisms

Game Design Mechanisms are elements used the larger goal and reward states and include:

- **Quests, missions and challenges:** these three aspects involve the completion of certain actions often in a particular path or order. Usually, quests and missions are mostly about exploration and discovery, while challenges often include time limits.
- **Competitions:** Events where users directly compete with each other to encourage healthy rivalry and leading to some prize, advantage or honor for the winners.
- **Virtual economies:** Implements the option for users to trade in their success for some kind of goods or services.

Physiological Mechanisms

Physiological Mechanisms take advantage of the ways that people think about situations they encounter. Some gain design elements for physiological mechanisms include:

- **Loss aversion:** Capitalizes on people's tendency evaluate any potential loss as greater, and of more significance, than the same amount of potential gain. This can help increasing motivation by framing situations as potential losses rather than gains.
- **Appointment dynamics:** Will require users to regularly "check in" to the system for either a positive effect such as bonuses, or -by implementing loss aversion- avoiding negative effects.

2.2.2 Related work

Gamification in computer security

Today there exist a quite a few different "games" online with the purpose of practicing your hacking skills and teaching computer security. Some examples of such challenges are OWASPs Hackademic Challenges and www.hackthissite.org. These tools are built up using a leveling system, where you start off doing relatively straightforward challenges to retrieve an access code new levels, gradually increasing in difficulty. There are also commercial tools such as Semantic's hacker academy, providing both instructional videos together with hands-on labs to practice your skills. Another type of computer security based game for education is the game CyberCIEGE aiming to teach computer and network security concepts through simulation of real world scenarios. This game takes its inspiration from games such as SimCity where you are set in a fictional world operating and defending your networks against threats by implementing different security measures while trying to balance between budget, security and productivity.

Analogies in computer security

Theanalogiesproject.org "The aim of the Analogies Project is to help spread the message of information security, and its importance in the modern world. By drawing parallels between what people already know, or find interesting (such as politics, art, history, theater, sport, science, music and everyday life experiences) and how these relates to information security, we can increase understanding and support across the whole of society." [25]

The analogies project is a nonprofit organization where information security people from all over the world can contribute by posting analogies they have used

to explain different information security concepts to a layman. The organization aims at drawing parallels between things ordinary people might already know or find interesting, like history, art or sports, to concepts and problems concerning information security. They invite anyone to contribute and post the analogies on their web page for anyone to see. As of April 2016, there are over 200 analogies posted by various people on their web page since September 2012.

Chapter 3
Methodology

In this chapter, the methodology used to construct this master thesis will be described. It can roughly be divided into three main sections

- Choice of topics
- Experiment setup
- Testing & evaluating

3.1 Choice of topics

When finding the right topics for the testing, there are a few factors that need to be taken into account. The first step is to choose what subjects this thesis will focus on as it is not feasible to have a general introduction to all of information security. Subjects best suited for this thesis would be the ones where it is possible to make usefl, understandable analogies for the basic concepts as well as being subjects important for the general understanding of information security. The main criteria, however, is the possibility to make strong, understandable analogies as this are what the thesis is looking into in its problem description.

When suitable subjects are found the first step is to design a suitable analogy of the topic. It is crucial to understand the limitations of an analogy as they are seldom 100% translatable to the actual topic it is representing. The boundaries of the analogies made in this thesis will be discussed in the Discussion chapter later on. At first, as many topics as possible would be considered for the project. Through an iterative process, they would then get narrowed down to the most promising ones, regarding their relevance and quality of an analogy. Many topics were either not that relevant, or difficult to make reasonable and helpful analogies about.

For each of the subjects chosen, two versions explaining it were made, one with ordinary explanations, and one using analogies to explain the same thing. These two versions would then be used int the testing explained in the next section.

3.2 Experiment setup

In this section, all the different analogies that were used when creating explanations of the different topics are introduced in the same way they are presented to the test subjects.

3.2.1 Key concepts

Instead of looking at the security of a computer system. We are here going to look at the physical security of a hospital taking this matter very seriously. As with many computer systems, hospitals are always running and rely on many types of security to be properly trusted by the public. Below are some examples of security measures that can be implemented in a hospital that could also apply to computer systems

1. In the case of a power outage, the hospital has its own backup generator so it would be able to stay operational even then.
2. The people working in the hospital all have their own employee card with their picture and use this to get into the building with their personal access code as there are areas in hospitals only meant to be accessed by the employees and not the public.
3. Inside the building each employee only has access the areas that have something to do with what they work in and not the rest. For example, the secretaries by the main entrance have access to many areas in the hospital which the public does not, but not to the operating rooms, as it would have no purpose for the to do so, and they can potentially cause harm by being there.
4. Employee's access cards are also used when performing different working tasks like logging onto a computer, printing out documents or modifying journals. In this way, the managers will have control over who used what tools in the hospital and at what time.
5. When doctors or nurses hand out prescriptions and such, they are required to sign the required papers to ensure their involvement in what went down, if someone is found injured due to the wrong treatment, it would be easy to see who is responsible, as prescriptions would have been signed by the doctor or nurse that handed them out. It would, therefore, be difficult for this person to deny their involvement.

6. All communication between employees and patients about their medical condition is to be strictly private and only shared with the parties involved. All information about a patient is only to be obtained by the patient and his or her doctor, no one else.

Participants will here be asked to identify which of the key concepts are covered by the different security measures put in place by this hospital.
Correct answers are: (1) availability, (2) Authentication & Authorization, (3) Least privilege, (4) integrity, (5) non-repudiation, (6) Confidentiality.

3.2.2 Cryptography

In today's society, it seems that everything you can imagine is just a few mouse-clicks away at any time. There is little information that is not in some way represented digitally and securing sensitive information is no longer as simple as locking it away in a safe. With the introduction of computers, the need for some digital counterpart to the safe and lock was necessary. The answer was found in cryptography.

In this task, we will look at how communication over the Internet, and other insecure communication lines, is encrypted. You can look at encryption as locking the information you are sending in a box with a padlock. Only the people who have the key -or the combination- for the padlock will be able to access the information inside. In the digital world, padlocks do not exist, however, but a mathematical version of a padlock is possible to create. This is called a cipher which was explained in the previous part. So you can look at the cipher as the padlock, and the key used for that cipher as the combination for that padlock. This means that knowing the cipher used to encrypt a message does not alone give you the possibility to decrypt that message. You will need the key as well. The same goes for padlocks, the same type of padlocks does not all have the same combination to open it, even though they all use the same kind of mechanism actually to lock. A problem that occurs by using padlocks to encrypt messages, however, can be seen in the example below.

Imagine two people, Alice, and Bob, wishing to communicate privately with each other. They live in houses on the opposite side of the road and cannot leave their respective houses, so their messages would have to be delivered by a third person, Eve. The problem is that Eve is terribly curious and reads all messages she gets her hands on. Alice has a box and a padlock she can choose combination key for. Bob also has his own padlock with his own choice of combination.

How can Alice and Bob send messages to each other without letting Eve know what the messages are about, knowing that Eve will deliver messages between them?

For example, if Alice wants Bob to know the combination for her padlock so he can open her locked -and secret- box, this combination would also be known to Eve as it will have to be sent unlocked to Bob since he does not yet know the combination. On the other hand, sending messages locked in the box -so that Eve can not read them- will also result in Bob not being able to read them. This problem is a very applicable issue to Internet communication and encryption, as the sharing of a common key also will have to be done in the eye of the public unless people actually meet each other in person to exchange keys, which is highly unpractical. This is a problem solved by the public key encryption mentioned in the previous part. The answer is given below, but try to give this some thought before reading it.

A way to do this is if Alice sends her message in a box locked with a lock only she has the key to. Eve can therefore not open this box as she does not have the key, the problem is that neither can Bob. However, there is a solution to this. Receives the box, he can not open it, but what he can do is to put on yet another lock on the box which only he has the key to. Great, now there are two locks on the box, and neither of the there people can open it alone, very helpful Bob. However, this is where the "magic happens" Bob then sends back the double-locked box with a very confused and frustrated Eve, to Alice. When Alice receives the box, she then continues to remove her own lock from it before yet again sending it back to Bob. Now the box only has Bob's lock on it, but curious Eve can still not open it. However, when she delivers the box back to Bob, only his lock is left, and he can now safely open it to see what Eve has sent without worrying if Eve has snooped in their privacy.

3.2.3 Code breaking

In this section, we will look closer into how encryptions and codes meant to keep information private and secure can be broken. The reason why this is possible is that all ciphers in some way give away some information about the data it originated from. This is easiest to see when looking at simple ciphers but also apply to modern ciphers used in online communication today.

The example below is intended to show how looking at frequencies of characters in encrypted messages can reveal information about the original message before it was encrypted.

Frequency analysis

Analogy: You have a pile of mixed ingredients for a magic potion in front of you. You know that this is the ingredients for a "healing potion" and you also have a recipe for healing potions at hand. The recipe, however, is not always followed to the exact amounts, but you assume that the amounts of each ingredient is somewhat the same. Your potions professor asks you to identify all the ingredients in the pile.

Your healing potion recipe says:

- 800g of fairy dust
- 200g of Asian dragon scales
- Testing & evaluating
- 325g of finely blended lammasu manes brined in snakes
- 100g of angel feathers
- 625g of elf toenails
- 3g of roots from a juniper berry bush tended by a pixie
- 50g of chopped dragonfly thoraxes
- 730g of bark from a poplar tree gathered by a dwarf

As for most ordinary people, your knowledge about magical ingredients is rather limited, you can differentiate between them, but you have no idea which ingredient is which. How can you, using the information you have been given here, manage to identify which ingredients in the pile is correspondent to the different ingredients in the recipe? (hint: you have a scale at your disposal)

Attack models of cryptanalysis

In the next task, you are given four examples of how an encryption cipher can be broken using different techniques described below, where each technique uses a different kind of information when trying to break the code. Pair the examples with their correct definitions. How would you order the attack models from weakest to strongest regarding how easy it would be to break a code?

Example 1: Eve steals some encrypted messages from Bob intended for Alice, and will try to decrypt them without any additional information.

Example 2: Eve knows that Bob always starts his encrypted messages with the word “begin”, and will try to use this information to help her decrypt Bob’s messages.

Example 3: A type of file storage encrypts all files it receives with the same encryption key and then lets everyone see everyone else’s encrypted files. Eve, knowing Bob has encrypted files in the filesystem, also encrypt some of her own files there and looks at the resulting ciphertext. By looking at her own files both in

plaintext and encrypted she will try to obtain the encryption key to decrypt Bob's files.

Example 4: Eve intercept encrypted messages from Bob to Alice and exchanges Bob's encrypted message with her own ciphertext. She then eavesdrops on Alice to see how she tries to decrypt the message she thinks is from Bob.

Known ciphertext: The attacker is in the possession of some ciphertext but does not know what the corresponding plaintext is. He does not choose the ciphertext obtained and is not able to produce any more of it.

Known plaintext: The attacker is in the possession of some plaintext, and it encoded ciphertext. He did not choose any specific text himself and does not know what the text is about. He is also not able to get any more plaintext-ciphertext pairs.

Chosen plaintext attack: The attacker can choose any plaintext and get to know the corresponding ciphertext through some kind of encryption oracle which is able to produce the ciphertext for any plaintext it is given without revealing the key to the encryption.

Chosen ciphertext attack: Attacker is able to choose any ciphertext and obtain the corresponding plaintext to this ciphertext.

3.2.4 Malware countermeasures

The analogies below describe specific groups of countermeasures for malware; these are the following: signature based detection, behavior-based detection, firewalls, sandboxes and security awareness training. Look up the definitions of these words and try to figure out what analogy describes what countermeasure.

Countermeasure analogy 1
Someone or something in the house knows what items that are unwanted inside by checking all items in the house against a "dictionary" containing a list of all unwanted items for the house. Now and then the house is checked to see if any items there match any of the items in the dictionary, if so, they will be thrown out.

Countermeasure analogy 2
Someone or something in the house is aware of how certain unwanted items usually behave and can recognize if anything would be behaving like this in the house no matter what the item might look like from the outside. So disguised unwanted items, like a camera hidden in a picture, could still be detected only by its behavior. It will also sense irregular activities not normally happening in the house and assume that this behavior is malicious.

Countermeasure analogy 3
You have a secure gateway to your house where everyone and everything are checked for what they are bringing there. It works kind of like a VIP list, only people and traffic fulfilling the criteria on this list will be allowed access to the house. If the list for example stated that only people wearing green socks were allowed, anyone not wearing green socks would have their access denied

Countermeasure analogy 4
Everyone you do not know, or trust is only authorized to stay in your hallway when visiting, which is isolated from the rest of your house and contains none of your valuables.

Countermeasure analogy 5
Everyone in the house has the responsibility to remember to lock doors, close windows and turn on the alarm whenever it is best. The best of door locks and security systems are pretty useless if you do not remember to lock the door and turn the security system on when it is supposed to be.

3.2.5 Match the analogy

In this section, several analogies are written down, but they do not explain what exactly they are the analogies for. Look through the previous part -all the definitions for these analogies are there- and see if you are able to match the analogies with their correct definitions.

Storage robot (SQL injection)

Imagine you are a robot working in a warehouse filled with boxes. Your job is to take boxes in the warehouse and deliver them to the checkout desk. As you are a robot and need to be told what to do, a programmer has given you a set of instructions on a paper request form. People can then fill out the paper form and hand it to you. The paper form looks like this:

```
Retrieve item number _____ from section _____ of ail number _____,
and place it on the checkout desk.
```

A normal request form can look like this:

```
Retrieve item number 1337 from section C3 of ail number 42,
and place it on the checkout desk.
```

The text in bold is provided by the user on the paper request form. You(the robot) do what the form says, you go to section C3, then find ail 42, retrieve box number 1337 and put it on the checkout desk.

However, what happens when the user gets a bit too creative when filling out the form and puts something like this?:

Retrieve item number **1337** from section **C3** of ail number **42, and throw it out the window. Then go back to your station and ignore the rest of this form.** and place it on the checkout desk.

As you are a robot, you do exactly what is said on the form. You find the item, and you throw it out the window, then you ignore the rest of the form and go back to your desk.

Rapunzel (Social engineering)

In this story from the Brothers Grimm, Rapunzel's is imprisoned in the tower of the Enchantress named Gothel, because her father is caught in the act of stealing some Rapunzel (a type of salad leaf) to satisfy the cravings of his pregnant wife. Catching him in the act, Gothel makes him promise to hand over his unborn child as punishment for the theft. As Rapunzel grows up in the tower, her hair becomes her only access route into and out of the tower, as Gothel makes a great attempt at protecting her asset (Rapunzel) by implementing an early example of biometric security, by Rapunzel identifying Gothel's voice before letting down her hair to her and only her. In short, goth was trying to keep something, safe and private; that be Rapunzel. She also has considerable resources at her disposal to do this being able to construct a very secure tower with seemingly no entrance. However, despite all Gothel's efforts, in the end, it takes very little in the way of persuasion for the prince to persuade Rapunzel to let down her hair and allow him access to both herself and the tower. After some basic surveillance, he mimics Gothel and Rapunzel lets down her hair, quickly resulting in the prince gaining access to the tower. Gothel used her considerable resources and intellect to safeguard her asset, but she forgot to address the human element of the story. She could not, in the end, prevent Rapunzel from human desires, fears, and emotions, and in the end, the measures she had put in place failed, completely, leaving her with a cool inaccessible tower, but NOT in possession of her Rapunzel.

Shooting competition (Password security)

Consider a target shooting challenge where the player has to shoot down several targets, and each target needs to be knocked down in succession to win a prize. You can not miss any targets, as you would then have to start over again until you can shoot all targets in the right order without missing. If there are fewer targets, it will be easier to win the prize. The shooter is also blindfolded to make this even harder, and they would have to guess the location of the targets to be successful. However,

if they remain methodical it can be expected that they will eventually be successful -it might take some while- but they'll get there in the end. To make the challenge even harder we can do a number of things:

1. Increase the number of targets. This will make it much harder to hit the correct targets in the right order.

2. Have a bigger area for placing the objectives. This gives you more possibilities on where targets might be.

3. Periodically move the targets so the shooter can no longer rule out unsuccessful shots, and will need to try them all again. If he found out where the first target is, he can no longer aim here first every round since the placement and order of the targets will have been changed.

Money table (Buffer overflow)

Imagine you have a list of people you owe money to, it looks like this:

Name:	Money (in $):

Figure 3.1: Your empty money table

When you get a new entry to your list, you have the routine of writing the money you owe first, and then the person you owe it to in that order. The pen you are using can also overwrite earlier text in the same position, as this is how computer memory works, being able to overwrite other data.

You borrow 100$ from someone and they tell you there name is Charles Ponzi. You write the amount(100$) and the name(Charles Ponzi) in your table that now looks like this:

Name:	Money (in $):
Charles Ponzi	100

Figure 3.2: Your money table with one entry

Your list will then later remind you that you owe Charles Ponzi 100$, so you pay him back the amount, and you then clear this person from your list.

Later you borrow another 100$ from someone, but this time, the person tells you his name is Charles Ponzixxxxxxxx99999999999. You write in the amount (100$)and then the name (Charles Ponzixxxxxxxx99999999999) in your table. The problem is that when writing the name, you do not stop when you got to the end of the "Name" column, but keep writing into the "Money" column overwriting the “100” written here before, so the list will now look like this:

Name:	Money (in $):
Charles Ponzixxxxxxx	x99999999999

Figure 3.3: Your money table with a malicious input

Your table later reminds you that you owe Charles Ponzixxxxxxx roughly 100billion dollars, so you pay him back the amount and clear this person from your list.

Phone terror (DDoS)

Imagine you want to cause some havoc to your very phone-addicted friend, Steve. You want to render his phone useless, but you do not have access to his phone in any way. All you have is his phone number. You are an over the average popular guy, so you ask 1000 of your closest friends to help you out by constantly calling Steve nonstop for an hour. If he answers, hang up and call again, if you do not get through, hang up and call again. Steve's phone will be bombarded with incoming calls, and whatever else he wanted to do with his phone will no longer be possible. If he were waiting for a call from someone, in particular, this person would most likely never be able to get through to Steve at all during that period.

Flu shots (Anti-virus)

Flu shots have some specific traits:

- They only protect against the known or more common viruses
- They are in no way replacements for other preventative actions to diseases
- They are regularly updated and to be effective it is important that they are as current as possible
- When there is an outbreak of a new disease, it will take some time to identify the new threat and develop a vaccine for it

3.3 Evaluation and testing methods

3.3.1 Qualitative interviewing

When conducting interviews, they can either be considered as qualitative or quantitative. Qualitative interviews being interviews not fully structured, and having it more flexibility than its quantitative counterpart. It exists much material on the subject and how to best conduct such interviews. The paper *Qualitative Interview Design: A Practical Guide for Novice Investigators* [26] is a good introduction to this topic and how to approach qualitative interviews. This paper identifies three main categories for qualitative interview design, found in the book *Educational research: An introduction.* [15]. These three categories are the following:

- **Informal Conversational Interview** It the most flexible and informal approach out of the three, relying entirely on spontaneous conversation and generation of questions [15]. The interview does not have any specific questions prepared and simply relies on making questions on the spot depending on how

the conversation is going. It is considered an "off the top of your head" type of interview where questions come to the interviewer as time progress rather than having ready-made questions at hand.

- **General Interview Guide Approach** Is a slightly more structured approach than the Informal Conversational Interview. Here an Interviewer has an idea of what types of questions he wishes to ask but is free to word a question any way he or she wishes depending on the situation. This ensures that somewhat the same area of information is covered in each interview as opposed to informal, conversational interviews where any kind of questions can end up being asked. It is also possible to add follow-up, or probing questions were needed to get even more specific information if necessary.
- **Standardized Open-Ended Interviews** is the most structured and least flexible approach out of the three. It is very specific when it comes to the wording of different questions, making a point of asking them in the exact same way to all interviewees. All participants are asked identical questions, but it is also important to leave the responses to these questions open-ended [15].

[26] also suggests the three stages of conducting a qualitative interview. These are the preparation for the interview, the constructing effective research questions, and the actual implementation of the interview. In the preparation part, one of the three approaches mentioned above should be chosen. Some principals that are essential for the preparation phase are to choose a location with few distractions, explain the purpose and format of the interview, inform approximately how long an interview takes, and do not rely on your memory to recall the answers. When constructing effective research questions, it is important to keep all questions asked unbiased in an attempt to affect answers from the participants of the interview as little as possible. It is also important to have in mind some follow-up questions that might be necessary during an interview to get the best answers possible. For the implementation of interviews, it is important always to stay neutral as an interviewer, only ask one question at a time and providing clear transitions between major topics.

3.3.2 Conducting the evaluation

For this thesis, a standardized open-ended guide approach seems the most suitable as there are very specific questions we want to be answered in our problem description. Adding follow-up questions to some carefully instructed ones for the problem description of this thesis. The goal of the testing is to see if using analogies to introduce beginners to information security will help them understand the concepts better. The tests intend to see whether the test subjects understanding of the material presented to them, is benefited by the use of analogies in the explanation process or not.

The testing of the problem will be done through a qualitative interview process, discussed in the related work chapter, on a few subjects, where each test will be done one person at a time. Each test subject will go through two different versions of introducing the topics, both trying to explain several concepts and themes of information security and hacking. Both versions will present the same topics, but the second one will do so using analogies, rather than ordinary definitions and explanations.

After finishing the first section, a short interview of the test subject will be done, discussing which topics he understood and did not understand. What the test subject knew from before will also be checked when going through the different topics. In both versions, there are several sections with various concepts covered. After the test subject has finished one section, a small discussion about that particular section with a few questions will take place. The template for the general questions asked for each section, as well as some general questions for the whole first part, are added in Appendix B. After going through the first version, without the analogies, the test subjects will start going through the second part where the same concepts are introduced, but *with* the help of analogies. For each of the topics, their understanding of them will again be discussed by asking whether the analogies helped them in understanding the topics better in any way or not. This is done by asking the test subject about their subjective opinion on the matter through discussion. There are also some small tasks to be performed some places in the two versions to see whether something was understandable in addition to the interview process.

These tests will be done incrementally, where a test will be performed on one person, and then the performance of that person will be evaluated, and feedback will be given. This will, in turn, be used to, if needed, revise both the testing method and possibly also the explanations and analogies given to improve the test for later iterations.

Chapter 4
Results

In this thesis, two versions of a document meant for introducing beginners to information security and hacking were made. One version containing traditional definitions and explanations of the topics, and one version introducing the same subjects through different types of analogies. These documents were then used in a test scenario, where participants would go through the two separate versions while being interviewed about their understanding of the different topics, both with and without the analogies. As the themes used for the testing are all introduced in Chapter 2, the test version containing the conventional explanations of the topics will be added in Appendix A while the version using the analogies is introduced in Chapter 3 and reasoned for in Chapter 5. In this chapter, the results of the interview round will be given

4.1 Interview results

The questions asked to the interviewees are added in Appendix B. Here, the results from these interviews will be introduced. The results shown will e further discussed in Chapter 5.

Key concepts:

The general reaction to this part was that all of it was mostly understandable. The concepts found most difficult was usually non-repudiation, least privilege, and integrity.

Most participants found the analogies to be helpful in the understanding even though most of the concepts were understood from part one. This was mainly because the participants felt it helped to put the different concepts into some recognizable context.

Cryptography:

From before everyone had already heard of cryptography or encryption, but only one had heard about, and knew public key encryption. Participants were able to understand the concepts behind ciphers and symmetric encryption, but most of them did not understand how public key encryption worked except for the participant who had already come across asymmetric encryption before.

The padlock as an analogy for ciphers was for most considered a useful analogy to give a clearer picture of how a cipher works, but it was by one participant viewed as not helpful, as he had already understood this and the padlock analogy did not add anything new to him. The analogy for public key encryption was deemed very useful by all the participants of the interview in their understanding of how this worked and was carried out.

Code breaking:

All interviewees understood how frequency analysis worked, but only the participant that already knew about the different attack models felt like he understood them from the explanations given in this part. When trying to rate the attack models from best to worst all but one participant was able to recognize that chosen plaintext or ciphertext attacks were stronger than known plaintext or ciphertext attacks, but had a harder time separating the two different kinds of chosen text attacks from each other.

All participants found the analogy for frequency analysis to be not helpful as the concept already was more than simple enough to explain without it. They all felt that this analogy was redundant when also given a conventional explanation of the technique. The results for the example-analogies given to the attack models were very mixed. For participants not understanding these concepts at all, the examples in the second part did not help to clarify things. However, they did manage to make it clear to one interviewee that attacks where you could choose either the ciphertext or plaintext would be stronger than the models where you could not.

Malware:

The general opinion of the interviewees was that this subject, in the form presented here, did not have any use of the analogies, as the different categories of malware were well enough explained without the analogies, and they did not bring anything new to the table.

Definitions:

As a general opinion, the SQL and Buffer overflow analogies were the ones the majority found the most helpful. Especially the transition from code to something else

in the SQL injection explanation helped the people that had no previous background with code, and only found this part of the explanation confusing. The social engineering analogy did not get the best response, and many of the participants felt like this was one of the least helpful analogies presented.

Overall:

In figure 4.1 the correlation between three questions asked to the participants. The first question asked what concepts they felt they understood the least after the first part of the test, before the analogies. The second question asked which of the analogies helped the most for their understanding of any subject introduced. The third question asked which of the analogies they felt helped the least for their knowledge of any of the subject introduced.

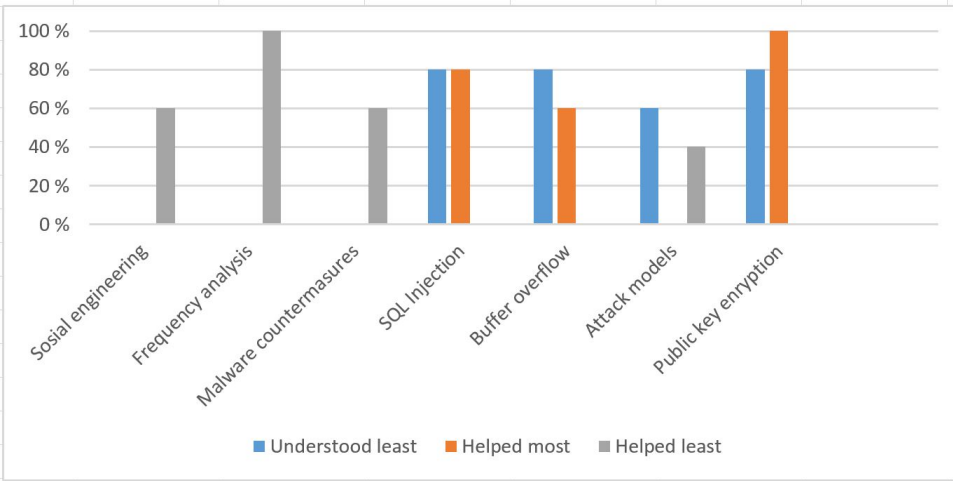

Figure 4.1: Overall questions answered

4.2 Encryption game design

From the interviews conducted, one of the most popular and helpful analogies was the one trying to explain public key cryptography. From this, I chose to design a gamification based problem based on this analogy.

4.2.1 Level design

The problem consists of 4 levels each gradually getting more difficult and exemplifying different concepts of encryption and public key cryptography. The general challenge for each level is to transfer some valuable asset from one person to another, without

it being compromised during transit. All packages sent between the two parties will pass through a third party that wishes to obtain the asset as it is being transferred. This is what the two people sending the asset wants to avoid. Each level all three parties are given some items in their inventory and the player will either get the task of transferring the asset safely or, playing the bad guy, try to obtain the asset without the other parties discovering.

The first level is a very simple one, asking the player to perform the transfer using a symmetric encryption and a shared secret, just to familiarise the user with the design functions.

Figure 4.2: Starting position of level 1, symmetric encryption

The second level will be and attempt to perform a RSA asymmetric encryption to transfer the asset. Here the to "good" parties will not have a previously shared key.

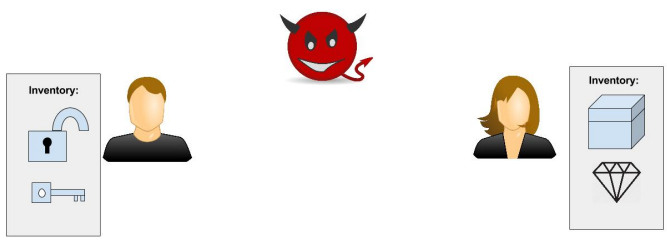

Figure 4.3: Starting position of level 2, RSA encryption

For the third level the player will try to perform a simple Man In the Middle attack on the same RSA transit performed in the previous part.

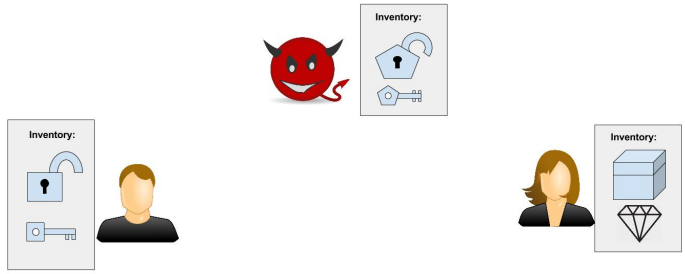

Figure 4.4: Starting position of level 3, MiM on RSA encryption

For the last level, the player will try to perform a dual-key encryption transfer the asset, much like the Diffie-Hellman key exchange the would not be vulnerable to the same MIM attack as the previous level.

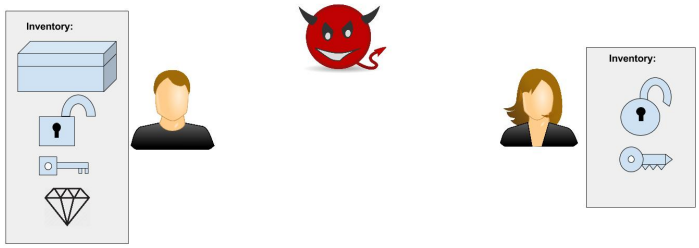

Figure 4.5: Starting position of level 4, Dual key encryption (Diffie-Hellman)

4.2.2 Game mechanics

On level 1, 2 and 4 the player will be given the same challenge, but they will be forced to tackle each level differently depending on what each player has available in their inventory. Figures 4.2, 4.3, 4.4 and 4.5 shows the starting position for each of the levels created and what items each party has in their inventory. You can see that in the first level, the two "good" parties both have the key for the lock, the shared secret. On the second level, however, the receiver does not have the key for the senders lock, so sending a locked box would not help as the recipient would not be able to open it.

The game works by the player being able to match up the different items in the parties inventories before sending them as he chooses. They can be matched in several different ways to perform the challenges. Some examples are shown in

figure 4.6. Every time something is transferred, the bad guy -or eavesdropper- will try to get access to what is being sent and can copy it for himself. This implies that anything sent unlocked or with a lock the eavesdropper had the key to, will be accessible for the eavesdropper. A player will fail a level if the eavesdropper obtains the asset in level 1, 2 and 4. In level 3 the player will lose if he does not manage to retrieve the asset from the other parties.

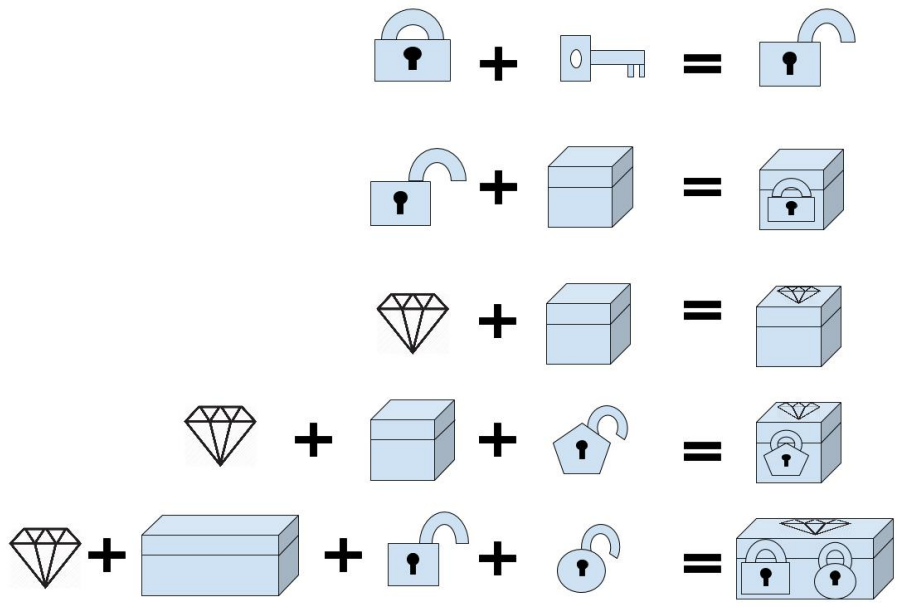

Figure 4.6: Examples of possible combinations of items in inventory

A player must first combine different items in the inventory, then send these items over to the other party. During the sending, the eavesdropper will try to obtain any unsecured items to possibly gain access to the asset, before sending it over to the recipient. Any open locks or keys sent will be copied by the eavesdropper and held in his inventory. There is no limit on how many messages can be sent as long as the asset is not retrieved by the eavesdropper, but the fewer messages, the higher score will be obtained by the player.

4.2.3 Game features

As different people have different abilities to solve the problems presented in this game, a way to adjust the difficulty of the game is preferable. One way this could

be performed is to have the possibilities of hints being given to the user if they are stuck. Such hints could, for instance, be suggestions on what items to combine in the inventory to pass the level or the order of messages that needs to be sent without revealing the content of the different messages. In figures [] the proposed solutions to the levels are presented. By giving the players some, but not all of the information shown here, useful hints can be given without disclosing the full answer to a level. The amount of hints used could also be reflected in the score achieved by the player A way to make the game slightly harder is to fill up the inventory of the parties with items not needed for that specific level that does not open for other solutions. When looking solely at the items handed out in each level, it is possible to obtain some information about exactly how a level is supposed to be done. By adding other random, but useless items in the inventory of parties this information will be obscured, making the levels harder. The eavesdropper can, for example, be handed the key to a lock the sender has so that the sender would have to take into account what locks he should use to ensure security.

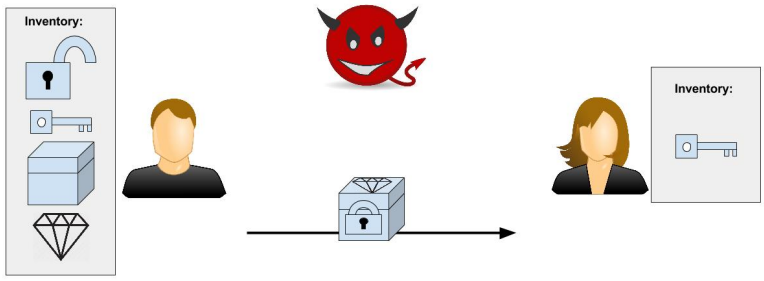

Figure 4.7: Solution to level 1 of the encryption game, suummetric encryption

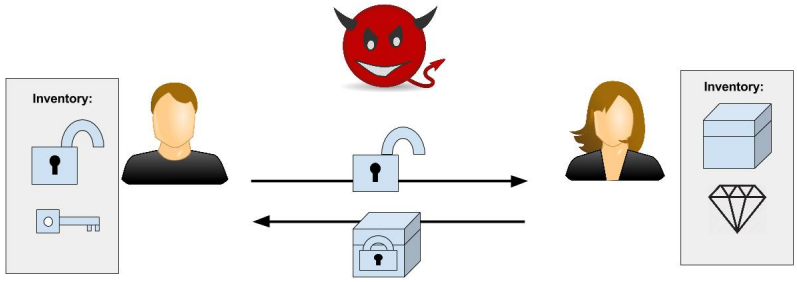

Figure 4.8: Solution to level 2 of the encryption game, RSA encryption

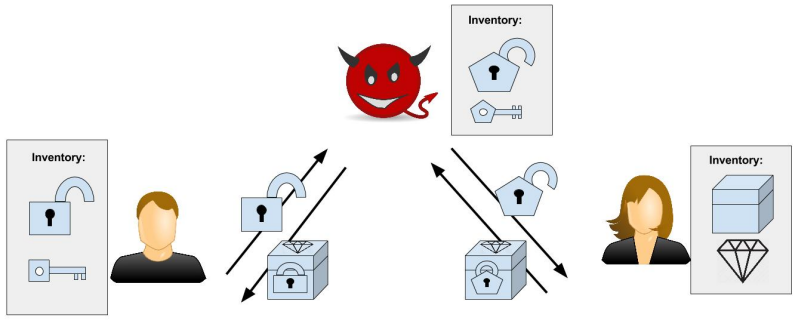

Figure 4.9: Solution to level 3 of the encryption game, MiM on RSA encryption. The eavesdropper copies the senders lock, sends his own lock to the recipient, then retrieves the asset and locks the box with the senders lock

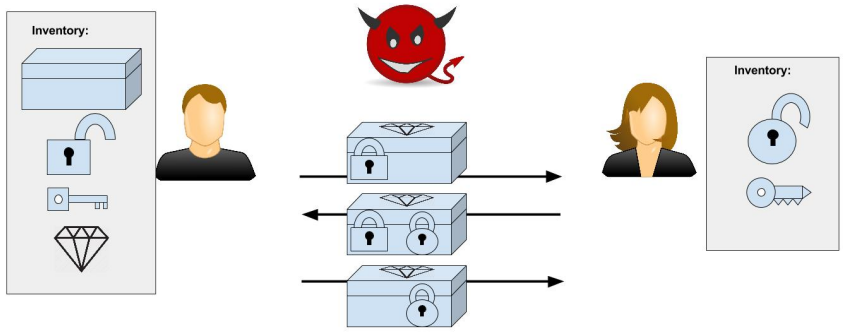

Figure 4.10: Solution to level 4 of the encryption game, dual-key encryption (Diffie-Hellman)

With some kind of score-system, awarding the least amount of messages sent and hints used can be added to a leader-board for player to compare their results. Doing i higher difficulty could also possibly help increase the end score. The competitive aspect of having a leader-board will often increase the motivation for player and ads an extra dimension tho the experience.

4.3 Achievement of goals

Chapter 5
Discussion

5.1 Choice of analogies

In this section, the reasoning behind the different analogies presented in this thesis will be discussed and reasoned for. If there are some limitations regarding some of the analogies, these limitations will also be mentioned, as analogies often tend to simplify concepts to make them more understandable, and certain aspects might get lost in translation. This might lead to some areas of a topic only applying to the actual definition of a subject, and not to the analogy it had been translated to, giving incorrect comparisons when looking more into the details of certain topics.

5.1.1 Key concepts

The key concepts of information security, confidentiality, integrity, availability, authorization, authentication, non-repudiation and least privilege, can be viewed as criteria for a computer system to maintain secure. It would, therefore, be logical to transfer these concepts over in the analog world by looking at more commonly known entities which also includes a lot of these concepts in their infrastructure. A hospital was a natural choice here, as concepts such as confidentiality, integrity, and availability are critical in health care today, and a lot more familiar to the layman than system security. The security of a bank was also considered, but concepts such as availability and confidentiality seemed more applicable and logical when talking about hospitals rather than banks. All the different concepts would quite seamlessly be translated over to a hospital without having to alter the definitions in any noticeable way. It is, however, important to mention that this is in no way a 1:1 translatable analogy, as the hospital analogy, only gives some examples of the different concepts but can not in itself stand as a definition of the different concepts. To explain this with another analogy, a thumb is definitely a finger, but a finger is not necessarily a thumb.

5.1.2 Cryptography

For cryptography, the classical and well-known analogy viewing ciphers as padlocks with combination keys was used. This gives a good example on how the same cipher can be used over and over as long as the key is changed, making it easy to understand that knowing what cipher is used, does not give away the encrypted information. Using combination keys also gets over the point that breaking the cipher by guessing, or brute-forcing, the key is possible but not at all practical. Padlocks also give a great example of how a one-way function works, easy in one direction, hard in the other. This is also the foundation of how ciphers work, utilizing one-way mathematical functions that are easy to compute in one direction but hard the other way around. An example of such a one-way function is the discrete logarithm problem used for instance in the Diffie-Hellman key exchange [6].

The second analogy used for cryptography aimed at explaining the concept of public key encryption. This is not as straight forward as normal symmetric encryption, and can be hard to wrap your head around when first introduced to it. An analogy using multiple padlocks locking a single box to ensure privacy was used to explain this. This explanation is a way of explaining the Diffie-Hellman key exchange without introducing all the math behind the concept, as this might become overwhelming for people without previous knowledge in the area. This analogy makes it easier to see how the eavesdropper, Eve, is not able to intercept and retrieve the private communication between Alice and Bob, even though they do not have any shared encryption scheme already established. With this analogy, explaining the concept of a man in the middle attack, where Eve puts her own locks on the box when the box is being transferred from Alice to Bob and back again, is also simple to explain. The concept of public key encryption is very well translated over to the physical world with this analogy but does not give any detail into how the actual math behind creating digital locks is done.

5.1.3 Code breaking

For the code-breaking section of the tests, two subjects were covered, frequency analysis and attack models. The analogy for the frequency analysis is a rather simple one where the letters in the alphabet are exchanged with the ingredients of a recipe. Not much differs between the normal definition of frequency analysis and the analogy made in this thesis.

For attack models, simple examples of specific scenarios for the different types of attack models were made. These serve as examples for a better understanding of the concept by putting the conventional definitions into context, to make it all more understandable. These examples are in themselves, not an attempt on defining the

concepts using analogies, but rather a way of better visualizing how the concepts of attack models would work in practice.

5.1.4 Malware countermeasures

Malware countermeasures are all about protecting your asset, that being your computer, against intruders. A fitting analogy for malware countermeasures could, therefore, be switching out the computer with your house, then implementing different security measures that would prevent intruders into said house, rather than protect your computer against malware. Any visitors to the house would be a representation of traffic coming to and from your computer through the Internet. There is however not any good analogies for malware hiding on your computer, other than equipment such as hidden cameras and microphones being a type of spyware secretly installed in your house. The different analogies succeed in portraying the general concept of how the different analogies work but do not hold up if one wants to look more into the details of each countermeasure. The analogies are only meant to grasp the general concept behind each countermeasure, but other material is needed for the specifics.

5.1.5 Password strength

On the subject of password strength the main idea is to make it as hard as possible for an attacker to be able to break your password. To make this as hard as possible there are a few things to remember when creating the password so that it will be as strong as possible. The analogy chosen to explain this tries to show how different decisions when making a password will affect how hard it will be for an attacker to guess it. As passwords are usually broken through brute-force attacks, this is represented by having the participant in the shooting competition being blindfolded, shooting in the dark. The analogy also tries to give reasoning to why choosing characters other than the standard alphabet and changing your passwords frequently will make it even harder for an attacker to break it.

5.1.6 Social engineering

Social engineering is not a very technical topic to explain so for the analogy the well-known story of Rapunzel is used as an example of how social engineering is used in all kinds of scenarios and to show how even the strongest of security measures can be broken by manipulating the people involved in maintaining the security implemented. The analogy does not explain how social engineering is performed but shows an example of how it can work in almost any setting thinkable to work around different types of security measures put in place, both digital and physical.

5.1.7 SQL injection

SQL injections can be a tricky concept to explain to people with no background in computer science or information security, as its explanation usually involves a bit of example code. Even though the SQL language is fairly similar to the English language at first glance, it can often lead to much confusion for people with no coding background.

Because of this, the main goal for this analogy was to find a way to explain the concept of an SQL injection without having to resort to using code in the explanation. The robot is used to justify the fact that computers have no sense of critical thinking other that what has been implemented by the programmer. A person reading the malformed request form would be a lot more skeptical and probably understand that this was foul play. With the analogy used, a simple example of how SQL injections work is being introduced, without any direct code involved, showing how an attacker can take advantage of poorly designed systems that can lead to disastrous results.

5.1.8 Buffer overflow

A buffer overflow exploit is a very technical and complicated attack that require a good understanding of how a computer works on an extremely detailed level.Trying to explain computer stacks, heaps memory allocation, and return addresses to the layman will most likely result in more confusion and resignation than anything else. It is thus desirable to have the analogy simple enough for anyone to understand, but also detailed enough to give a reasonable explanation of how buffer overflows actually occurs and can be exploited.

The analogy made aims at explaining the dangers imposed by the possibility of overwriting fields in memory when writing something to memory, passing over to data not intended to be altered by that input. It shows a simplified version of how this would unfold in a computer and what implications such vulnerabilities might have. It does this without going into any details of exactly how this is possible in a real situation, as this would seem too complicated for anyone without experience in coding and information security.

5.1.9 DDoS

Dynamic Denial of Service attacks an analogy involving flooding a person's phone with incoming calls from several other phones. This was to show how any system can be broken down and rendered useless by occupying all of the system's, or in this case, phone's, resources. As of the accuracy, this analogy is pretty much exactly how normal DDoS attacks would work, changing the calls with for example a SYN flood to a target system.

5.1.10 Anti-virus

As anti-viruses are a type of signature-based detection system, which is already covered in an earlier section about malware countermeasures, this analogy is made to explain in what way anti-viruses are protecting your computer, and how they in many ways do not. It is made to show that having anti-virus in itself is not a full protection against malware, only one of several precautions you can take to stay safe. This is shown by comparing anti-virus to regular flu shots, they give a certain amount of protection, but does not justify reckless behavior.

5.2 Advantages and disadvantages of analogies

Analogies are used to make the unfamiliar familiar. They great tools for introducing new, and unknown concepts to people by comparing them to other domains to make sense of it. Connecting new information to existing information makes it easier to understand, as similarities can help one generalize certain aspects to a greater extent. Therefore, analogies can be immensely helpful introducing new concepts, pointing out similarities between the new information and the real world. I find myself unconsciously using analogies in many situations when trying to wrap my head around new concepts by comparing them to areas that are already familiar to me.

There are also risks connected to the use of analogies, however. As analogies often tries to simplify information they are never based on an exact one to one fit between analog and target. There will always be some features of the analogy that differ from its target, and these features might mislead. This might lead to analogies causing just as much harm as they are helpful. To use an analogy to explain this, an analogy can be seen as a double-edged sword. If we take the analogy for buffer overflow as an example, this analogy was found very helpful for most participants in the interviews for giving a simple explanation on how this exploitation worked, but this analogy is not able to translate all aspects of a buffer overflow. When looking more into the details of the exploit the analogy used does no longer apply and, if not careful, this can lead to misunderstandings and simplifications that are simply wrong. This shows that it is important to be aware of the limitations of the analogies used in the learning process so that the analogies are only applied where they are appropriate and not everywhere else. An analogy usually only aim at aiding in a specific area of its target's domain. There is also a risk that a person has misconceptions on how the analogical domain is working, and these misconceptions can then be transferred over to the target domain.

5.3 What the interview results tell us

The interviews showed that the appreciation of the different analogies highly correlated with what topics participants had the most problems understanding with just an ordinary explanation. Analogies used to explain topics considered to be simple by the participant, were, in general, not viewed as helpful. It was when the subjects introduced became more difficult the interviewees found them to be the most useful overall. What was considered to be difficult, however, was different for the participants, making some analogies work better with specific interviewees.

Chapter 6
Conclusion

From the research done on the security analogies in this thesis, it was clear that analogies only served a purpose when concepts and topics were considered hard to understand by an individual. For information viewed as straightforward by a person, the analogy only seemed redundant without adding anything to the understanding of said subject. When the topics got more complicated, however, the interviewees tested in this thesis, found the analogies to aid them in their understanding of topics they could not entirely grasp from just an ordinary explanation of the same topic. This thesis has shown that there are many possibilities for analogies in the field of information security and hacking that would aid a layman to a better understanding of these fields. It also shows how analogies are not to be used as a teaching tool in all scenarios, as many concepts do not gain from the use of certain analogies. As with many other things, analogies are to be used sparingly and only where it is needed.

When good analogies are found, they do have a good possibility to be used in different games as an introductory tool to information security. As many find computer science and code quite intimidating, using games based off of analogies might be a good way of softening the blow when it comes to computer security and hacking.

Good analogies are also a sound basis for creating interesting teaching games as they can put otherwise complicated and specific topics into an environment more familiar to people without any background in computer science. This again leads to a less intimidating basis for newcomers as many concepts of information security might not be the easiest for the layman to wrap his or her head around.

6.1 further work

In this thesis, preliminary research on the use of analogies in games used for the introductory teaching of information security has been made. One example game revolving around the topic of public key encryption has also been designed. The next step now is to design more challenges to various subjects so that a complete introductory tool can be made. As this thesis has concluded that analogies are not always the best choice, new games or challenges does not necessarily have to revolve around analogies if they are not found appropriate.

These games would also have to be implemented in some way as for instance a website. Further gamification elements can then be added like badges for completed topics, progression indicators, and appointment dynamics to motivate users for further use of the tool to mention a few.

References

[1] Paul C. van Oorschot & Scott A. Vanstone Alfred J.Menezes. *Handbook of Applied Cryptography*, chapter 1. CRC Press, 1996.

[2] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, chapter 5. WILEY, 2001.

[3] Lynn Erla Beegle. *Rootkits and Their Effects on Information Security.* Taylor & Francis, Inc, 2007.

[4] Fred Cohen. *Computer Viruses.* 1986.

[5] ComputeWeekly. Nikeplus, 2014. http://www.computerweekly.com/blogs/inspect-a-gadget/nikeplus1.jpg.

[6] Bert den Boer. Diffie-hellman is as strong as descrete primes for certain primes, 1988.

[7] Chad Dougherty. *Practical Identification of SQL Injection Vulnerabilities.* US-CERT, 2011.

[8] EdTeckReview. What is game based learning, 2013. http://edtechreview.in/dictionary/298-what-is-game-based-learning.

[9] Sari Stern Greene. *Security Policies and Procedures: Principles and Practices*, chapter 3. Pearson Prentice Hall, 2006.

[10] Chris Hadnagy. The official social engineering portal, 2016. http://www.social-engineer.org/g.

[11] Stephen Hernandez. *Official (ISC)2 Guide to the CISSP CBK*, chapter 5: Cryptography, pages 866–885. Auerbach Publications, 3 edition, 2006.

[12] Stephen Hernandez. *Official (ISC)2 Guide to the CISSP CBK*, chapter 4: Software Development Security, pages 709–720. Auerbach Publications, 3 edition, 2006.

[13] Stephen Hernandez. *Official (ISC)2 Guide to the CISSP CBK*, chapter 2: SecurityNetwork Architecture & Design, pages 347–348. Auerbach Publications, 3 edition, 2006.

[14] Kevin Jackson. *As easy as ABC.* Independent, 1994. http://www.independent.co.uk/arts-entertainment/as-easy-as-abc-georges-perec-once-wrote-a-whole-novel-without-the-letter-e-kevin-jackson-summarises-1420994.html.

[15] Walter Borg Meredith Gall, Joyce Gall. *Educational research: An introduction.* Allyn and Bacon, 2003.

[16] Minecraft.net. Minecraft, 2016. https://minecraft.net/.

[17] Rachna Dhamija & Jens Grossklags Nathaniel Good. *Stopping spyware at the gate: a user study of privacy, notice and spyware.* ACM New York, 2005.

[18] Encyclopedia of Mathematics. Law of large numbers, 2012. https://www.encyclopediaofmath.org/index.php/Law_of_large_numbersy.

[19] Elias Levy (Aleph One). *Smashing The Stack For Fun And Profit.* Phrack, 1996.

[20] Oracle. Gamification guidelines, 2014. http://www.oracle.com/webfolder/ux/applications/uxd/assets/sites/gamification/phase_3.html.

[21] Saatchi. Gamification for business, barnds, and loyalty, 2011. http://www.slideshare.net/Saatchi_S/gamification-study.

[22] Jerome H. Saltzer & Michael D. Schroedinger. *The Protection of Information in Computer Systems*, chapter I.A. Y, 1975.

[23] Social-engineer.org. The social engineering infographic, 2014. http://www.social-engineer.org/social-engineering/social-engineering-infographic/.

[24] Symantec. Symantec intelligence report, 2015. http://www.symantec.com/content/en/us/enterprise/other_resources/intelligence-report-06-2015.en-us.pdf.

[25] theanalogiesproject.org/. The analogies project, 2016.

[26] Daniel W. Turner. Qualitative interview design: A practical guide for novice investigators, 2010.

Appendix

Appendix A

Explanation of the key concepts given to the participants of the interviews

Basics of cyber security & Hacking

There have been written countless books on the subjects of information security and hacking over the last decade, and the field seems to be growing larger for every year as the subjects are rather new and still in continuous development. It is therefore no short simple introduction to get a full view over the entirety of information security and hacking. In this test you will be introduced to some important concepts and examples of how computers, and the information they contain, are being both protected and attacked today.

Key concepts

There are a few key concepts forming the foundation for information security that are important to understand and are considered some of the most crucial components of security in computer systems. They are refereed to as the CIA triad, which stands for Confidentiality, Integrity and Availability, and a system should include all these concepts in their design to be considered to be secure. Other concepts such as non-repudiation and authentication however, are also important to have in mind when trying to design and construct secure systems.

Confidentiality: One can look at confidentiality as the privacy of information. The goal of confidentiality is the protection of information or data from disclosure to unauthorized people.

Integrity: Information integrity refers to the protection from unauthorized modification of data. It is often very important to be sure that only authorized parties have been modifying information as this information is only of any value if it

is correct. At first glance it might seem like confidentiality also ensures Integrity, but this is not always the case as it is possible to alter information without having the access to it. Another important trade of integrity is that if some information were to be altered, this alteration would be detectable.

Availability: Availability of information refers to the ability of authorized parties to access this information when they need to. Any information however important and secure it is, will be of no use to anyone if no one can access it when they need it.

Authorization & authentication: Authentication is proving that a person or party is who they are claiming to be. Authorization is checking if this person or party has the rights to perform a given action such as running a program or altering a file. Authentication and authorization are very connected as parties must be authenticated to be able to perform tasks they are authorized for.

Non-repudiation: Non-repudiation is an assurance that a party in any form of transaction between two parties can not deny either sending or receiving some data. If a party performs some action, it is impossible for that party to later deny performing the action in question.

Least privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete a certain job.

Cryptography

A way to implement some of these attributes mentioned above is to protect data using encryption. Encryption will convert data into another form that will be unreadable for people not in possession of the encryption key to decrypt this message. Encryption is done with the use of something called ciphers.

Ciphers: Ciphers are algorithms used to encrypt and decrypt messages. When you use a cipher to encrypt a message, often called a plaintext, you will output the ciphertext of said plaintext. This ciphertext will without the correct key to the cipher be close to impossible to convert back to the original message depending on the strength of the cipher used. One of the earliest ciphers known is called the caesar cipher, it is said that it was used by Julius Caesar to communicate with his generals on the battlefield. In this way untrusted messenger delivering the messages to the enemy was not a concern as the message would be unreadable to anyone but the generals of Caesar. His cipher was very simple, it works by substituting a letter in the message with the letter 3 steps to the right of it in the latin alphabet. So the message “attack at dusk” would translate into “dwwdfn dw gxvn”. Here the cipher would be the substitution of letters elsewhere in the alphabet while the key for that

cipher would be 3, letting you know that the letter have been changed with the letter three steps to the right of it in the alphabet.

Ciphers have changed a lot in the modern days with the introduction of computers, so there is often made a distinction between modern and classical ciphers. Modern ciphers take advantage of the powerful computing power of today's computers to make ciphers much harder to break than for example the caesar cipher.

Public key encryption: Another problem with many ciphers, is that they are dependent on a "shared secret" between the parties that wish to communicate. To use the caesar cipher as an example the person decrypting the message, would also need to know how many steps in the alphabet the text has been shifted so he can reverse the encryption back to its original form. This would imply a private exchange of this key before hand, but that is not possible on for example the Internet. A cipher dependent on a shared key is called a symmetric cipher. There is, however, another type of encryption, called asymmetric encryption. Asymmetric encryption does not have the same key for encryption and decryption. Instead it relies on having a pair of two different keys. One private and one public. Because of this, asymmetric encryption is also known as public key cryptography. The private key is, as the name states, not to be shared with anyone. But the public key can be given to anyone without compromising the secrecy of any messages you send. An important trait of public key cryptography is the fact that encrypted communication can be achieved without having to first exchange a secret key. If you encrypt some message with someone's public key, only the private key will be able to decrypt that message. This works by having parties disclose their public key to everyone, hence public, and if someone wishes to communicate privately with this party, they will encrypt their message with the other party's public key. Now only the party with the private key matching this public key will be able to decrypt the given message.

Code breaking: In this section we will look closer into how encryptions and codes meant to keep information private and secure can be broken. The reason why this is possible is because all ciphers in some way give away some information about the the data it originated from. This is easiest to see when looking at simple ciphers, but it also applies to modern ciphers used in online communication today.

Frequency analysis can be tool to help break certain ciphers. This analysis looks at how often different letters appeared in the ciphertext. This may give indications to what the original text might have been. Taking the caesar cipher as an example again, when looking at the frequency of each letter in the ciphertext, knowing for example that "e" is by far the most used letter in the English alphabet, one can simply find out what letter is most common in the ciphertext and assume that this letter is most likely represented as an "e" in the original message.

Attack models: When trying to break an encryption, the attacker may have different levels of access to the encryption mechanism. These different levels of access are called attack models, each giving the attacker a different amount of information to work with. The more elaborate access the attacker has, the more useful information he is able to get. Here we will mention the four most common attack models:

Chosen plaintext attack: The attacker is able to choose any plaintext and get to know the corresponding ciphertext through some kind of encryption oracle which is able to produce the ciphertext for any plaintext it is given without revealing the key to the encryption.

Known ciphertext attack: The attacker is in the possession of some ciphertext, but does not know what the corresponding plaintext is. He does not choose the ciphertext obtained and is not able to produce any more of it.

Chosen ciphertext attack: Attacker is able to choose any ciphertext and obtain the corresponding plaintext to this ciphertext.

Known plaintext attack: The attacker is in the possession of some plaintext and it encoded ciphertext. He did not chose any specific text himself and does not know what the text is about. He is also not able to get anymore plaintext-ciphertext pairs.

Malware countermeasures:

The word malware comes from the merging of the words malicious and software. This is a term that describes all kind of software that made with malicious intentions. Examples of such software are computer viruses and trojan horses. These programs can invade your privacy and do a lot of harm to a computer, and it is therefore important to protect computers against this threat. In this section the main categories of malware protection are introduced and explained.

Signature based detection: Signature based detection is a device or software that monitors network or system activities for malicious activities, by looking after code on the machine resembling already known malicious code.

Sandboxes: Sandboxing isolates programs from the rest of a system, preventing malicious or malfunctioning programs from damaging or snooping on the rest of your computer. It provides strictly controlled resources for guest programs on a computer so that potentially malicious code can not cause any harm to your computer when only run inside the sandbox.

Behavior based detection: Behavior based detection is a device or software that monitors network or system activities for malicious activities, looking for

activities normally associated to different types of malware or any other activity on the system deviating from what is considered to be normal for that system.

Security awareness training: Security awareness training a general term for educating people about computer security.

Firewalls: A firewall is a network security system, either hardware- or software-based, that controls incoming and outgoing network traffic based on a set of rules. Acting as a barrier between a trusted network and other untrusted networks - such as the Internet - a firewall controls access to the resources of a network through a positive control model. This means that the only traffic allowed onto the network is the traffic following the policies defined by the firewall; all other traffic is denied. Firewalls differ from intrusion detection systems in that they only look "outwards" for possible threats, as opposed to behavior based- and signature based detection systems, that look for threats already in the system.

Definitions

In this section definitions to some techniques to prevent your data or computer to be compromised as well as some attacks on computer systems that can potentially harm your system if not careful.

Password strength: The strength of password relates to the amount of different possibilities there is for the password as well as its randomness/entropy. If someone wants to break a password they would have to guess what that password is in some way, so the more possibilities there are to guess, the harder it would be to break it. Making a password longer is one way of increasing the amount of possible keys, also the use of more than just standard characters will have this effect. Not using well known words will also strengthen a password as an attacker would have to try combinations of letters that is not in any dictionary, making more guesses required. Passwords are also strengthened by replacing them once in awhile so that an attacker cannot rule out earlier failed guesses when trying to crack a password.

Anti Virus: Everyone is familiar with anti virus and that is prevents viruses and other malicious code from infecting your computer. But how they do that is something few people bother to look into. Anti-virus software works by having a database of already known viruses. It then scans all the files on the computer and compares the code in the files on the computer, to the known types of viruses collected in its database. This implies that anti-virus software can only protect against malware it is already familiar with, and will not work against new types of malware before this malware is discovered and added to the anti-virus software's database of malicious code. As anti-virus software gets more advanced, they also have the capability to identify malware and viruses by looking at how suspicious

code is behaving, not only if it is similar to some malware in its database. So even if the code might seem safe comparing it to other malware, it might still be flagged as malware based on how it is behaving on the computer.

SQL injections: Most websites has its data stored in databases which it communicates with to extract certain information from this database. The programming language used to communicate with databases is called SQL. With this language you have the power to both look up in the database and modify it in any way you see fit. There are commands for looking up and extracting certain information, and there is commands able to alter the database, including adding and deleting data. Websites use database for all sorts of different things, one example being to store different users and their passwords for that website. The language is quite similar to the English language when written, so command can often be understood even if one is not familiar with the language.

Lets say that when the website checks for matching passwords to a username, it directly inserts the text written by users in the "user:" and "password:" fields in their website, into a SQL command directly. This command will look something like this:

SELECT * **FROM** users **WHERE** email = "input from user" **AND** pass = "input from user"

This will return the profile of the user written into the user-field if the password written into the password-field matches the one in their databases, if not, access will not be granted. The words in capital letters are commands the SQL language wishes to perform on the database and the red text is input given to SQL from the user.

Here is an example where user@email.com has been written into the user-field and password has been written into the password-field:

SELECT * **FROM** users **WHERE** email = 'user@email.com' **AND** pass = 'password'

Here SQL will get the profile of the user user@email.com if the password matches the password stored in their databases. Password is not the correct password, so the login will not be successful.

SQL injection works by manipulating the input a user can give to the SQL language through the two input-fields "User:" and "Password:" Notice how the SQL statement above adds quotation marks to the parameters given by a user. After the quotation marks, SQL commands are given that are used to talk to the database. In this way a user can for example input the password password" DROP table accounts– which would delete all accounts in that database if there was a table there happened

to be called “accounts”. This happens because you have inserted an extra quotation mark, making SQL think that the input is done and a command is coming next, this command would in this case be to drop, or delete, the table called “accounts” in the database. It would look something like this:

SELECT * FROM users **WHERE** email = “user@email.com” **AND** pass = “password” **DROP** table accounts–”

Buffer overflow: An exploit called a buffer overflow happens when a buffer, a space in the computer’s memory, is passed more data than it is intended for. A buffer, or buffer memory, is a portion of a computer’s memory that is temporary set aside as a holding place for data. The data will then continue to overwrite other parts of the adjacent memory which is not intended. This vulnerability usually happens when input from untrusted users are being processed by a program. If the data received is bigger than the buffer the data is allocated, it will overwrite adjacent memory locations with that data. The consequences are unpredictable when this happens, but savvy computer enthusiasts can also construct their input data carefully to run his own code or alter the way the program was intended to be used. This is done by filling up the buffer of memory allocated to this input and “overflow” that buffer, so that the rest of the data will overwrite other crucial parts of the computer’s memory and possibly run malicious code that a user has placed in his input. This technique is a lot more advanced than what is explained here, but requires some more insight into how the memory of a computer is built.

Social engineering: Social engineering is the act of getting other people the information you seek usually by gaining their trust in some way. It aims to attack on what is usually a organisations weakest security link, its people. Even extremely secure systems can be compromised if the users of that systems are tricked in some way by an attacker. An unbreakable password, is no use if someone is able to trick the person who knows the password into revealing it. Some categories of social engineering are phishing, vishing and impersonation. Phishing is the method of using emails appearing to be from legitimate sources to gain influence or personal information, vishing involves gaining information or trying to influence people’s actions over the phone, while impersonation is the practice of presenting oneself as someone else to obtain private information or access to a person, company or computer system.

DDoS In a distributed denial-of-service (DDoS) attack a victim web page is being flooded with illegitimate traffic to their online services, causing them to become unavailable for legitimate users trying to gain access to that particular web page. The attack can be performed at as good as all web pages, from banks to media websites, and there is no fault proof way of defending against it other that being

able to handle all the traffic that comes your way. DDoS is a type of DoS attack but differs because a normal DoS attack usually stems from a single source, trying to flood the system where as a DDoS attack comes from multiple computers and internet connections, making it harder to block it of.

Appendix

Appendix B

Interview template

Key concepts:

Part 1:

- *Have you learned about/ heard of these concepts before?*
- *Was the concepts understandable from what you read?*

Part 2:

Match the analogy to key concept:

- *Did the analogies in any way help for your understanding of the topic(s)?*

Cryptography:

Part 1:

- *Have you learned about/ heard of these concepts before?*
- *Was the concepts understandable or not from what you read?*
- *What would you say is the fastest of asymmetric(public) and symmetric key encryption?:*

Part 2:

- *Did the analogies in any way help for your understanding of the topic(s)?*

Code breaking:

Part 1:

Order attack models form best to worst:

- *Have you learned about/ heard of these concepts before?*
- *Was the concepts understandable from what you read?*

Part 2:

- *Did the analogies in any way help for your understanding of the topic(s)?*

Malware:

Part 1:

- *Have you learned about/ heard of these concepts before?*
- *Was the concepts understandable from what you read?*

Part 2:

Match-up results:

- *Did the analogies in any way help for your understanding of the topic(s)?*

Definitions:

Part 1:

- *Have you learned about/ heard of these concepts before?*
- *Was the concepts understandable from what you read?*

Part 2:

Match-up results:

- *Did the analogies in any way help for your understanding of the topic(s)?*

Overall:

- *What subjects did you understand the least after part 1?:*
- *For what subjects did you feel that the analogies in part two helped the most?:*
- *For what subjects did you feel that the analogies in part two helped the least?:*
- *How would you compare the two different parts in regards to understanding the different topics presented by them?:*