Master Thesis

Weaponize personnel against social engineering attacks via gamified active inoculation

Jean-Philippe O. Kok

Cyber Security Specialisation
Computer Science
Faculty of Electrical Engineering, Mathematics & Computer Science
EEMCS
Delft University of Technology
The Netherlands
Student Number: 4953193
14-04-2021

Preface

Abstract

The number of cybercrime cases rises rapidly, and the type of crime takes more and more diverse forms. However, the protection against these risks lag behind and becomes quickly outdated. This thesis follows the Fake News Game example using active inoculation in the form of a game against social engineering risks [70]. Inoculation draws the analogy with vaccines and says humans can be injected with small pieces of persuasion to trigger the development of antibodies against that persuasion, similar to how vaccines protect humans against diseases. The player is placed in a social engineer's shoes and learns six often used psychological techniques in social engineering attacks in the game. These techniques are shown in short, interactive sections, where the player experiences how these techniques feel. This way, the body learns how to recognize these techniques and develop its antibodies against them, thus learning how to protect against them when they are used for real. The game is built to be flexible and modular. The flexible and modular setup of the game allows for adjustment to the target audience. This way, it can also keep up with the rapidly changing developments within cybercrime. The intervention was tested within the Dutch Armed Forces in a three-group pretest-posttest quasi-experiment. The experiment showed no evidence the intervention was successful in raising resilience against social engineering attacks. However, the intervention data shows evidence that the intervention is an effective way of raising resilience against social engineering risks.

Keywords: social engineering, inoculation theory, awareness interventions.

Thesis committee

Delft supervisor: Dr.ir. Verwer
Second chair: Dr.ir. Brinkman
Twente supervisor: Prof. Dr. Junger

Acknowledgements

Over the past twelve months, I have received support and help from many different people without whom this thesis would probably never have been completed. First of all, I would like to thank my supervisor Prof. Junger, who believed in the idea for this thesis from the beginning and whose expertise brought this work to a higher level. I also would like to thank my other supervisor Dr. Verwer, who accepted me as his graduate student, even though this work is not directly in his line of work. Without him, this thesis would have never got off the ground.

Secondly, I would like to acknowledge my colleagues from my internship at the Dutch Armed Forces. Even though most work was done from home, they still did their best to help and guide me with this work. I would also like to thank everyone at the experiment location for their help and their willingness to participate in this experiment.

Finally, I would like to thank my family and friends for pulling me through the past twelve months. I would like to thank everyone who helped play-test in the game's early alpha and beta phases and help proofread the thesis. A special thanks to my parents, who supported me in my studies from the beginning all the way until the end and my girlfriend, who had to live with me while I was busy being grumpy with my thesis.

Philippe Kok - Utrecht - 14-04-2021

Contents

Acronyms **8**

1 Introduction **9**
1.1 Introduction . . . 9
1.2 Research Question . . . 10

2 Theoretical Framework **12**
2.1 Cyber Security . . . 12
2.1.1 Definition of Cyber Space . . . 12
2.1.2 ICT Security vs Information Security vs Cyber Security . . . 12
2.1.3 Human Factors in Cyber Security . . . 14
2.2 Social Engineering . . . 15
2.2.1 Definition . . . 15
2.2.2 Types of Social Engineering Attacks . . . 15
2.2.3 Social Engineering Techniques . . . 16
2.3 Behavior change . . . 18
2.3.1 Awareness Interventions . . . 18
2.3.2 Behavioral Change Theories . . . 18
2.4 Inoculation Theory . . . 20
2.4.1 Threat and Counterarguing . . . 21
2.4.2 Contemporary inoculation theory . . . 21
2.4.3 Meta-analysis . . . 22
2.4.4 Inoculation Theory and Fake News . . . 22
2.4.5 Inoculation Theory and Cyber Security . . . 23
2.4.6 Inoculation Theory and Social Engineering . . . 23

3 Social Engineering Attack Scenarios and Real-life Examples **24**
3.1 Introduction . . . 24
3.2 Generic Scenarios . . . 24
3.2.1 Dumpster diving . . . 24
3.2.2 Misleading the cleaner . . . 24
3.2.3 Pretending to be a cleaner . . . 24
3.2.4 Building a trust relationship . . . 25
3.2.5 Tailgating into the building . . . 25
3.2.6 Pretending to be the network administrator . . . 25
3.2.7 Call from the bank . . . 25
3.2.8 USB sticks . . . 25
3.2.9 QR codes . . . 26
3.3 Real-life Examples . . . 26
3.3.1 Undercover in Nederland (Dutch TV-show) 1 - 2008 . . . 26
3.3.2 Undercover in Nederland (Dutch TV-show) 2 - 2016 . . . 26
3.3.3 Undercover in Nederland (Dutch TV-show) 3 - 2018 . . . 26
3.3.4 Streetlab (Dutch TV-show) - 2015 . . . 27
3.3.5 419 Scam or "Nigerian scam". . . . 27
3.3.6 Sextortion phishing . . . 27
3.3.7 The dishonest programmer . . . 27
3.3.8 Mitnick - Hacking into the feds. . . . 28
3.3.9 RSA . . . 28
3.3.10 Operation "Red October" . . . 28

3.3.11 Virtual kidnapping . . . 29
3.3.12 Malware parking tickets . . . 29
3.3.13 Francophoned . . . 29
3.3.14 CEO fraud . . . 29
3.3.15 Microsoft Tech Scam . . . 30
3.3.16 Whatsapp Fraud . . . 30
3.3.17 SMS Fraud . . . 30

4 Design **31**
4.1 Inoculation Theory and the Social Engineering Simulator . . . 31
4.1.1 Introduction . . . 31
4.1.2 Passive Inoculation . . . 31
4.1.3 Active Inoculation . . . 31
4.2 Game Flow . . . 32
4.3 Scenario Design . . . 32
4.3.1 Part 0 . . . 32
4.3.2 Part 1 . . . 33
4.3.3 Part 2 . . . 34
4.3.4 Part 3 . . . 36
4.4 Scenario Editor . . . 37
4.5 Normal Nodes . . . 38
4.5.1 Story Node . . . 38
4.5.2 Image Node . . . 38
4.5.3 Input Node . . . 39
4.5.4 Conversation Node . . . 39
4.5.5 Whatsapp Node . . . 40
4.5.6 Mail Node . . . 40
4.6 Principles Nodes . . . 40
4.6.1 Principle of Time Node . . . 40
4.6.2 Principle of Need and Greed Node . . . 42
4.6.3 Principle of Distraction Node . . . 42
4.6.4 Herd Node . . . 44
4.7 Technical Nodes . . . 45
4.7.1 End Node . . . 45
4.7.2 End Screen Node . . . 46
4.7.3 Achievement Earned Node . . . 46
4.7.4 GDPR Node . . . 47
4.7.5 Property Changer Node . . . 47
4.7.6 Score Changer Node . . . 48
4.7.7 Credibility Changer Node . . . 48
4.7.8 Conditional Node . . . 48
4.8 Technical Design . . . 49
4.8.1 Unity . . . 49
4.8.2 Scene . . . 49
4.8.3 Scripts . . . 49
4.8.4 Data Model . . . 52

5 Method **53**
5.1 Experiment Design . . . 53
5.1.1 Participants . . . 53
5.1.2 Pretest . . . 53
5.1.3 Playing The Game . . . 53
5.1.4 Posttest . . . 54
5.1.5 Debrief . . . 54

6 Results **55**
6.1 Results From The Experiment . . . 55
6.1.1 Pretest . . . 55
6.1.2 Posttest . . . 55
6.2 Results From The Game . . . 56
6.2.1 General Results . . . 56

6.2.2 Example Results . . . 56
6.2.3 Duration Results . . . 57
6.2.4 Technique Section Results . . . 58
6.2.5 Quantifying Cyber Awareness . . . 60

7 Conclusions **63**
7.1 Discussion . . . 63
7.1.1 Research Question 1 . . . 63
7.1.2 Research Question 2 . . . 63
7.1.3 Research Question 3 . . . 64
7.1.4 Other Findings . . . 64
7.2 Limitations . . . 64
7.2.1 Experiment Design . . . 64
7.2.2 Corona Virus . . . 65
7.2.3 Time Constraints . . . 65
7.3 Further Research . . . 65

A Scene Overview **70**

B UML of scenario editor scripts **71**

C UML of game behavior scripts **72**

D Answers given for the question of the Social Engineering Simulator **73**
D.1 First Round . . . 74
D.2 Second Round . . . 76

List of Figures

2.1 Cyber space at the overlap of data, system, and human [23]. . . . 13
2.2 Information and Communication Technology security [82]. . . . 13
2.3 Information security [82]. . . . 13
2.4 Cyber security [82]. . . . 14
2.5 The relationship between Information and Communication Technology security, information security, and cyber security. [82]. . . . 15
2.6 The Fogg Behaviour Model with its three factors: motivation, ability and triggers [26]. . . . 18
2.7 The COM-B model with its three factors: capability, opportunity and motivation [52]. . . . 20
2.8 The Behaviour Change Wheel [52]. . . . 20

4.1 The first prompt the player sees while playing the Social Engineering Simulator. . . . 32
4.2 The credibility meter shown in the left side bar of the Social Engineering Simulator. . . . 34
(a) The credibility meter. . . . 34
(b) The credibility meter starts blinking when its value becomes low. . . . 34
4.3 An example of a story made in the scenario editor of the Social Engineering Simulator. . . . 37
4.4 The story node and the corresponding screen of a story node. . . . 38
(a) The story node. . . . 38
(b) The screen of a story node. . . . 38
4.5 The image node and the corresponding screen of an image node. . . . 38
(a) The image node. . . . 38
(b) The screen of an image node. . . . 38
4.6 The input node and the corresponding screen of an input node. . . . 39
(a) The input node. . . . 39
(b) The screen of an input node. . . . 39
4.7 The conversation node and the corresponding screen of a conversation node. . . . 39
(a) The input node. . . . 39
(b) The screen of a conversation node. . . . 39
4.8 The whatsapp node and the corresponding screen of a whatsapp node. . . . 40
(a) The whatsapp node. . . . 40
(b) The screen of a whatsapp node. . . . 40
4.9 The mail node and the corresponding screen of a mail node. . . . 41
(a) The mail node. . . . 41
(b) The screen of a mail node. . . . 41
4.10 The principle of time node and the corresponding screens of the different stages. . . . 41
(a) The principle of time node. . . . 41
(b) The principle of time screen during gameplay. . . . 41
(c) The principle of time showing a mistake. . . . 41
(d) The resolution screen of the principle of time. . . . 41
4.11 The principle of need and greed node and the corresponding screens of the different stages. . . . 42
(a) The principle of need and greed node. . . . 42
(b) The principle of need and greed screen during the survey. . . . 42
(c) The principle of need and greed showing the resolution screen. . . . 42
4.12 The principle of distraction node and the corresponding screens of the different stages. . . . 43
(a) The principle of distraction node. . . . 43
(b) The principle of distraction node showing a number the player should remember. . . . 43
(c) The player is asked to fill in the sequence during the principle of distraction gameplay. . . . 43
(d) In the second phase of the distraction game, the player is also asked to fill in simple math sums. . . . 43
(e) The first resolution screen of the principle of distraction. . . . 43

(f) The second and final resolution screen of the principle of distraction. . . . 43
4.13 The principle of herd node and the corresponding screens of the different stages. . . . 45
(a) The principle of herd node. . . . 45
(b) The start position of the principle of herd game. . . . 45
(c) During the principle of herd game, the smaller cubes will move to either side. . . . 45
(d) The resolution screen of a question of the principle of herd. . . . 45
(e) The first resolution screen of the principle of herd. . . . 45
4.14 The end node. . . . 46
4.15 The end screen node and the corresponding screen of an end screen node. . . . 46
(a) The end screen node. . . . 46
(b) The screen of the end screen node. . . . 46
4.16 The achievement earned node and the corresponding screens of the different stages. . . . 47
(a) The achievement earned node. . . . 47
(b) The first screen of the achievement earned node. . . . 47
(c) The second screen of the achievement earned node. . . . 47
4.17 The GDPR node and the corresponding screen of a GDPR node. . . . 47
(a) The GDPR node. . . . 47
(b) The screen of the GDPR node. . . . 47
4.18 The property changer node. . . . 48
4.19 The score changer node. . . . 48
4.20 The credibility changer node. . . . 48
4.21 The conditional node. . . . 49

6.1 The percentage of wrong answers given in the first round of examples (part 1), compared to the second round of examples (part 3), for both legit and malicious examples of the game. . . . 57
6.2 The percentage of wrong answers given in the first round of examples (part 1), compared to the second round of examples (part 3), for only malicious examples of the game. . . . 57
6.3 The average duration per example in the first and second round. . . . 58
(a) The average duration per example in the first round. . . . 58
(b) The average duration per example in the second round. . . . 58
6.4 The amount of participants that were willing to give away certain information during the *need and greed* section. . . . 59
6.5 The answers given during the *herd* section. . . . 59
6.6 The longest correct sequence given in the first and second phase of the *distraction* technique section. . . . 60
(a) The longest correct sequence given in the first phase of the *distraction* technique section. 60
(b) The longest correct sequence given in the second phase of the *distraction* technique section. 60
6.7 Histogram of the amount correct and wrong answers given for the *time* technique section. . . . 60
(a) Histogram of the amount correct answers given for the *time* technique section. . . . 60
(b) Histogram of the amount wrong answers given for the *time* technique section. . . . 60
6.8 Histogram of the quantified cyber awareness values. . . . 62

A.1 UML Diagram of the Scene used in the Social Engineering Simulator (SES). . . . 70

B.1 Simplified UML Diagram of the scenario editor scripts in the Social Engineering Simulator (SES). 71

C.1 Simplified UML Diagram of the game behavior scripts in the Social Engineering Simulator (SES). 72

D.1 Answers given for question 1 till 5 of the first round of examples of the Social Engineering Simulator. . . . 74
(a) Answers for example 1, first prompt. . . . 74
(b) Answers for example 2, first prompt. . . . 74
(c) Answers for example 2, second prompt. . . . 74
(d) Answers for example 3, first prompt. . . . 74
(e) Answers for example 4, first prompt. . . . 74
(f) Answers for example 5, first prompt. . . . 74
D.2 Answers given for question 5 till 7 of the first round of examples of the Social Engineering Simulator. . . . 75
(a) Answers for example 5, second prompt. . . . 75
(b) Answers for example 6, first prompt. . . . 75
(c) Answers for example 6, second prompt. . . . 75

(d) Answers for example 7, first prompt. . . . 75
(e) Answers for example 7, second prompt. . . . 75
(f) Answers for example 7, third prompt. . . . 75
(g) Answers for example 7, fourth prompt. . . . 75
D.3 Answers given for question 1 till 5 of the second round of examples of the Social Engineering Simulator. . . . 76
(a) Answers for example 2, first prompt. . . . 76
(b) Answers for example 2, first prompt. . . . 76
(c) Answers for example 3, first prompt. . . . 76
(d) Answers for example 4, first prompt. . . . 76
(e) Answers for example 5, first prompt. . . . 76
(f) Answers for example 5, second prompt. . . . 76
D.4 Answers given for question 6 till 7 of the second round of examples of the Social Engineering Simulator. . . . 77
(a) Answers for example 6, first prompt. . . . 77
(b) Answers for example 6, second prompt. . . . 77
(c) Answers for example 6, third prompt. . . . 77
(d) Answers for example 6, fourth prompt. . . . 77
(e) Answers for example 7, first prompt. . . . 77

Acronyms

BCW Behaviour Change Wheel. 5, 7, 19, 20

C&C Command-and-Control. 7, 29

CBS Statistics Netherlands. 7, 42

DCC Defence Cyber Command. 7, 33

ENISA European Union Agency for Cybersecurity. 7, 12, 19, 20

FBM Fogg Behaviour Model. 5, 7, 18, 19, 21

GDPR General Data Protection Regulation. 3, 6, 7, 47

GUID Globally Unique Identifier. 7, 50, 51

IBAN International Bank Account Number. 7, 37

ICT Information and Communication Technology. 2, 5, 7, 12–15, 33

IDE Integrated Development Environment. 7, 49, 50

ITU International Telecommunication Union. 7, 12

RAT Remote Access Trojan. 7, 28, 29, 35

SES Social Engineering Simulator. 4–7, 9, 10, 16, 24, 31, 32, 34–38, 41, 42, 44, 48, 49, 53, 56, 60–63, 70–72, 74–77

UI User Interface. 7, 49, 51, 52

UML Unified Modeling Language. 7, 49, 50

WHO World Health Organization. 7, 9, 68

Chapter 1: Introduction

1.1 Introduction

In 2020 the world faced one of the biggest crises since the second world war; the corona pandemic. Around the globe, a huge amount of employees suddenly are bound to work remotely from home. While working from home brings a new work ethos with it, it also grants new opportunities for criminals, who proved themselves to be very flexible. According to Europol, there was a rapid increase in cybercrime in March and April, when most of the employees started working from home [66]. According to the Dutch Police, in 2020, there was an increase in cybercrime by 127% [63].

According to the Dutch Police, cybercrime contains all offences committed using a computer, smartphone, smartwatch or tablet, in short anything with a processor in it [61]. In this thesis, the focus lies on cybercrime, where the computer is used to exploit or persuade the human; social engineering. Criminals were very inventive with coming up with new social engineering possibilities in the corona crisis. One of the most known social engineering attacks is phishing emails. During the corona crisis, criminals sent emails supposedly from the World Health Organization (WHO) with a malicious link to a fake WHO site [64]. Other examples of phishing include emails from the 'bank' claiming they have a new antibacterial debit card [40]. Besides phishing, social engineering during the coronavirus took other forms. Criminals hid a keylogger in a corona help site, which gathered the victims' credentials [3]. Other examples include malware masquerading as WHO-developed apps [3].

While the criminals are flexible and inventive, preventive social engineering training and tips are often lacking. When looking at the tips given by Cybersecurity Ventures, who claim to be "world's leading researcher and publisher covering the global cyber economy and a trusted source for cyber security facts", against phishing during the corona crisis, targeted at cyber security experts, some outdated tips come up [32]. They claim phishing emails often have spelling errors and poor grammar, which is not valid anymore, since criminals have evolved and learned from the past [32]. The following claim is that phishing emails often contain links without security certificates [32]. Nowadays, with free tools like Let's Encrypt, adding security certificates takes around three minutes, and criminals are familiar with this and know how to use it. Cybersecurity Ventures also claims phishing emails often have generic greetings [32]. Nowadays, there exists a significant chance a user name is leaked in a data breach, which is sold on the web. Via this way, it is easy for criminals to use the proper greetings in phishing emails. Phishing mails might even contain old passwords of the users.

Besides being quickly outdated, classical interventions face another issue; they often focus on the method. However, methods can also rapidly change in a fast-changing environment like the technological world. Phishing was a hot topic during the upcoming of the internet and email. Most preventive measures focused on the phishing mail itself and what to look for in those emails. During recent years, however, we have seen new social engineering methods rise. Such as Whatsapp fraud, where the criminal sends a message via Whatsapp claiming to be a friend in need of money, where more and more people fall victim to [62]. Social engineering attacks also become more and more sophisticated, where multiple vectors such as email, SMS texts, phone calls and physical conversations are combined.

This calls for a new innovative, flexible intervention, which can be easily adjusted, and should also offer protection against unknown attack vectors. In this thesis, a new intervention is presented: the Social Engineering Simulator (SES). The SES is a web-based game that focuses not on the attackers' methods but the psychological techniques behind those attacks. By looking at the techniques rather than the methods, the social engineer attack method becomes irrelevant. This way, someone can be protected against attack methods that are yet unknown. The SES also features a story builder, in which the intervention can be easily adjusted to the target audience and the current issues at play. This way, the intervention is always relevant.

The SES uses McGuire's inoculation theory [50, 51]. This theory says that protection against persuasion can work similarly to inoculation against diseases. To protect someone against a disease, a small part of the disease

can be injected to stimulate the body, making anti-bodies. To protect someone against persuasion, small pieces of that persuasion can be shown to stimulate that person in coming up with counter-arguments and protection methods against that persuasion. A distinction is made between passive and active inoculation. With passive inoculation, the persuasion is shown to the user, together with protection against that persuasion. During active inoculation, the user has to develop ways to protect itself against the persuasion actively. Active inoculation is seen to be more effective in conferring resistance to persuasion than passive inoculation [9, 50]. It has been shown that gamified active inoculation effectively increased reliance against fake news by Van der Linden and Roozenbeek. [9, 69, 70]. The works of Van der Linden and Roozenbeek sparked new interest in using inoculation theory in modern technology. This work hypothesises that active gamified inoculation is also effective in increasing resilience against social engineering risks.

The game generates active inoculation by placing the player in the shoes of a social engineer. While they are in this role, they have to develop different persuasion and social engineering methods. They will learn various techniques used in social engineering attacks and will learn how to apply those. According to inoculation theory, this would generate the "anti-bodies" against social engineering attacks. To help better understand the different techniques, little interactive sections and games are developed. This way, the player can feel how the psychological techniques are experienced, which can also trigger the generation of social engineering "anti-bodies".

Besides increasing the player's awareness, the game could also give insights into the users' behaviour. This can be helpful for companies to see how their employees perform. When most employees fall for a specific social engineering attack or technique, the company knows where to focus their security training or policies on. The SES delivers data about the behaviour by providing seven examples of interactions before and after the game. The user should indicate whether these are dangerous, suspicious or safe. The answers to these examples are analysed. If the player provides more correct answers after playing the game than before playing the game, it might indicate a positive effect of the game.

The effectiveness of the game is tested in the context of the Dutch Armed Forces. In 2020 the Dutch Armed Forces had around 66.000 employees. All of those employees should be adequately trained against social engineering risks. Especially when keeping in mind, the Dutch Armed Forces' adversaries have both the capacity and the resources to perform the most advanced social engineering attacks. The experiment was tested in a quasi-experiment, with a three-group pretest-posttest design. Where two groups played two versions of the game, and one group was the control group.

In a meta-analysis of social engineering interventions, Bullee and Junger selected 19 articles from 418 studies [12]. They found that generally using interventions are indeed effective in countering social engineering attacks. However, they also found varying results between different works. There is no consensus in the literary world about the effectiveness of awareness interventions and which intervention works best. In other works, Bullee et al. researched the effectiveness of interventions of physical, social engineering attacks and the effect of the *authority* principle [13]. Junger et al. looked into priming and warnings as tools of protection against phishing [39]. Gragg defined a multi-level defence against social engineering, which addresses the psychological triggers behind social engineering attacks [28]. Abawjy has researched different delivery methods of cyber security awareness interventions and found that a combined delivery method works best [2]. Bowen et al. have tried to quantify the security posture of human organisations. They did this by measuring the susceptibility to phishing attacks [10]. Gardner and Thomas wrote a book on how to design a security awareness program that helps to protect against both social engineering and technical threats [27]

1.2 Research Question

Multiple scholars have suggested that inoculation theory might be a feasible method in protecting people against social engineering risks [28, 60, 71, 80]. However, active inoculation has never been practically used as an intervention method. The works of Roozenbeek and Van der Linden sparked new interest in inoculation theory [69, 70]. This work is the first to use active inoculation in a gamified environment as an intervention method against social engineering risks. It can be argued that most interventions contain a form of passive inoculation by showing examples of existing social engineering attacks. However, using active inoculation as protection against social engineering risks has not been researched. This thesis will research the effectiveness of gamified active inoculation as a form of protection against social engineering risks. It has been shown that gamified active inoculation has effectively raised resilience against fake news [9, 70]. This research hypothesises that this is also an effective method in raising resilience against social engineering risks. The following research question has been formulated.

RQ 1: How effective is an intervention in the form of gamified active inoculation against social

engineering attacks?

To help the active inoculation and let participants truly experience social engineering techniques, as will be described in section 2.2.3, in a safe environment, the intervention should simulate those social engineering techniques. Experiencing these techniques is a form of active inoculation and could generate counterarguing and help people recognise threats earlier. This thesis will research the possibilities of that. The following research question has been formulated.

> **RQ 2:** Can physiological techniques and principles used by social engineers be simulated in a gamified active inoculation environment?

In The Fake News game, the researchers were able to see the difference in resilience against fake news by looking at the level before and after the game [69, 70]. In an intervention against social engineering risks, a similar method could be used. By looking at the intervention results and the participants' behaviour, it might be possible to extract certain risks. This could help specify where the focus on further training should be. The following research question has been formulated.

> **RQ 3:** Can social engineering risks be derived, and the level of resistance against social engineering attacks be quantified from a gamified active inoculation environment?

In section 2, the theoretical framework of this thesis is given. Next, in section 3, a list of various social engineering attack scenarios and known attacks is given to show different social engineering possibilities and build on in the following sections. In section 4, the game's design is explained on a general and a technical level. The method of the experiment is explained in section 5. Next, the experiment results are explained and analysed in section 6. Finally, in section 7, the conclusions, limitations and recommendations for future work are given.

Chapter 2: Theoretical Framework

2.1 Cyber Security

2.1.1 Definition of Cyber Space

This thesis is a work in the extensive discourse of the science of cyber security. First, the term cyber, also defined as cyberspace, will be further explored to understand this concept better. Cyber security can be defined in a straightforward definition as the act of making cyber space safe from damage or threat [23]. This raises the question, what is cyber space? In their book, Edgar et al. talk about the different perspectives of cyber space and how the definition has evolved over time [23].

Data Perspective

The first perspective centres around the data or information that resides in cyber space, where the focus is on how to encode and construct information into transferable data [23]. This perspective came from the information theory, which is directed toward the digitisation and encoding of information.

Technology Perspective

The next perspective is that the technological perspective says that cyber space encapsulates data or information and the technology necessary to transmit it [23]. This includes all the hardware as well as software.

Cybernetic Perspective

The most recent perspective, and the perspective that will be used from now on in this thesis, is that cyber space includes the human in addition to the data and technology [23]. Edgar et al. argue that cyber space is a metaphysical construct created by the confluence of hardware, the data it creates, and the humans that interact with said hardware and produce/consume the information contained in the data [23]. Systems would not function without human intervention. It also takes into account that humans are often the weakest security link, who are targeted with attacks against their psychological behaviour [23]. A model of the definition can be seen in figure 2.1.

2.1.2 ICT Security vs Information Security vs Cyber Security

According to Von Solms and Niekerk, the term cyber security derives from the term information security and is sometimes used interchangeably with the term cyber security, but this is not correct [82].The International Telecommunication Union (ITU) and European Union Agency for Cybersecurity (ENISA) define cyber security as:

> "Cyber security is the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organisation and user's assets. Organisation and user's assets include connected computing devices, personnel, infrastructure, applications, services, telecommunications systems, and the totality of transmitted and/or stored information in the cyber environment. This refers to the protection of information, information systems, infrastructure and the applications that run on top of it from those threats that are associated with a globally connected environment. The general security objectives comprise the following:
>
> - Availability
> - Integrity, which may include authenticity and non-repudiation
> - Confidentiality" [24, 36]

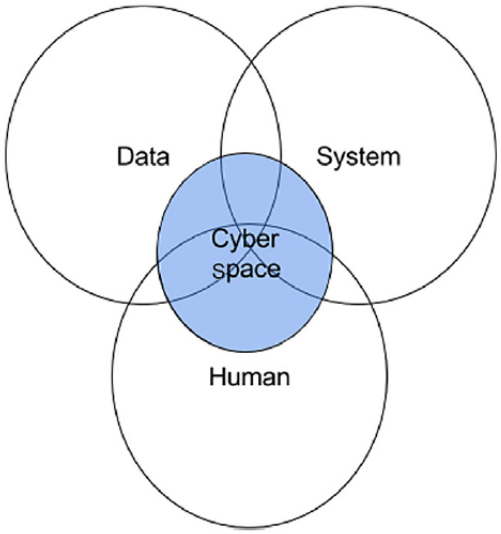


Figure 2.1: Cyber space at the overlap of data, system, and human [23].

As described earlier, the definition of cyber space flows from a perspective that looks at data and a perspective that looks at technology. Cyber security defined in the same way. The definition was first focused on technology, next on data, and finally on cyber space. Von Solms and Niekerk look at this by looking at the difference between Information and Communication Technology (ICT) security, information security and cyber security. [82]. They argue that all security is about the protection of *assets*, from *threats* which are posed by certain *vulnerabilities*. These threats can be countered by *controls*, which help to reduce the *risk* posed by these vulnerabilities [82].

ICT Security

They say that in the case of ICT security, the asset(s) that need to be protected are underlying information technology infrastructure (see figure 2.2) [82].

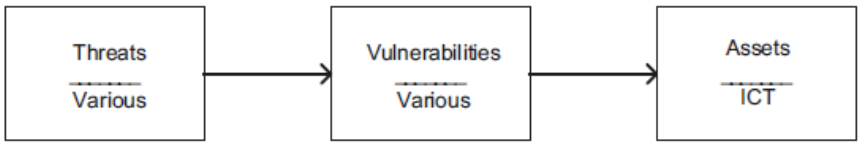


Figure 2.2: Information and Communication Technology security [82].

Information Security

On the other hand, information security extends this definition by saying that assets to be protected include all aspects of the information itself. It thus includes the protection of the underlying ICT assets and then goes beyond just the technology to include information that is not stored or communicated directly using ICT (see figure 2.3) [82].

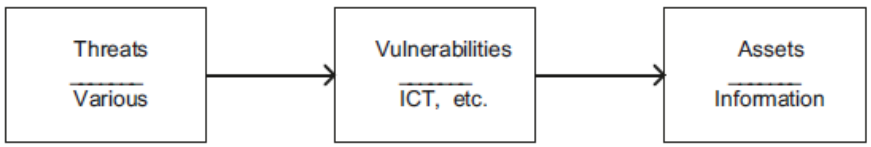


Figure 2.3: Information security [82].

Cyber Security

The most crucial difference between ICT and information security and cyber security are the assets they protect. In cyber security, the assets that need to be protected can range from the person him/himself to household appliances to society's interests, including critical national infrastructure, or anything and anyone in cyberspace [82]. In cyber security, information and ICT are the underlying cause of the vulnerability [82]. The most defining characteristic of cyber security is that all assets that should be protected need to be protected because of the vulnerabilities that exist due to the use of ICT that forms the basis of cyberspace.

Von Solms and Niekerk also take intangible assets into account, such as an ethical dimension. They give the example of botnets, where being a part of a botnet does not always mean that confidentiality, integrity or availability have been directly affected [82]. However, if such a botnet is used to commit a crime, the owner of the computer might be an unknowing accomplice [82]. Another intangible asset they state is the protection of trust that citizens have in using cyberspace for commercial purposes, which is seen as vital by different nations [82].

In cyber security assets include the personal or physical aspects, both tangible and intangible, of a human being [82]. It also includes the protection of societal values (intangible) and national infrastructure (tangible). Cyber security thus includes both tangible and intangible assets relating to the well-being of either the individual or society at large [82]. In the case of cyber security, the information itself can be classified as a vulnerability (see figure 2.4) [82].

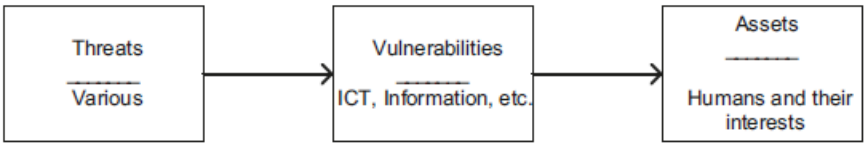


Figure 2.4: Cyber security [82].

Just as information security expands on the concepts of ICT security, cyber security needs to be seen as an expansion of information security. Cyber security is about the protection of more than just the information of a person or an organisation. Cyber security is also about the protection of the persons(s) using resources in a cyber environment and about the protection of any other assets, including those belonging to society in general, that have been exposed to risk as a result of vulnerabilities stemming from the use of ICT. The relationship between these three overlapping concepts can be seen in figure 2.5 [82].

Von Solms and Niekerk conclude by talking about the role of the human, which will continually expand. In ICT security, the human was often seen as a threat. In information security, the role has grown to become an increasingly integral part of the supporting system; thus, they have grown to a vulnerability. Currently, in cyber security, humans and societies have become assets that need to be protected. While they still can be seen as a threat and a vulnerability, they can also be seen as an asset that needs to be protected in cyberspace [82]. They claim that the human element is playing an ever-increasing role in cyber security, and the current set of standards and best practices is not comprehensive enough to secure cyberspace.

2.1.3 Human Factors in Cyber Security

We have seen that the human plays a fundamental part in both the definition of cyber space as well as cyber security [23, 82]. The role humans play in cyber security is sometimes defined as the human factor of cyber security. Young et al. say that the human factors' perspective is mostly missing from the wider cyber security dialogue [84]. They say the human factor is not a static element, and to enhance cyber security, we need more fit for purpose interventions to change human behaviour, which take into account the dynamic context of the cyber system [84].

Pfleeger et al. argue that the key principle of the human factors' study field is to design technology that fits a person's physical and mental abilities, namely fitting the task to the human [60]. It looks at the capabilities of the human and how policies should fit into those capabilities. This approach is often straightforward for behavioural experts, but not for the technologists and policy-makers who make the security policies [60].

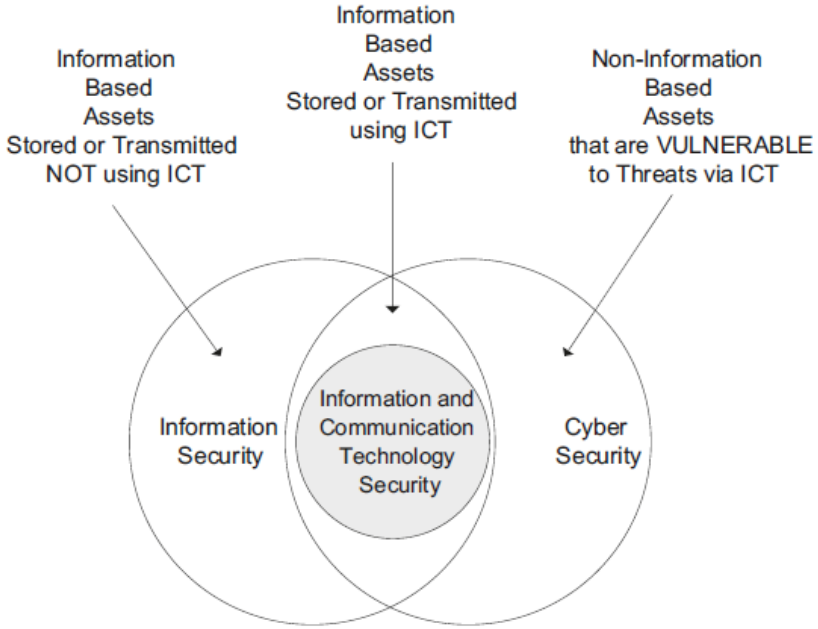


Figure 2.5: The relationship between Information and Communication Technology security, information security, and cyber security. [82].

The Human as the Weakest Link

Often humans are called the weakest link in the cyber security chain [60, 74, 79]. Users should practice safe and secure behaviour online, but often fail to do so [60]. If the human violates the security policies, it becomes an insider threat, which means that the threat is coming from inside the company [31, 84]. There are three categories of insider threats:

1. The accidental insider who violates the security policies non-volitional, these include human errors.
2. The accidental insider who might show volitional behaviour but who is not motivated by malicious intentions, these also include human errors.
3. The malicious insider who intentionally violates policies for malicious proposes [31, 84].

2.2 Social Engineering

2.2.1 Definition

Attacks that focus on humans and misuse the human as the weakest link in the cyber security chain are social engineering attacks. With a social engineering attack, the attackers target humans to compromise information systems [45]. One of the most famous social engineers is Mitnick, who was convicted for hacking, and later worked for the FBI [54]. Mitnick defines social engineering as 'using influence and persuasion to deceive people and take advantage of their misplaced trust to obtain insider information' [54]. In the past years, multiple different definitions of social engineering were given by different scholars. Mouton et al. looked at the different definitions and tried to give a definitive definition. This definition will be the definition that will be used in this thesis:

> The science of using social interaction as a means to persuade an individual or an organisation to comply with a specific request from an attacker where either the social interaction, the persuasion or the request involves a computer-related entity [55].

2.2.2 Types of Social Engineering Attacks

Social engineering attacks are multifaceted. Kromgholz et al. define five different types of social engineering attacks [45]. Section 3 will look at several examples in more detail.

Physical Approaches

As the name suggests, with this approach, the attacker performs some form of physical action to gather information on a future victim. This approach can range from looking through trash (dumpster diving) to trying to gain access to an office, but also theft or extortion [45].

Social Approaches

According to Krombholz et al., the most essential aspect of successful social engineering attacks are social approaches. Attackers rely on socio-psychological techniques to manipulate their victims. These techniques will be further explored in section 2.2.3. The attacker might first try to develop a relationship with their victim to increase the chance of such an attack. This type of attack can happen via different vectors, such as computers and phones.

Reverse Social Engineering

This type of attack is an indirect approach where the attacker tries to make the victim believe they are trustworthy. The goal is to make the victim approach the attack. This attack has three parts: sabotage, advertising and assisting [57] in [45].

Technical Approaches

Technical approaches are primarily carried out over the internet. According to Granger, the internet is especially interesting for harvesting passwords, because most users often use the same simple password [29]. With the technical approach, attackers use search engines or automated tools to gather personal information [45]. This is still considered social engineering, because these technical tools abuse the human element and are targeted to misuse human flaws.

Socio-technical Approaches

This approach is arguably the most potent approach and combines several of the different approaches discussed above. In this approach, the attacker combines the technical and social approach. Attacks of this type include leaving attractive USB's, which contain malware or send emails with links with malware in them (phishing). With a phishing attack, the attackers send their emails to many users, hoping that a few of them will click the link. With spear-phishing, the attacker makes a more targeted email using information gathered from various sources.

2.2.3 Social Engineering Techniques

As described in section 2.2.2, attackers use different social techniques to persuade their victims in social engineering attacks. Cialdini is a scholar in persuasion in marketing, and most scholar underlines his six principles of persuasion as the fundamental principles of social engineering [4, 14, 28, 33, 45]. Cialdini's principles were primarily focused on marketing purposes, but still viable for social engineering. Stajano and Wilson took the principles of Cialdini and focused them more on criminal behaviour for security purposes [73, 74]. This thesis will use a combination of both Cialdini's and Stajano and Wilson's principles and will play a crucial part in the SES. It is essential to notice that most attacks do not use only one principle but use a combination of multiple principles.

Distraction

The first principle Stajano and Wilson describe is distraction, which is a principle Cialdini does not mention. They argue that distraction is at the heart of innumerable fraud scenarios. The core of the principle is that while people are distracted by what has their interest, attackers can do malicious activities, and the victim will not notice it. This can also be seen in a security context with the friction between security and usability. The user might be so focused on their task, they might not pay attention to the security policies and thus might become an accidental insider. Attackers might also abuse this by targeting security policies that users might find inconvenient and thus use less [73].

Authority

The following principle is called authority by Cialdini and social compliance by Stajano and Wilson. Humans are 'trained' by society to trust authority figures. This means that people have an automatic trust for other people in uniforms [14, 73]. According to Mitnick, this is an even bigger problem for law enforcement and the

military, where the respect for a higher rank plays an even more prominent role [54]. As long as an attacker poses as a higher rank, the victim will not question the authority of the attacker [54]. Stajano and Wilson describe that the psychological factor here is that it is hard for a stranger to force a victim to behave in the desired way, but it is easier for an attacker to let the victim behave according to an already-established pattern, namely that of obeying a recognised authority. However, the authority principle includes more than only established authority figures like law enforcement and the military. It also includes the expected people in the expected place. For example, in the workplace, we expect workers, and in a bank we expect people in formal clothing. This principle also extends beyond just human authority figures. People, especially non-technical users, are likely to trust email from their bank, which is abused in (spear)phishing attacks [14, 73].

Herd

The herd principle is called social proof by Cialdini. The principle says people generally feel safe when they see other people around them, acting and doing the same. Stajano and Wilson give the example of a street scam, where the scammer works together with another scammer to show the game is easy and winnable. When the victim sees the other scammer winning, he is also triggered to play the game, however when he plays the game, the scammer uses a trick, and the victim will always lose. Cialdini also notes that the likes of other people influences people. This can be used online by using fake profiles and fake reviews. Fake identities can also be used to promote a political candidate or party, which is called astroturfing [14, 73].

Dishonesty

This principle is only noted by Stajano and Wilson and not by Cialdini. If people do something illegal, it can be used against them by attackers, making it harder for them to seek help. An example of this is are phishing emails, in which the victim is promised much money if they help with money laundering. When they realise they are scammed, the victims are afraid to go to the police, because they helped with something illegal. Besides illegal things, the principle can also be used to interpret things where people are ashamed of. For example, phishing emails where the attacker claims he has video material of the victim watching porn (sextortion).

Kindness

Stajano and Wilson call this principle kindness, and Cialdini calls this liking. This principle can be seen as the dual of the dishonest principle. People like to be seen as a friendly and helpful person. This is why most people are helpful to other people. Attackers could misuse this by abusing or taking advantage of people's generosity or kindness [14, 74]. For this thesis, this principle will also include Cialdini's principle of reciprocation under the kindness principle. The reciprocation principle says if we receive something, we automatically want to return something. An excellent example of this is street sellers, who first give the target something for free, such as a free newspaper. After we receive this free sample, we are more likely to listen to the street seller [14].

Need and Greed

The following principle is only named by Stajano and Wilson. We are vulnerable to the needs and desires we have. When attackers know what we want, they can manipulate us. In practice, this is used by offering money to victims. However, this can also be used in love scams, where the attacker claims to be a partner of the victim who lives in a foreign country and is in need of money [73].

Time or Scarcity

This is a combination of Stajano and Wilson's time principle and Cialdini's scarcity principle. The time principle by Stajano and Wilson says that while people are under time pressure to make crucial choices, they use a different decision strategy and thus think less clearly. Attackers misuse this by forcing the victim to make a decision quickly so that the victim will act on impulse according to predictable patterns [74]. Cialdini's scarcity principle is comparable. However, this principle says that scarce items trigger us. If something is rare and scare, it becomes more valuable. Attackers might misuse this by claiming there are only a few units left of an item, which makes the item rarer and thus valuable, but also creates time pressure for the victim [14].

Commitment and Consistency

This principle was only described by Cialdini. He describes that people like to be seen as consistently committed to something good, and our words and actions are in alignment. This might be misused by first letting the victim do or say something and letting them follow up on this, even if it is not in their best interest. This principle is not often used in social engineering attacks, but more focused on marketing purposes [14].

2.3 Behavior change

As we have seen, social engineering has a significant social aspect. In contrast to a technical or physical aspect, it is impossible to add extra controls like encryption or extra physical security in social engineering. The human aspect can be protected by creating awareness via awareness interventions, with the ultimate goal of changing human behaviour.

2.3.1 Awareness Interventions

The goal of an awareness intervention is to create awareness and mitigate the risks of social engineering attacks. In the past, different interventions using different methods have been developed by different scholars. Using embedded training interventions are shown to be effective in preventing phishing [46]. Using video games as an intervention tool has been shown to be effective [17, 33, 34]. Anti-phishing training effectively protects children against phishing risks, although there is a decay over time [47, 59]. While only using priming and warnings against social engineering risks are not effective [39, 46]

In a meta-analysis of social engineering interventions, Bullee and Junger selected 19 articles from 418 studies [12]. They found that generally using interventions are indeed effective in countering social engineering attacks. They also provide several suggestions. For instance, they recommend to keep performing social engineering mock attacks in combination with an intervention, which could reduce victimisation. Besides, they recommend sharing and publishing the results of the tested intervention [12].

2.3.2 Behavioral Change Theories

To counter undesirable behaviour, an awareness intervention can be conducted. Awareness interventions can have multiple different forms and can be both technical or physical. The science behind awareness interventions comes from the area of behavioural change. Behavioural change studies look at how human behaviour can be influenced and ultimately can be changed. Various behavioural change theories have been developed over the past years. These theories look at how human behaviour can be changed for the better. The most important theories will now be set out.

Fogg Behavior Model

The Fogg Behaviour Model (FBM) is behavioral change model designed by Fogg in 2001 [26]. This psychological model identifies and defines three factors that control behaviour. The framework was specially designed for persuasive technology. The three principal factors of the FBM are motivation, ability and triggers. This is why the FBM is also somethings called the B=MAT model. Fogg argues that all three factors must be present at the same time for the behaviour to occur. Figure 2.6 shows a visualization of the FBM. The vertical axis is for motivation, and the horizontal axis is for ability. This creates a plane in which a diagonal arrow can be drawn. This arrow says that as people have increased motivation and ability, they will be more likely to perform the target behaviour. This shows that only having motivation, or only having the ability, is not good enough; the targeted behaviour requires both [26].

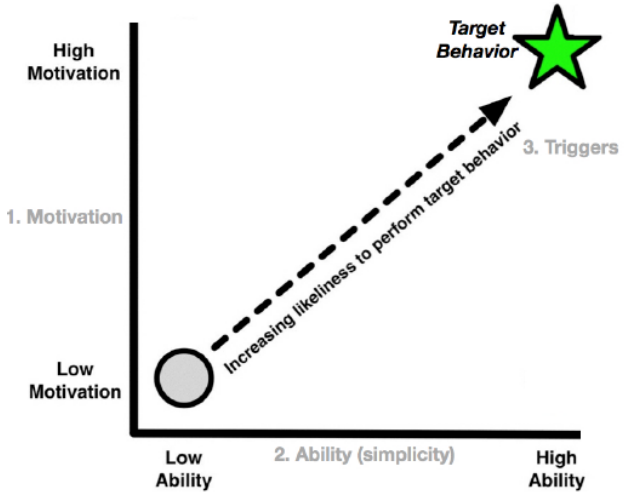


Figure 2.6: The Fogg Behaviour Model with its three factors: motivation, ability and triggers [26].

The FBM implies that motivation and ability are a trade-off of sorts. People with low motivation may perform the behaviour if the ability is high, which indicates an effortless task. However, the inverse scenario also applies; if the motivation is very high, people might go to extremes to perform the behaviour, even if their ability is low [26].

Fogg argues that behaviour also must be triggered, which is often the missing piece. Without an appropriate trigger, the behaviour will not occur even if both motivation and ability are high. A trigger can take many forms, but it has three characteristics; we need to notice it, we need to associate with the target behaviour and both motivation and ability need to be high. When the combination of motivation and ability places a person above the behaviour activation threshold, the trigger will cause that person to perform the target behaviour [26]. ENISA says the FBM could be used to help think about possible awareness interventions [25].

MINDSPACE

The MINDSPACE model was designed by Poldan et al., who work for the Institute for Government in the United Kingdom. They present a new model for behavioural change. They argue that there are two focus points when changing behaviour; either changing the context or changing the mind. They present nine influences on human behaviour and change. Together they form the acronym MINDSPACE.

1. **Messenger** - We are heavily influenced by who communicates information.
2. **Incentives** - Our responses to incentives are shaped by predictable mental shortcuts, such as strongly avoiding losses.
3. **Norms** - We are strongly influenced by what others do.
4. **Defaults** - We 'go with the flow' of pre-set options.
5. **Salience** - Our attention is drawn to what is novel and seems relevant to us.
6. **Priming** - Our acts are often influenced by sub-conscious cues.
7. **Affect** - Our emotional associations can powerfully shape our actions.
8. **Commitments** - We seek to be consistent with our public promises, and reciprocate acts.
9. **Ego** - We act in ways that make us feel better about ourselves [20, 21].

Interestingly, even the MINDSPACE model was designed to change behaviour for the better, the influences they state are overlapping with the techniques social engineers used that were described earlier. We see that both state the commitment influence. The norms influence is similar to the herd principle, and the messenger influence is similar to the authority principle. This indicates that the MINDSPACE model can also be used for persuasion purposes.

Behavior Change Wheel

The Behaviour Change Wheel (BCW) was designed by evaluating different behaviour change frameworks. One of the frameworks Michie et al. looked at was MINDSPACE. They criticised that even though MINDSPACE recognised two systems in which human behaviour can be influenced, the reflective and the automatic system. It only focused on the latter and did not attempt to link influences on behaviour with these two systems. Michie et al. argue that three factors influence the behaviour; capability, opportunity, and motivation, which they call the COM-B model (see figure 2.7). They define capability as the psychological and physical capacity to engage in the activity, including having the necessary knowledge and skills. Motivation is defined as all the brain processes that energise and direct behaviour, including habitual processes, emotional responding and analytical decision-making. Opportunity is defined as all the factors that lie outside the individual that make the behaviour possible or prompt it [52].

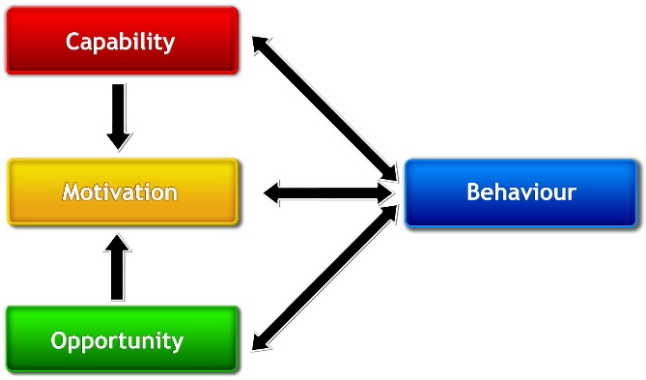


Figure 2.7: The COM-B model with its three factors: capability, opportunity and motivation [52].

This model of behaviour can provide a basis for designing interventions aimed at behavioural change. An intervention might change one or more components in the behaviour system. With these three components that generate behaviour, Michie et al. developed further subdivisions that capture important distinctions noted in the research literature. They identified six components within the behavioural system. Based on the literature, they developed the BCW, a framework for behavioural change, see figure 2.8. One of the strengths of this framework is that it incorporates context very naturally. Behaviour in context is the starting point of intervention design [52]. ENISA says the COM-B model and the BCW can be used to identify why the desired behaviour may or may not be carried [25].

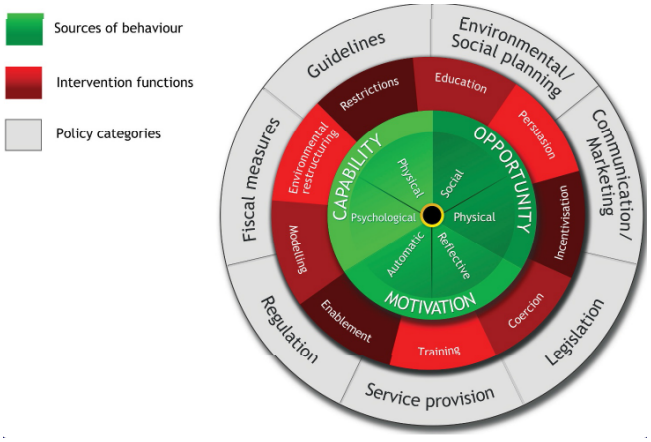


Figure 2.8: The Behaviour Change Wheel [52].

2.4 Inoculation Theory

As seen earlier, during social engineering attacks, different persuasion tactics are used. We have seen that one way to protect one against those attacks are changes to the behaviour. However, also theories exist that specifically focus on protection against persuasion. One of the most famous theories is inoculation theory [50, 51]. This theory was recently used in an online game against fake news, with positive results [69, 70]. This thesis will also build on this theory.

Inoculation theory is created by McGuire and talks about the resistance of persuasion [50, 51]. McGuire draws the parallel between medical inoculation and persuasion inoculation [15]. Compton gives the example of an annual flu shot, where weakened versions of the influenza virus are injected to trigger the response in the form of the production of antibodies [15]. Similarly, during persuasion inoculation, weakened versions of the persuasion messages are offered, weak enough to not persuade someone, but strong enough to trigger the response. This form of pretreatment is called refutational pretreatment, in contrast to the more classical supportive pretreatment [51]. When looking at the medical analogy, giving a dose of vitamin to boost the health before facing a potentially harmful virus would be a supportive pretreatment [15, 51]. Supportive pretreatment can be seen as the more classical approach by just giving a warning against persuasion on the forehand. To test whether his pretreatment was successful, McGuire did experiments where all participants were exposed to an attack message or messages, but only some received pretreatment before the attack [15]. McGuire moved past

proving that refutational pretreatments work into how refutational pretreatments work. He did this by making a basic model consisting of a catalyst (threat) and activity (counterarguing) [15].

2.4.1 Threat and Counterarguing

Threat

The threat in inoculation is not the message itself but the response to that message. It is the vulnerability that an existing position, which was once thought to be safe from change, may be at risk. The presence of unexpected challenges to an existing position threatens that position's perceived security, which McGuire called implicit threat [15, 51]. In contrast, inoculation messages may contain forewarnings of impending persuasive attacks. McGuire calls these explicit threats [15, 51]. It is assumed that for the inoculation process to be effective, receivers must perceive a threat to motivate them to strengthen their current attitudes [7]. This theory can also be linked to the COM-B model or the FBM where the threats generate the motivation, which creates the behaviour [26, 52].

Counterarguing

Counterarguing is the process of generating counterarguments and refutation after the inoculation pretreatment. This process is different from rephrasing the counterarguments learned in pretreatment, but solely means the generation of own arguments after the pretreatment [51]. This means the user cannot use the counterarguments learned in a pretreatment, but can also generate them independently.
Compton describes this as the basic model of inoculation-conferred resistance to influence: raising and refuting challenges confers resistance to future more substantial challenges by (1) revealing the vulnerability of the position (threat) through an explicit forewarning and/or the presence of counterarguments, which motivates (2) counterarguing [15].

2.4.2 Contemporary inoculation theory

While inoculation theory is an old theory, it still has contemporary relevance [15]. Eagly and Chaiken describe it in their famous book *The psychology of attitudes* as "the grandparent theory of resistance to attitude change" [22] in [15, 81]. Inoculation Theory has found application in multiple contemporary uses, such as; health, politics, commerce, and, more recently, cybersecurity [15, 16, 71, 80]. Compton has conducted a review of issues in contemporary inoculation theory, which includes an overview of what we know of inoculation theory's moderators, mediators, and outcomes.

Moderators

Important moderators are pre-attitude, perceived involvement, and self-efficacy [15]. Since inoculation is a preventive strategy, it is crucial that the target attitude must be in place before inoculation. Else the inoculation messages can have a persuasive effect. The amount of perceived involvement with an issue can influence how people experience a particular threat. Some scholars argue that it is the suggested involvement that leads to counterarguing; however, this is not supported by a meta-analysis of Banas and Raines [7, 15]. The amount of self-efficacy influences how inoculation messages should appeal to the target audience [15].

Mediators

As described earlier, essential mediators in the basic model are threats and counterarguing. Beyond the basic model, it is argued that mediators are perceived involvement, affect, and post-inoculation talk [15]. Scholars argue that the involvement of the target issue helps the resistance, and research suggests that even a single inoculation pretreatment message can enhance perceived involvement with a target issue. Affect shows how different emotions influence the resistance against persuasion. Recent evidence suggests that counterarguing in inoculation is not only an internal process but can also trigger external dialogue, which is called the post-inoculation talk. Some researchers argue that this also strengthens attitudes about the issue [15].

Outcomes

The outcomes of inoculation theory are attitudes and behavioural intentions. For many inoculation theory scholars, the most critical benchmark for inoculation's efficacy is whether a target attitude is resistant to persuasion [15]. Simply said, was the attack message more persuasive to those who did not receive an inoculation pretreatment compared to those who did [15]? There is also research done exploring attitude strength, certainty, and confidence. Inoculation scholars have also assessed the impacts on behavioural intentions, such as the side effects of the pretreatment. For example, can inoculation in a political context also affect voting intentions [15]?

2.4.3 Meta-analysis

Banes and Raines have conducted a meta-analysis of 54 studies, testing the effectiveness of inoculation theory at conferring resistance and examining the theory's mechanisms. The analyses revealed inoculation messages to be superior to supportive messages, such as only giving participants a message promoting an attitude they held and no-treatment controls at conferring resistance. The findings also provide evidence of inoculation's effectiveness compared to control conditions, including supportive treatments and no-treatment controls. The results suggest inoculation treatments are more effective than no-treatment controls or supportive treatments in creating resistance to attitude change [7].

2.4.4 Inoculation Theory and Fake News

Introduction

Van der Linden and Roozenbeek used inoculation theory against fake news by using a video game. They found that most inoculation papers were focused on one domain of fake news, such as conspiracies or climate change. However, they saw an opportunity where inoculation could prove a 'broad-spectrum vaccine' against misinformation [69, 70]. They did this by focusing on the common tactics used in misinformation production rather than just the content of a specific persuasion attempt [70]. They were the first to implement the principle of active inoculation in a novel experiential learning context: the Fake News Game, a serious social impact game that was designed to both entertain, as well as educate [70]. With active inoculation, the authors mean that participants have to actively generate pro- and counterarguments, rather than passive inoculation, where participants are provided with both the counterarguments and refutation [7, 69]. Providing cognitive training on a broad set of techniques within an interactive and simulated social media environment can help people apply these skills across various issue domains [70]. The game can be seen as a novel psychological intervention that aims to confer cognitive resistance against fake news strategies.

The Fake News Game

The game is a freely accessible browser game, which takes around 15 minutes to complete. The game engine can render text boxes, images, and Twitter posts to simulate the spread of online news and media [70]. It is choice-based; players are presented with various options that will affect their pathway throughout the game. In the game, the player will take on the role of a fake news creator. They need to gain followers and keep credibility. While playing the game, the player earns one of six badges, which stands for one of the six strategies used to spread misinformation [69, 70]. They place the participants in the shoes of a fake news producer. This way, the player is not only exposed to small portions of misinformation (as is the case with passive inoculation) but are instead prompted to think proactively about how people might be misled in proceeding to achieve a goal (winning the game). Van der Linden and Roozenbeek argue that this process of active inoculation positively affects the participant's ability to recognise and resist fake news [69].

Think Thief

This principle of putting a student in an attacker's shoes is sometimes called 'think thief' [8]. In research by Barth et al., University students were given the task to develop interventions by thinking like a thief. After this course, 49% of the students were confident they were less likely to fall for a social engineering attack [8]. Coti et al. recommend security academia to look at hacker competitions and the hacker mindset. They say that by learning the hacker perspective and considering the unanticipated use of technology, students will be better prepared to deter attacks and defend against them. On social engineering, they say using forethought and creativity, and educators could use human-centric competitions to great educational benefit [18].

Initial Research

Van der Linden and Roozenbeek used a voluntary in-game pre-post survey. 14.266 participants filled in the survey. Participants were asked to rate the reliability of misleading tweets and headlines on a standard 7-point scale, both before and after playing. They had to fill in 6 questions in total, 2 of which were control questions [70]. They found that the process of active inoculation through playing the Bad News game significantly reduced the perceived reliability of tweets that embedded several common online misinformation strategies [70]. They recommend further research to also explore the boundary conditions of inoculation theory by, for example, looking at the extent to which the observed inoculation effects extend beyond the game environment [70].

Further Research

The original research had a few shortcomings; there was a lack of a control group. The testing only happened within the game environment. The study only looked at reliability judgments and could not determine how confident or certain people were in their beliefs [9, 70]. In later research, they addressed these shortcomings by including a randomised control group, adding a larger battery of items, and evaluating whether the intervention also boosts confidence in reliability judgments [9]. In this newer research they employed a 2 (game vs control) * 2 (pre-post) mixed design to test the efficacy of active (gamified) inoculation [9]. They did this by having one group play the Fake News game, and one group play Tetris before and after they were shown and asked to point out realistic, but not real, fake news tweets. They had two groups of almost 100 participants each. Participants could give their perceived reliability of each tweet on a 7-point Likert-scale. They could also indicate how confident they were on a 7-point Likert-scale. The authors found clear evidence supporting the intervention and found similar effect-sizes as the original research [9].

2.4.5 Inoculation Theory and Cyber Security

In recent studies, scholars have suggested inoculation theory might be helpful in protecting humans against social engineering as an intervention tool [28, 60, 71, 80]. However, this is a new research area, and not many studies have been conducted in this area. Treglia and Delia argue that inoculation can help against social engineering via a technical tool that exposes people to social engineering attacks and help develop response and resistance [80]. However, they do not present a concrete example or solution. This study will be the first work to examine how effective active inoculation is against social engineering risks.

2.4.6 Inoculation Theory and Social Engineering

Implicit and Explicit Threats

When looking at social engineering and inoculation theory, we can identify the threat and the following counterarguing. The implicit threat of a social engineering attack is how the victim will respond to the attacker's message. The victim can provide either information or resources to the attacker, which he would never do before the attack, and he was not aware someone was able to manipulate him in doing that.

According to the inoculation theory of social engineering, the explicit threat can be seen as the messages themselves and how they contain persuasion. Arguably, attackers' psychological techniques and strategies are also explicit threats because these can be seen as forewarnings of the attack.

Counterarguing

In the fight against social engineering, counterarguing is learning to recognise a social engineering attack and learning how to react to this. People should learn to counter persuasion attempts to protect against implicit threats. Learning to recognise the explicit threats can help better know when to counter a persuasion attempt and reveal the vulnerabilities.

Chapter 3: Social Engineering Attack Scenarios and Real-life Examples

3.1 Introduction

To better understand which kinds of different social engineering attacks exist and are used, this list was composed of various attacks known and used over the years. The lists consist of two different categories; generic scenarios and real-life examples. Generic scenarios give a more broad overview of the attack vector, and do not cover one specific event and can be seen as a blueprint for an attack. The real-life examples are specific attacks or events that go into more detail of that attack. They also include more context and a time indication of when that specific attack was executed.

This list will give more insight into which social engineering techniques, as described in section 2.2.3, are commonly used in practice during social engineering attacks. In the SES various examples will be given to the player. These examples will be based on examples that are given in this list.

It is important to note that this list should not be seen as an all-inclusive, exhaustive list. It is impossible to write down all social engineering attacks or examples ever to occur because of the amount of various social engineering attacks that exist. This lists give only an indication of the broad spectrum of different attacks. The only limitation of a social engineering attack is the attacker's creativity, so it is safe to say that more and different kinds of attacks will come up in the future.

3.2 Generic Scenarios

3.2.1 Dumpster diving

What: The victim throws away valuable information, such as sensible or credit card data. The attackers go through the waste of the victim and find this information and misuse this.
Principles:

- Distraction

Source: [37].

3.2.2 Misleading the cleaner

What: The attacker pretends to be an upper-level manager who lost his key. He asks a cleaner who has full-access to open doors and gains access to the building.
Principles:

- Authority
- Kindness

Source: [37].

3.2.3 Pretending to be a cleaner

What: The attacker pretends to be a cleaner, nobody pays attention to him and he can plant a keylogger.
Principles:

- Authority

Source: [19, 37, 56].

3.2.4 Building a trust relationship

What: The attacker builds a online trust relationship (either romantic or platonic) with the victim. Once the attacker has the trust of the victim he will send photos with embedded viruses.
Principles:

- Distraction
- Kindness
- Need and Greed

Source: [37].

3.2.5 Tailgating into the building

What: The attacker dresses in the appropriate attire and pretends to work at the building. When all the employees enter the building, he tailgates an employee into the building. He now has access to the building.
Principles:

- Distraction
- Authority

Source: [29, 56].

3.2.6 Pretending to be the network administrator

What: The attacker sends an email to the subject pretending to be the network administrator and requests the organisation to provide or reset a user's password on the organisation's system. If the victim complies the attacks has its password.
Principles:

- Distraction
- Authority
- Kindness

Source: [29, 56].

3.2.7 Call from the bank

What: The attacker calls the victim claiming to be its bank, requesting information to address security concerns. The victim has to visit a webpage and enter confidential information.
Principles:

- Authority
- Kindness

Source: [56].

3.2.8 USB sticks

What: The attacker scatters interesting looking USB drives on frequently visited places by employees, such as the parking lot or smoking areas. Once the employee plugs in the USB, malware is copied onto the device.
Principles:

- Distraction
- Need and Greed

Source: [56].

3.2.9 QR codes

What: A variant of the USB sticks attack. The attacker would print and hang posters instead of leaving USB sticks. On the poster, a discount is offered for a nearby restaurant. To get the discount, the victim needs to scan a QR code, which leads to a malicious website, or a sign-up that would harvest usernames and passwords.
Principles:

- Need and Greed
- Distraction

Source: [43, 56].

3.3 Real-life Examples

3.3.1 Undercover in Nederland (Dutch TV-show) 1 - 2008

What: In the Dutch TV-Show 'Undercover in Nederland' multiple security flaws in the Dutch defence organisation were shown. In 2008 the presenter Stegeman shaved off his hair to look more like a soldier. He and a colleague were able to enter a bus that picks ups military personnel from a train station, without showing identification. Once on the base, an anonymous tipster had arranged a room to sleep in for Stegeman and his colleague, who was also equipped with military clothing. The next day he visited the airbase in military clothing and reached fighter jets and helicopters.

In a second attempt, Stegeman sat in military clothes in the trunk of a car of a colleague who entered the base for an interview. He got a pair of keys of the anonymous tipster which works on all military vehicles. Via this way, he was able to steal a military truck and drive in on the public road.
When: 2008.
Principles:

- Kindness
- Authority
- Distraction
- Time or Scarcity

Source: [77].

3.3.2 Undercover in Nederland (Dutch TV-show) 2 - 2016

What: In this episode Stegeman hid in the back of a car while wearing military clothes, bought at a military dump and camouflage. He carried the rank of lieutenant. He also got a key from an anonymous tipster again. A colleague drove the car to the entree ports behind another car with real soldiers in it. When the first car got the green light, the car with Stegeman in it drove also drove through the gate. Once in the gate, he was able to steal a military vehicle and drove away with it.
When: 2016.
Principles:

- Distraction
- Authority
- Kindness

Source: [76].

3.3.3 Undercover in Nederland (Dutch TV-show) 3 - 2018

What: In 2018, ten years after the first show, they gained access into different military bases via a transport service of the Army. The people of the show called the taxi service, but the driver did not ask for identification. Two times they entered via an automated gate, where only the driver needed to show his identification. The third time they were all asked for identification. The two people of the show did not have identification; however, they were still allowed on the premise. This way, they were transported onto the base without identification, but with a fake bomb.

After airing the show, parliamentary questions were asked and the Minister of Defence Bijleveld promised improvement [58]. The presenter of the show Stegemans was convicted for leaving a fake bomb and had to pay 2000 euro, he was not convicted for entering the premise, because he had journalistic reasons [5].
When: 2018.
Principles:

- Kindness
- Distraction
- Time or Scarcity

Source: [75].

3.3.4 Streetlab (Dutch TV-show) - 2015

What: Although this is not a real 'attack', this example does show the gullibility of people. In this short sketch, four people closed of a part of the street with safety ribbon, while wearing orange vests and equipped with hired materials. They where also able to hire a mobile toilet. They were not bothered anyone and were able to open up the street.
When: 2015.
Principles:

- Authority
- Kindness
- Herd

Source: [78].

3.3.5 419 Scam or "Nigerian scam".

What: This famous scam is often called 419 scam after the Nigerian penal code. In this scam, the fraudster tricks the victim into paying a certain amount of money under the promise of a future, and larger payoff [35]. Commonly used topics are a prince who wants to give away money or a won lottery. Before the subject can receive the money, they first have to send a small fee. Of course, they never actually receive the inherited or won money. These scam activities are often still performed manually. Sometimes the scammers involve the subject in illegal practices (e.g. money laundering) to make sure the subject is too afraid to contact the police.
When: Since 1980s.
Principles:

- Need and Greed
- Dishonesty
- Distraction

Source: [35, 73].

3.3.6 Sextortion phishing

What: In this kind of phishing the scammers send an email to their subject with a message telling their subjects they have images of them watching porn. Often they also include passwords found on the internet or old hacks. They require an amount of money, or else they will release the footage. Of course, they do not have those images, but only the threat and recognition of the password is enough for some to pay the money.
When: Unknown
Principles:

- Dishonesty

Source: [11].

3.3.7 The dishonest programmer

What: An example of the famous Abagnale; companies who were preparing for Y2K hired cheap programmers from off-shore firms from India, Russia and Taiwan [1]. While the office executives were so focused on their task (fixing Y2K bugs), they did not realise that this would be the ideal opportunity to implant a backdoor.

When: Around 1999
Principles:

- Distraction

Source: [1].

3.3.8 Mitnick - Hacking into the feds.

What: In his famous book 'The Art of Deception', Mitnick gives examples of his methods [54]. In a particular example, Mitnick pretends to be a law enforcer to a law enforcement agency. He found a copy of an instruction manual on the internet. In this manual were codes and formatting for retrieving information on criminals and crimes from the national database. Because he now knows how to access specific databases, he gains credibility, which leads to trust. With this gained trust and information, he could get information out of the database via the telephone. Especially military and police are vulnerable according to Mitnick:

> Like people in the military, law enforcement people have ingrained in them from the first day in the academy respect for rank. As long as the social engineer is posing as a sergeant or lieutenant–a higher rank than the person he is talking to - the victim will be governed by that well-learned lesson that says you do not question people who are in a position of authority over you. Rank, in other words, has its privileges, in particular the privilege of not being challenged by people of lower rank. [54]

When: Unknown
Principles:

- Authority

Source: [54].

3.3.9 RSA

What: In 2011 hackers send a well-prepared email to some RSA employees with the subject '2011 Recruitment Plan' [68]. The email was sent to the junk folder for some employees, but the email was compelling, so they still opened it. The email contained a spreadsheet, which contained a zero-day exploit that installed a backdoor via an Adobe Flash vulnerability [45]. It used a Poison Ivy RAT to gain control of the device and later the entire network. The attackers were able to steal information on the RSA SecurID system. 40 Million users received a new authentication token [42].

When: 2011
Principles:

- Authority
- Distraction

Source: [45, 68].

3.3.10 Operation "Red October"

What: Similar to the RSA attack described in section 3.3.9, in operation "Red October" attackers used spearfishing to install malware on the device of the targets [41]. The targets were different diplomatic, governmental, and research organisations [45]. Attached to the email was malware that exploited Microsoft Office security Vulnerabilities. This operation was operational for six years, and they were able to steal enormous amounts of sensitive data and credentials.

When: 2007 - 2013
Principles:

- Authority
- Distraction

Source: [41, 45].

3.3.11 Virtual kidnapping

What: There are multiple scenarios for this scam. In the first scenario, scammers search for their victim's personal information or place spyware on their devices. If they know the victim is on a trip, they will call their loved ones and say they have the victim hostage and demand a ransom as soon as possible. This is fake, but because they have much personal information, this may seem credible. In the second scenario, students who study overseas are targeted. Scammers pretend to be authorities from their motherland and mask their location and number to be the same as the real authorities. They threaten the victim or the family of the victim. The victim is demanded to rent a hotel and take a photo of themselves tied up and blindfolded. These photos are then sent to the family overseas. The family is demanded to pay a ransom.

When: Since at least 2017
Principles:

- Authority
- Time or Scarcity
- Need and Greed

Source: [48, 83].

3.3.12 Malware parking tickets

What: Attackers made yellow fliers to look like parking tickets, urging people to visit a specific website. When victims visited that website, a DLL was installed into System32. The DLL was installed as an Internet Explorer Helper Object. This would pop-up a fake security alert and would redirect the user to a malicious site, where the victim would be asked to install a fake anti-virus scanner, which was malware in disguise.

When: 2009
Principles:

- Authority
- Distraction

Source: [56, 85].

3.3.13 Francophoned

What: The vice-president of a French multinational company received an email with an invoice hosted on a file sharing service. She also received a phone call from supposedly another vice president of the same company, instructing her to process the invoice, spoken in perfect French. The invoice was fake, and the other vice president was an attacker. The invoice was a RAT which contacted a Command-and-Control (C&C) server in Ukraine. Via this way, the computer was immediately taken over.
When: 2013
Principles:

- Authority
- Distraction
- Time or Scarcity

Source: [38, 56].

3.3.14 CEO fraud

What: The Dutch CEO of Pathé received multiple emails from attackers claiming to be presidents from the French mother company. The emails spoke of a foreign takeover of the company. They requested multiple transactions, which needed to be kept secret. In the end, more than 19.2 million euro was wired to the attackers. The CEO resigned, and the CFO was fired.
When: 2018
Principles:

- Authority
- Distraction
- Time or Scarcity

Source: [72]

3.3.15 Microsoft Tech Scam

What: A famous scam running since at least 2008. Attackers call their victims, claiming there is something wrong with the victim's computer that they can fix. Often they convince their victims by letting them use simple Windows commands. For example, they let their victim runs the `assoc` command and claims that all things printed are malware traces. They can also let the user open the Event Manager, which will always show dozens of errors. Once the victim is convinced their computer needs a fix, the attack will let the user install a program like Team Viewer and give the attack full access. Once he has access, he might lock the system, ask for a payment, or steal the victim's computer's entire contents.
When: Since 2008
Principles:

- Authority
- Distraction
- Kindness

Source: [6, 49, 53]

3.3.16 Whatsapp Fraud

What: The attacker claims to be a friend or family member of the victim with a new phone number. He claims to have a financial problem, like an invoice that urgently needs to be paid. The attacker can have stolen information from social media to be more believable. Because the victim thinks they are helping a friend, they are happy to help. If the victim pays, often more payment requests follow.
When: Since 2010s
Principles:

- Distraction
- Kindness

Source: [62]

3.3.17 SMS Fraud

What: The attacker sends an SMS to their victims in the name of the bank. Sometimes they can spoof the number, so the number matches the bank's real number, and the messages are mixed between the real messages. The messages often contain a warning, such as losing access to the account when the victim does not handle it quickly. The message contains a malicious link, or the attacker asks to send security codes.
When: Rapid increase since 2019
Principles:

- Distraction
- Authority
- Time or Scarcity
- Need and Greed

Source: [65]

Chapter 4: Design

4.1 Inoculation Theory and the Social Engineering Simulator

4.1.1 Introduction

This research will be the first to use inoculation theory as an intervention method against social engineering risks, which various scholars suggested as a feasible method [28, 60, 71, 80]. This research hypothesizes that gamified active inoculation can be used in the same way against social engineering risks as it was used against fake news in The Fake News Game, because both social engineering and fake news use persuasion to influence human behaviour. The SES shall use inoculation theory in the same way as The Fake News Game and show different social engineering attacks strategies. It is also broad-spectrum and focuses not only on one kind of social engineering attack but also on all kinds of attacks and the tactics used by social engineers. The SES has a scenario editor, which can create and customize interventions. For this research, only one intervention will be used. The intervention can roughly be divided into three parts. The first part and third part contain examples of social engineering attacks and passive inoculation. The second part contains the active inoculation and learning the different techniques used by social engineers.

4.1.2 Passive Inoculation

The first part is an introduction with interactive examples of social engineering attacks. The user is prompted with imitations of either email, phone messages, or conversations, which are either legit or malicious. The user can choose between three options: trust, suspicious, dangerous. If the user chooses to trust a prompt, they will continue the simulated interaction, if possible. The option 'suspicious' will also continue the interaction, but with extra care. If the user chooses the option 'dangerous', the interaction will stop. The last part will also contain examples that are presented in the same way. This setup has two intended goals; first, it trains the user via passive inoculation, and second, it can provide insight into the intervention's effectiveness.

These examples can be seen as a passive way of inoculation. The user is presented with small pieces of persuasion in the form of social engineering attacks. Seeing and interacting with these small pieces generates counterarguing, as explained by inoculation theory.

Besides training via passive inoculation, these examples indicate the user's risk profile and the intervention's effectiveness. The trust, suspicious, dangerous system was inspired by the Dutch driving exam, where the candidate has to recognize dangerous situations in the car. Here, the candidate must choose either doing nothing, take one's foot off the gas or braking. The advantage of using such a system is making the answers to the examples less binary. Every choice the user makes can be logged and analyzed. This means that it is possible to see which persuasion tactic the user is most likely to fall for. Because there are three answers, it shows whether the user was fully confident an answer is correct or was more doubtful.

It is also possible to compare the third part with examples with the first part of examples. This difference can indicate the effectiveness of the second part, which contains the active inoculation. When a user scores better in the third part, it can indicate that the active inoculation had positive results. The examples also contain legit cases to check for false positives and see if they have not become needlessly sceptical. These results can indicate but are not entirely accurate because they result from a safe test environment. This is why to test the effectiveness truly, a mock attack should be performed on the participants and a control group.

4.1.3 Active Inoculation

The second part of the SES contains the active inoculation. In this part, the player will be placed in a social engineer's shoes, comparable to how The Fake News Game places the player in a fake newsmaker's shoes. This way, the player is actively learning how social engineering attacks are built up. The player can also see where

their 'victim' could have defended itself. When playing this part, the player has to actively come up with ways to use social engineering. Involving the user actively stimulates the production of counterarguing. In this part, the player will also learn the various social engineering persuasion principles used. Besides just telling the player what the several principles are, the principles will be explained via interactive sections or games. This way, the player will feel how a victim would feel when such persuasion is used. These sections also add an extra layer of active inoculation. The principles will be selecting the combination of principles coined by Cialdini and Stajano and Wilson as described in section 2.2.3 [14, 73]. The following principles will be shown in the SES: distraction, authority, herd, kindness, need and greed and time or scarcity. The player will, comparable to The Fake News Game, earn achievements when they complete a principle, to add an extra gamification layer.

4.2 Game Flow

The flow of the game is decided by the scenarios made in the scenario editor. These scenarios are called stories. The scenario editor will be explained in more detail in section 4.4. The game always offers the user a prompt, and which prompt is shown next depends on the user's answer. Prompts are always pieces of content with answers. The content can be things like texts, images, emails, or phone messages. The only exception is the principle elements, which are the interactive sections that show the social engineering principles. These sections can take over the entire screen and do not follow the standard content plus answers design. During these sections, the player does not move from prompt to prompt, but the sections alter the game flow. A score is kept and shown to the user to add an extra layer of gamification. It is also possible to add a timer to prompts for extra time pressure. When the player starts part 2, the credibility meter is shown beside the game score. This meter goes down if the player chooses unbelievable answers and goes up when he chooses credible answers.

4.3 Scenario Design

As described earlier, with the scenario editor, it is possible to create an endless amount of different stories and interventions. For this thesis, only one design was created. This design was used in the experiment to test the effectiveness of the SES. The playtime of this scenario is around 45 minutes. The design of the scenario will now be explained.

4.3.1 Part 0

Introduction section

The game starts by asking the player to agree to the legally obligatory privacy statement. After the player agrees, he is first playfully introduced to the main mechanics of the game, choosing between different answers. The game tries to mimic a person to give the impression the player is talking to the game. Here the questions or main texts are things the game says to the player, and the answers are things the player can say to the game. For example, the first prompt is *'Hi! And welcome to this game!'*. The player can choose the answer either *'Hello!'* or *'Ehm, who are you?'*. The first prompt can be seen in figure 4.1. Via this way, the player learns his answers are things he can say back to the game. This interaction style is continued throughout the entire game.

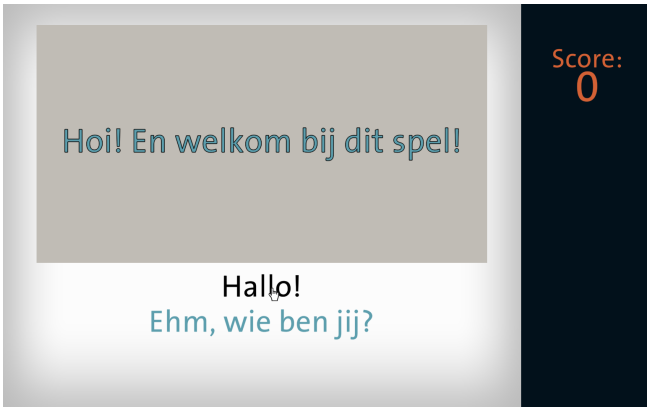


Figure 4.1: The first prompt the player sees while playing the Social Engineering Simulator.

Real world examples

The rest of the introductory section tells the player what the goal of the game is. It does this by explaining what social engineering is. The game gives an email prompt of a very obvious phishing mail, based on the 419 scam (section 3.3.5). The email prompt will be further elaborated in section 4.5.6. This is an example that most players will know, which makes the term social engineering less abstract. Besides being an example of a known phishing example, this prompt also introduces the player to a new game mechanic, namely interactive emails. The text invites the player to hold the cursor over the link in the email, something the player will need to do more often in the game. After the phishing mail, the game explains more elaborate forms of social engineering by given the example of CEO fraud (section 3.3.14) and virtual kidnapping (section 3.3.11).

Collecting research information

The final part of the introductory section is collecting information about the player. First, the name of the player is asked. This name is only used within the game and not saved, so the player can either fill in a real or a fake name. The following questions are about the information used in this research. The following information is asked; age and on a scale from 1 to 10, how safe does the player think his internet behaviour is.

4.3.2 Part 1

In this part, the player will look at various examples and answer if they are dangerous, suspicious or safe. After each example, the player will be told which answer was the right one. If the answer was suspicious or dangerous, the player will be told how they could have recognized the danger. The player scores points for each correct answer and loses points for each incorrect answer.

Example 1

The first example is a legit email. It is a confirmation of a subscription for an online course, with a link to confirm the account. Although the email looks a bit suspicious, this email was a copy of an actual email sent by an educational institution of the Dutch Military.

Example 2

The second example is a malicious phone call. The player is in a phone conversation with someone who claims to be a lieutenant of the DCC. He says the player needs a software update. Although this a strange conversation, it would be too early to call it dangerous, so clicking dangerous already would not be the right answer. If the player continues the conversation by clicking suspicious or safe, the lieutenant says the player has to download software from an obvious fake site and says this will be checked soon. This example is based on the 'francophoned' scam (section 3.3.13). The principles of authority and time of scarcity are used.

Example 3

The third example is again a legit email. In this email, the ICT organization of the Dutch Military asks the player to change his password in the digital work environment.

Example 4

The fourth example is a malicious phishing email. The email sender is the commander of the armed forces. However, the email address ends with malicious `@mindeff.nl` instead of the legit `@mindef.nl`. The email itself is a mourning notice with a link to the funeral card. The link refers to *www.mindeff.nl/rouwbericht-generaal-middendorp.pdf.exe*. The player could recognize here again the misspelling of the word 'mindef' and the link ending on '.exe' instead of '.pdf'. This example is based on a generic spear phishing scenario. The principles used are authority, kindness, and distraction.

Example 5

The fifth example is a malicious phone text. The example starts with a random number saying the sender is the player's dad with a new phone number. The player could already click on dangerous. If the player continues the conversation by clicking suspicious or safe, they send a text asking for money. The player should now click on dangerous. This example is based on the WhatsApp fraud scenario (section 3.3.16). The principles used are kindness and authority.

Example 6

The sixth example is a legit phone text. The player receives a text from someone claiming to be a lieutenant who got the player's phone number from the digital work environment. He wants to move the meeting to tomorrow and ask him to call if the player has any questions.

Example 7

The last example is a malicious phone conversation. The player is confronted by someone who claims to work for the ING Bank and says they are under a cyber attack. If the player continues the conversation, a timer of 10 seconds starts, and the person from the ING Bank says they have to handle it quickly. If the player continues the conversation, the person from the bank says he will prove he is really from the bank by saying the correct amount of money the player has. After that, he will ask for a code on the phone of the player. This is based on the "call from the bank" scenario (section 3.2.7). The principles used are time or scarcity, authority and need and greed.

4.3.3 Part 2

Introduction

In this part, the active inoculation will occur, and the techniques, as described in section 2.2.3, will be explained. The player will be placed in the shoes of a social engineer and has actively come up with ways to use various social engineering strategies and techniques. This is explained in the introduction of part 2. The player is told he is now in service as a social engineer of the fictional country 'The Union', whose arch-rival is The Netherlands. The President of The Union wants to invade The Netherlands, and the player has to make sure this is possible. Part 2 follows the stages of an advanced social engineering attack by first collecting information and slowly building up the attack. A different technique will be explained in each stage, and the player has to actively use this newly explained technique. At the end of the stage, the player earns an achievement to show he now has mastered this technique.

Also, at the beginning of this part, the credibility meter will be visible. This meter is an extra layer of gamification and shows that a social engineer should always be credible. When the player makes unbelievable answers, the meter will go down. The credibility meter alters the amount of score the player earns per answer. When the meter is down, the player earns fewer points per answer. When the meter is up, the player earns more points per answer. When the meter is below a certain threshold, it starts blinking with a warning sign. The credibility meter can be seen in figure 4.2.

(a) The credibility meter.

(b) The credibility meter starts blinking when its value becomes low.

Figure 4.2: The credibility meter shown in the left side bar of the Social Engineering Simulator.

Need and Greed

The first technique that will be explained is the 'need and greed' technique. This technique starts with the interactive section, which will be further explained in section 4.6.2. This interactive part is meant to give the player the feeling of the *need and greed* technique and is a form of inoculation. After this interactive part, the technique will be further explained. After that, the player has to choose his first step, where the correct answer would be *collecting information*. After this choice, the player can choose from which social media he wants to collect information. This choice does not affect the gameplay but does give the player the feeling he affects the course of the game.

Next, the player can choose from which person he wants to collect the information, which is again only a visual choice. This person becomes the target. If the player has chosen a person from whom to collect information, the game explains an idea to collect information. The idea is to hang posters on the work environment of the

target, which is based on the "QR codes" scenario (section 3.2.9). The player has to choose which poster to hang. With this choice, the player has to choose the poster that uses the *need and greed* technique. This is a poster of a lottery for workers, with a jackpot of €1000. This choice enables the player to actively link a possible attack to a social engineering technique, which is, in turn, a form of active inoculation. After the player chooses the suitable poster, the game says the poster action was a success, and the target has sent his information. Now the player earns the *need and greed* achievement.

Authority

The player has now collected the information of his target in the game. The game explains that he should now try to break on a military basis with the help of an access card. To get this access card, the player has to lure his target to a physical location. The player has to choose how he wants to do this. The idea is to lure him via an invitation letter. The player has to choose who will be the sender of this letter, choosing between a high or a low rank. After that, he has to 'call' the target. He has to choose credible answers in this conversation while also pretending to be a high ranking officer. Via this way, the player learns while playing that his victim listens to the high ranking officer's authority, which is also a form of active inoculation. This attack was loosely based on the 'francophoned' scam (section 3.3.13). After this section, the game will explain the *authority* technique a bit more. In the end, he earns the *authority* achievement.

Herd

This part is again introduced by an interactive section, which is a little game the player will be playing. This game will be further elaborated in section 4.6.4. The game should give the player a feeling of the *herd* technique, which is a form of inoculation. After the game, the *herd* technique is further elaborated. Next, the game continues with the story, where the target will come to the player for a checkup of the access card. To make sure the target gives his access card, the player has to use the *herd* technique. He can do this by first letting some other colleagues give away their passes. These colleagues are also people who work for The Union. This is based on a street scam observed by Stajano and Wilson [74]. After the player receives the access card of the target, the game shows an image of a Proxmark3 and says this device makes a copy of the access card. The player now earns the *herd* achievement.

Kindness

This part starts by introducing the *kindness* technique. It also explains that during most attacks, not only one technique but multiple techniques are used. The game asks the player what the *authority* technique is, so the player is stimulated to not only focus on the current technique but also keep remembering the previous techniques. Next, the player has to set up the 'pretending to be a cleaner' attack scenario (section 3.2.3). They do this because the player has no access to the military base via the obtained access card of the previous section, but they also want to break into the commander's office, for which they do not have access. The player has to recognize that he can open a door by acting as a friendly cleaner. After he recognizes this, he has to act like the cleaner to open the door. This is a form of active inoculation. When he has talked his way inside, the game shows an image of a keylogger and says the keylogger is now planted at the keyboard of the commander. The player now earns the *kindness* achievement.

Time or Scarcity

An interactive section introduces the following part in the form of a game, which will be further elaborated in section 4.6.1. After the interactive section, the story continues. Via the keylogger, the player has now obtained the password of the commander. With the use of this password, the player can now send emails on behalf of the commander. The goal is now to take over the entire network with a RAT. This will follow the attack of the 'francophoned' attack (section 3.3.13), be first sending an invoice, and later calling to put on time pressure. The player must first recognize that an invoice can put on time pressure by selecting the correct email. Next, he has to make the call and select the correct answers to increase the time pressure. Selecting the email and the correct answers in the phone conversation is a form of active inoculation. When the player has finished the phone call, he earns the *time or scarcity* achievement.

Distraction

The last technique that will be covered in the SES is also introduced in the form of an interactive section in the form of a game. This game will be further elaborated in section 4.6.3. The *distraction* technique is one of the harder techniques to understand and is always used in combination with other techniques. This is why this technique is covered as last. In this last part, the player tries to take the entire military network offline. This

can be seen as the big final, where the player has to use all his newly learned abilities by combining multiple techniques. Because of the previous sections' work, the player can now send emails on behalf of the network administrator. He can now send an email asking everybody to change their passwords based on the "pretending to be the network administrator" scenario (section 3.2.6), because the network is 'hacked'. In this scenario, the victims would be distracted by the fact they are 'hacked' they no longer focus on whether the email is real or not. When the player selects the right email, the game tells the player he has shut down the entire network. The Union can now freely attack The Netherlands. The player now earns the final *distraction* achievement.

Epilogue

The game ends by repeating the techniques the player learned while playing this part of the game. The game now warns the player to be vigilant for these techniques and warns the player to recognize these techniques better. If the player recognizes these techniques being used, the player should perform a double-check, for example, by video calling. The game recommends the player when in doubt not to accept a request.

4.3.4 Part 3

Introduction

In the third part of the SES, the game again presents examples to the player, where the player has three options; dangerous, suspicious or safe. This is explained to the player in the introduction of the third party. The credibility meter is hidden because the player does not have to act like a social engineer anymore.

Example 1

The first example is a spear-phishing example. The email tries to emulate an invoice from the energy company Vattenfall. However, the email ends with `@e-mail.vattenfalll.nl`. The email says there is an overdue invoice, which needs to be paid as soon as possible. The link in the email refers to the malicious site `vattenfalll.nl`. Techniques used are: *authority* and *time or scarcity.*

Example 2

The second example is a legit email. This email is sent by a colleague, who refers via a link to the annual calendar on the internal work environment.

Example 3

The next example is a real-world example of SMS fraud (section 3.3.17), received in December 2020. The senders' number has been spoofed like the SMS was send by ING Bank. The SMS text says that online banking access will be restricted if the receiver does not handle it quickly by clicking on a malicious link. Techniques used are: *authority*, *time or scarcity*, *need and greed* and *distraction*,

Example 4

This example is a phone call where the caller says he is from the telecom company KPN. The caller says the player has to quickly pay an invoice that is still open. The caller gives a malicious link. A timer of 20 seconds is started to stimulate the real pressure of a phone call. Techniques used are: *authority*, *time or scarcity* and *distraction*,

Example 5

The following example is a legit WhatsApp conversation, which looks much like WhatsApp fraud. The player receives a WhatsApp message from someone in their contact list. The message uses friendly text like *"hey buddy!"*. The sender also says he has a problem, and he asks for money. This example is an essential measurement of whether the player has become too sceptical or not. Because nothing dangerous is asked yet, the player should not press dangerous just yet. When the player selects either suspicious or safe, he reacts to the messages *"I think this a strange request, can we video-chat?"*. The sender responds *"Yes, of course! No problem."*. This simulates a normal conversation, and an attacker would not be able to video-chat. Thus the player does not have to select the dangerous option.

Example 6

The next example is a malicious interaction between the player and someone on the street. The person introduces herself as a college student who wants to asks some questions. The first questions the age and living place of the player. The following questions are already a bit more intrusive, by asking which social media the player is active and what the usernames of the player are. The next questions are the last three numbers of the IBAN and social security number. This example gives a good insight into what moment exactly the player stops a conversation. Techniques used are: *kindness*, *commitment and consistency* and *distraction*,

Example 7

The last example is an obvious sextortion phishing email (section 3.3.6). In the email, the sender claims to have the player's password and says he caught the player looking at porn. This email is based on the *dishonesty* technique, which was not covered in the SES. This might provide interesting data to see how players react to a technique that was not covered in the game.

Epilogue

The game ends with providing the score and commenting on it. If the player scored above the threshold, it would compliment the player. If the player is below the threshold, it will encourage the player to try harder next time. It also asked how much the player enjoyed the game on a scale of 1 to 10, how educational the player think the game was, and how quick the player will fall for a social engineering attack now. The end screen says this game is inspired by The Fake News game and links to that game.

4.4 Scenario Editor

In the scenario editor, the sequence of prompts can be made, which the SES will follow, these are called 'stories'. As explained earlier, the user moves from prompt to prompt. The scenario editor is represented similarly. In the scenario, editor flowcharts can be made, where every node is a prompt, and every edge represents to which prompt the story moves next. There are three types of nodes; Normal nodes, principle nodes and technical nodes. Normal nodes are the default kind of prompts with content and answers. Principles nodes are the interactive section that takes over the game flow for a short time. Technical nodes do not show something on the screen but can, for example, alter the flow or the score of the game. In a node, answers can be added, and those answers can be linked to other nodes, representing which prompt the story should continue when that answer is selected. Depending on the node type, different variables can also be altered in the node. An example of a story made in the scenario editor can be found in figure 4.3. In the scenario editor, it is also possible to add exposed variables. These variables are strings that can be given a value, or the user can add a value to this. It is possible to call these variables in a string in a prompt. In this way, the text on the screen can be customized to show the players' name.

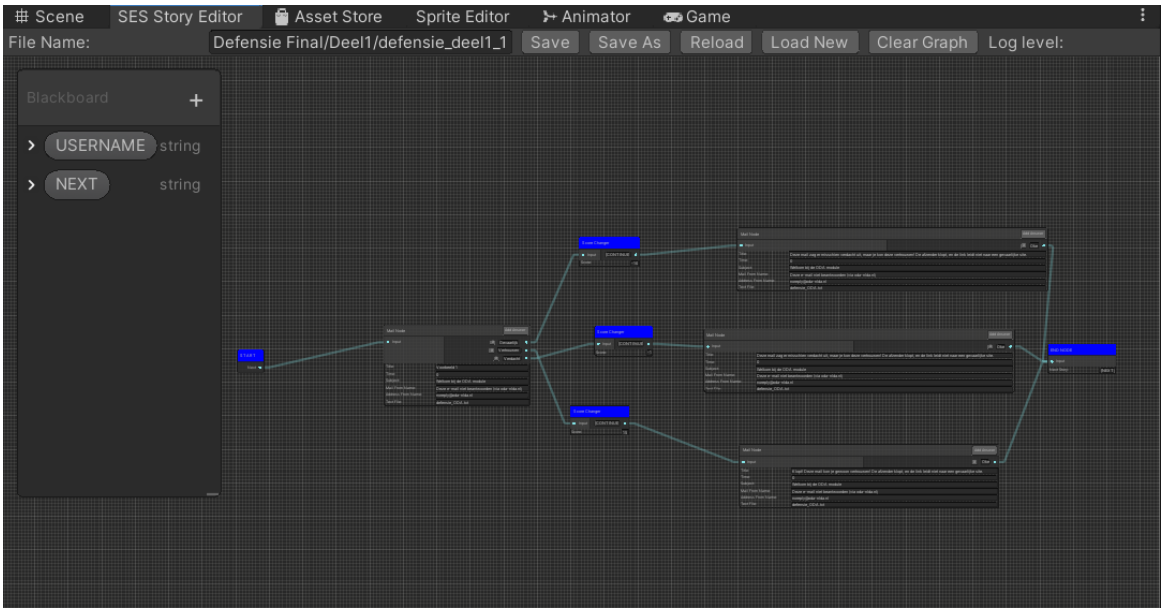


Figure 4.3: An example of a story made in the scenario editor of the Social Engineering Simulator.

4.5 Normal Nodes

Normal nodes are the nodes that form the backbones of a story in the SES. They always have at least a title and at least one answer. A timer can be set to put on time pressure.

4.5.1 Story Node

This is the most default prompt. This node can be used for all kinds of situation, for example, showing information or asking simple questions to the player. The prompt shows some text, and the user can choose between different answers. The story node and the corresponding screen can be found in figure 4.4.

(a) The story node.

(b) The screen of a story node.

Figure 4.4: The story node and the corresponding screen of a story node.

4.5.2 Image Node

The image node is almost identical to the story node, with the addition that an image node can also contain an image. The image is prominently shown on the screen. This can be used to make the text more clear or give the prompt more energy. The image should be saved in the corresponding map and referenced via the `image` variable. The image node and the corresponding screen can be found in figure 4.5.

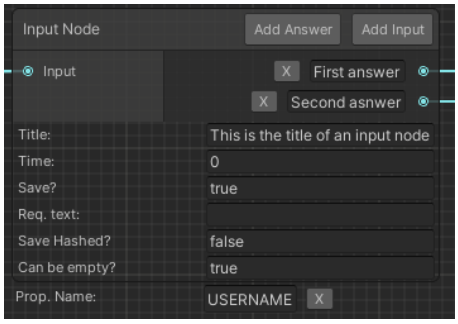
(a) The image node.

(b) The screen of an image node.

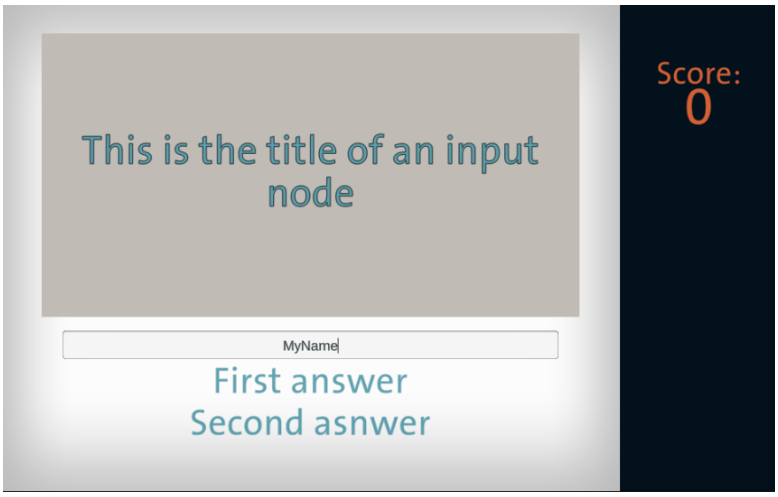
Figure 4.5: The image node and the corresponding screen of an image node.

4.5.3 Input Node

This node can be used to get written information from the user. The input node has a title, an input field and can have different answers. This node works with the exposed variables; the input the user gives alters an exposed variable. For example, first, an input node asks the player for his name and alters the exposed variable `USERNAME`. Later, the game refers to the `USERNAME` variable to use the player's name in the game. An example of this will be given in section 4.5.5. The input node has a few other variables. The `save` variable defines if the given input should be saved. Besides, the `save hashed` variable defines if the input should be saved as a hash. This can be used to save sensitive information. The `can be empty` variable defines if the given input field can be left empty. The `required text` variable can be used to require specific text. For example, a "@" string can be given if the input should be an email address. The input node and the corresponding screen can be found in figure 4.6.



(a) The input node.

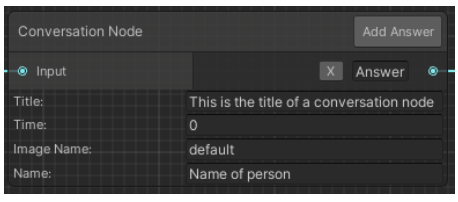


(b) The screen of an input node.

Figure 4.6: The input node and the corresponding screen of an input node.

4.5.4 Conversation Node

The conversation node can be used to represent conversation held in person or via the phone. The node has a title, which can be used for text the person says. It also has an image, which can be used to represent how the person looks. For phone calls, an image of a phone can be used. A subtitle can be added to represent the name of the person the player has a conversation with. The conversation node and the corresponding screen can be found in figure 4.7.



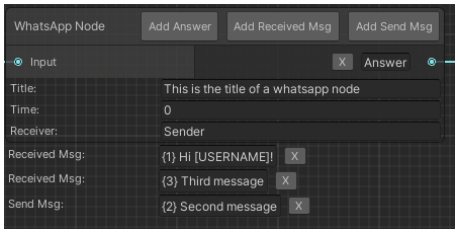
(a) The input node.

(b) The screen of a conversation node.

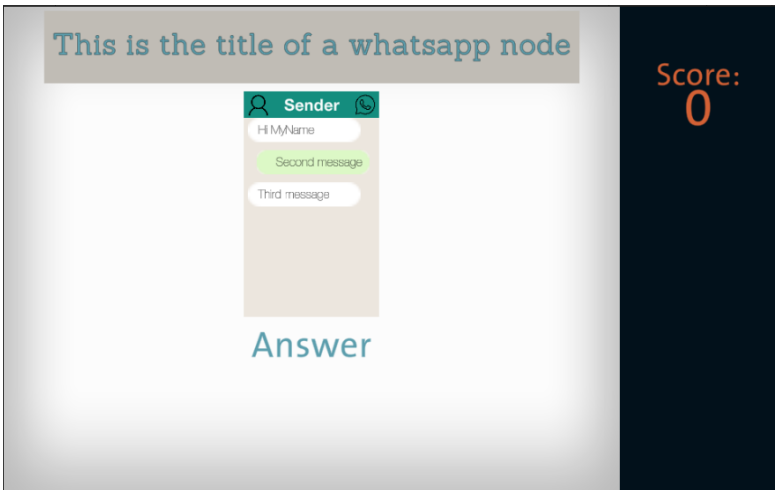
Figure 4.7: The conversation node and the corresponding screen of a conversation node.

4.5.5 Whatsapp Node

The WhatsApp node can be used to represent WhatsApp or SMS conversations. The look of the screen is modelled after a WhatsApp conversation. The sender of a message can be adjusted. There is also an indefinite amount of messages that can be added; overflowing messages will be represented via a scroll view. The WhatsApp node and the corresponding screen can be found in figure 4.8. Figure 4.8a shows that the first message refers to the exposed variable `USERNAME`. This variable was set by the input node (section 4.5.3). In figure 4.8b, it can be seen that this variable has been replaced by the value 'MyName', which was filled in during the input node screen.



(a) The whatsapp node.



(b) The screen of a whatsapp node.

Figure 4.8: The whatsapp node and the corresponding screen of a whatsapp node.

4.5.6 Mail Node

The mail node can be used to represent received emails. This screen is modelled after the Gmail environment. The subject, sender and sender address of the email can be adjusted. In this case, the sender name uses the exposed variable `USERNAME`. The mail node also supports hyperlinks in the email. If the player hovers his mouse over the hyperlink, a text will be presented at the bottom of the mail. The content of the text should be saved in a separate text file. The mail node and the corresponding screen can be found in figure 4.9. The text file of that figure contains the following text:

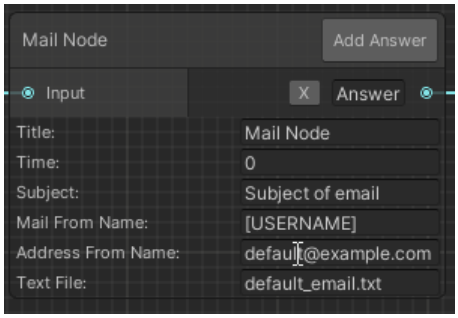
```
This is the text of a default email.

<hyperlink "Link" "www.example.com">

With kinds regards,

Name.
```

4.6 Principles Nodes

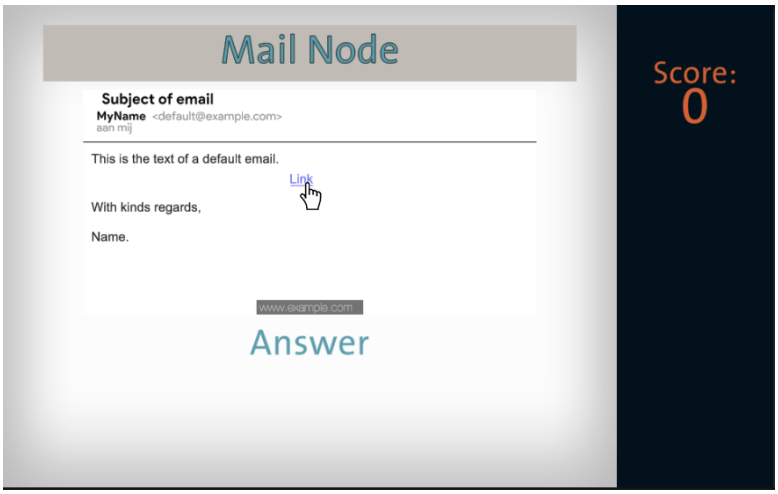
The principles nodes are the interactive sections meant to let the player feel the various social engineering techniques. The principles nodes take over the flow of the game completely. When the section is completed, it continues with the normal flow. The principles nodes are coloured red in the scenario editor.

4.6.1 Principle of Time Node

This node intends to give the player the feeling of time pressure for the *time or scarcity* technique. The user plays a game where he has to look at different passports while being under time pressure. A timer is very clearly ticking down and makes noise. The user has to look at passports and try to see whether they are correct or false by looking at tiny details. The user either presses the green 'correct' or the red 'false' button. A different sound plays if the answer is right or wrong. The gameplay can be seen in figure 4.10b. When the player chooses for 'correct' while the passport contains a mistake, the game will highlight this mistake (see figure 4.10c). At the end of the game, a resolution screen will be shown, showing how many answers were correct and how many were incorrect (see figure 4.10d).



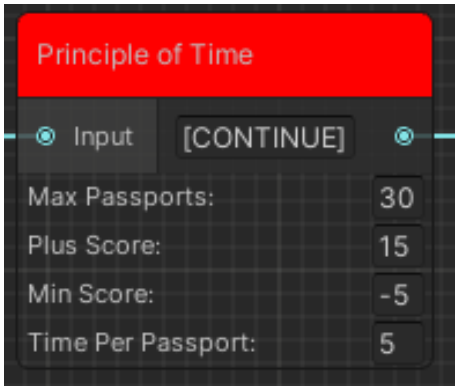
(a) The mail node.



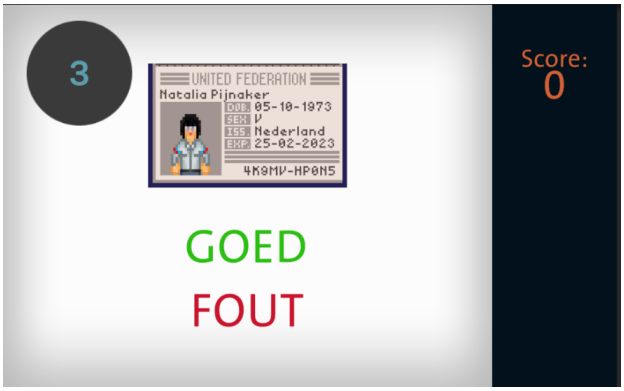
(b) The screen of a mail node.

Figure 4.9: The mail node and the corresponding screen of a mail node.

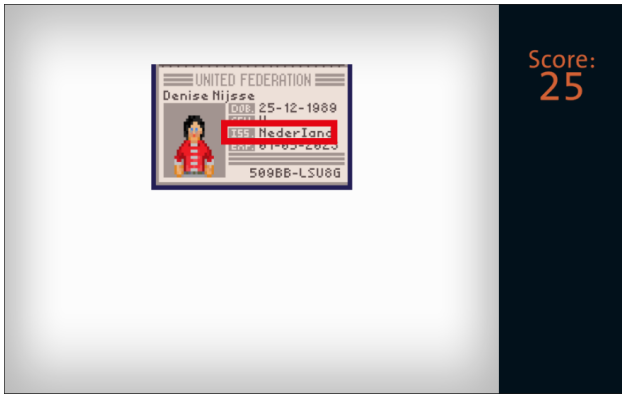
A few variables can be altered. The number of passports the game should use can be altered. The score a player gains for correct answers and the number of points a player loses can be altered. Furthermore, the time per passport can be altered. The node can be seen in figure 4.10a. New passports can easily be added; this can be done by adding the image of the passport into the right folder. The passport name should contain whether it is a correct or an incorrect passport. If the passport is incorrect, it should also contain which part is incorrect so that the game can highlight the right area. During gameplay, the game will select the given amount of passports randomly. For the SES 18 incorrect and 33 correct passports were added.



(a) The principle of time node.



(b) The principle of time screen during gameplay.



(c) The principle of time showing a mistake.

(d) The resolution screen of the principle of time.

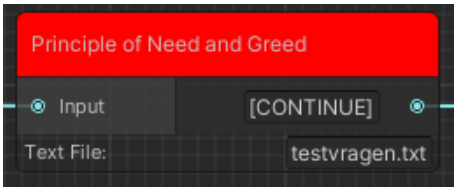
Figure 4.10: The principle of time node and the corresponding screens of the different stages.

4.6.2 Principle of Need and Greed Node

This node is meant to give the player the feeling of the *need and greed* technique. Because this technique plays into the player's desires, a game would not truly deliver this feeling. To truly play into the desires of the player, he is promised real money. The player is told he participates in a survey of the CBS into online shopping behaviour. The player is promised money for every correct answer but can also always choose to skip a question. The amount of money is always prominently shown on the screen to increase the desire of the player. The survey can be seen in figure 4.11b. At the end of the survey, the player is told no actual money was to be earned, and no data was saved. Also, the player is confronted by which data he would give away for which amount of money (see figure 4.11c). Although these questions might seem innocent on their own, these combined can be used for identity fraud. It is important to note that even though not all players might believe they would receive the money, the feeling of the *need and greed* technique can still be perceived.

In the node itself, only the questions can be altered. The questions should be given in a text file. The text file should contain the question to be asked, the amount of money the player could get for the question, and which data is obtained (for the resolution screen). For the SES the text file contains the following (translated from Dutch for clarity):

```
Welcome, on behalf of Statistics Netherlands (CBS) we would like to ask you some questions.
What is your name?;0.2;your name
How old are you?;0.2;your age
At which shop did you last buy something online?;0.25;your last online store
What was the last thing you bought online?;0.25;your last purchase
How expensive was your last purchase?;0.25;the price of your last purchase
Where do you work?;0.5;your work location
How many colleagues do you have?;0.5;the amount of colleagues you have
What is your work email address?;0.5;your work email
What is the password for that?;1;your password
```



(a) The principle of need and greed node.

(b) The principle of need and greed screen during the survey.

(c) The principle of need and greed showing the resolution screen.

Figure 4.11: The principle of need and greed node and the corresponding screens of the different stages.

4.6.3 Principle of Distraction Node

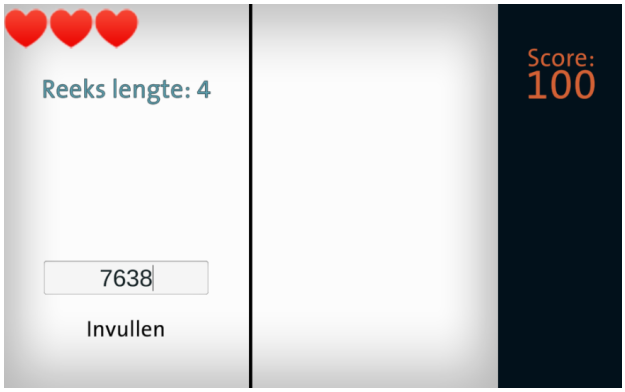
This node is to designed give the player the feeling of the *distraction* technique and is loosely based on the Distraction card game made by Thinkfun [67]. The games consist of two phases. The goal of the game is to remember a sequence of numbers. The game starts with a sequence of zero numbers, which slowly increases. A number on the screen is shown, together with a sound effect (see figure 4.12b). This number is added to the sequence. For example, first, the number '1' is shown, next the number '2' is shown, the sequence is now '12'. In the first phase, after every two number, the sequence is asked to check whether the player still remembers

the sequence (see figure 4.12c). In each phase, the player starts with three lives, and one life is subtracted for every mistake. When the player is out of lives in the first phase, a resolution screen is shown with the sequence length, and the player continues to the second phase (see figure 4.12e). In the second phase, the sequence is reset to zero, and the gameplay is mostly the same. However, now between every two number, the player is asked to fill in a simple maths sum (see figure 4.12d). Because the player is distracted by the math sum, which also requires using numbers, it is much harder to remember the correct sequence. After the second phase, a resolution screen is shown, with the maximum sequence with and without distraction and the number of sums filled incorrectly.

In the node itself, nothing can be altered (see figure 4.12a).

(a) The principle of distraction node.

(b) The principle of distraction node showing a number the player should remember.



(c) The player is asked to fill in the sequence during the principle of distraction gameplay.

(d) In the second phase of the distraction game, the player is also asked to fill in simple math sums.

(e) The first resolution screen of the principle of distraction.

(f) The second and final resolution screen of the principle of distraction.

Figure 4.12: The principle of distraction node and the corresponding screens of the different stages.

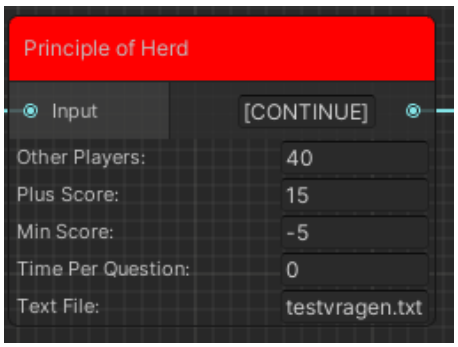
4.6.4 Herd Node

This node is designed to give the player the feeling of the *herd* technique. During this game, the player is given questions with two answers; one is correct; the other is not. The player controls a cube and can move to the left or the right, representing the two answers (see figure 4.13b). When the player moves to one side, he can press on the spacebar to fill in his answer, and he sees whether the answer is correct or not (see figure 4.13d). On the screen, there are also little transparent cubes. The player is told his behaviour is recorded, and the little cubes are other players' behaviour (see figure 4.13c). However, the game does not record the player's behaviour, and these little cubes are only programmed to mimic other players' behaviour. The little cubes are set to go deliberately to the wrong answer after a few questions. If the player does not know a particular question, he might be pressured to follow what he thinks is the other players. This will create a feeling of following the group. At the end of the game, a resolution screen is shown with the amount of correct and incorrect answers (see figure 4.13e).

The node itself has a few variables that can be altered (see figure 4.13a). The number of other players can be set. The amount of score a player gets for a correct answer and the amount subtracted for an incorrect answer can be set. Also, a timer can be set to put pressure on the player. However, during testing this principle, it turned out that the player focused less on the other players and more on the timer with a timer. Thus, creating the *time or scarcity* technique, instead of the *herd* technique. The questions are set via a text file. The text file also requires a few other variables. The text file requires the following information, in this order: the question text, the left answer, the right answer, which answer is correct (either L or R), the percentage of players moving to the correct answer, the percentage of players moving to the wrong answer, the maximum waiting time of the other players, and the chance a player will make a fake move. The other players are programmed to wait a random amount of time; the maximum waiting time will put a maximum on that wait time—they then move a certain random time to a side. The percentages given in the text file determine how many move either the left or the right side. Some will also make a faker move, which means they will first move a random time to the wrong side and then turn back and move to the correct side. This will make the other players' movement more realistic. The chance of a faker move is given in the text file as one in the given number. Thus a higher number means a lower chance. For example, if the given number is 100, the chance of a faker move is 1%.

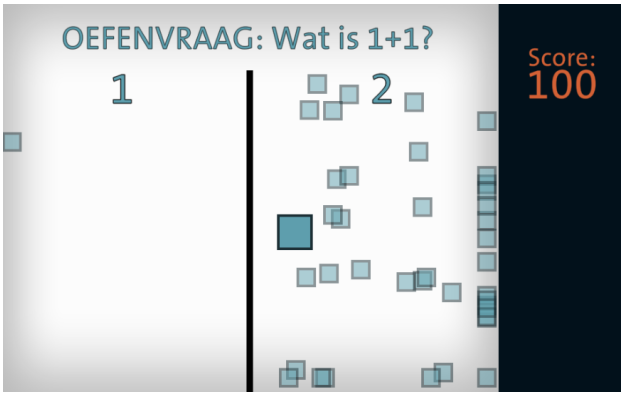
In the SES the following questions and variables are contained in the text file (translated from Dutch for clarity):

```
PRACTICE QUESTION: What is 1+1?;1;2;R;99;1;1;70
What is the distance to the moon?  ;Less than 300.000km;More than 300.000km;R;80;20;4;14
What percentage of the human body is water?;70%;80%;L;90;10;4;9
What are the first 6 letters of a French and Belgian keyboard?;Qzerty;Azerty;R;30;70;4;9
Since which year was Queen Beatrix the Queen of the Netherlands?;1975;1980;R;20;80;4;9
What is the name of Superman's girlfriend;Lois Lane;Mary Jane;L;80;20;4;10
Since which year dates the current Dutch Constitution?;1983;1976;L;20;80;4;9
What is meant by proscopy;Predicting the future;Preventive keyhole surgery;L;23;77;4;9
By what name is spiritual leader Siddharta Gautama better known?;Ghandi;Buddha;R;33;67;4;9
```

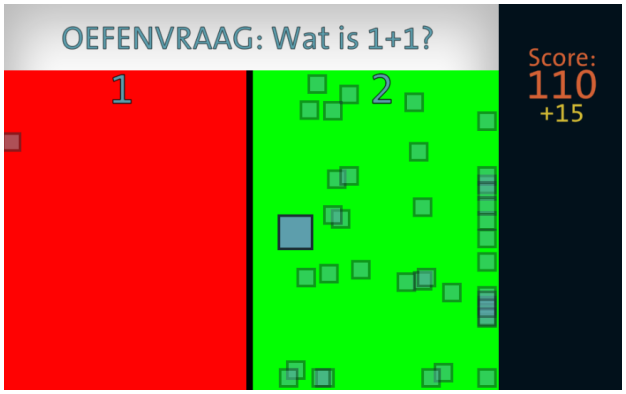


(a) The principle of herd node.

(b) The start position of the principle of herd game.



(c) During the principle of herd game, the smaller cubes will move to either side.



(d) The resolution screen of a question of the principle of herd.

(e) The first resolution screen of the principle of herd.

Figure 4.13: The principle of herd node and the corresponding screens of the different stages.

4.7 Technical Nodes

The technical nodes do not follow the pattern of the normal nodes of showing some content with answers. Instead, they show a single static screen or show no content at all. The technical nodes are blue in the scenario editor.

4.7.1 End Node

This node is always at the end of a story. There are two possibilities: the next story is loaded, or the game is ended. When a story is to be ended, the value `[END]` should be filled in. If a new story should be loaded, the name of that story should be filled in. Another possibility is to fill in the value `[NEXT]`. When this value is filled in, the story with the name of the value of the `NEXT` exposed variable is loaded. This is especially handy when there are multiple end nodes in a story. The end node can be found in figure 4.14.

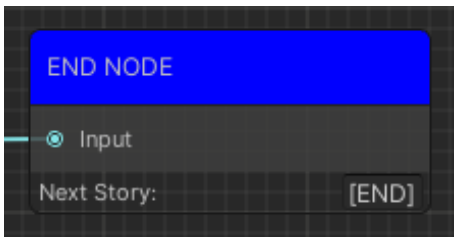
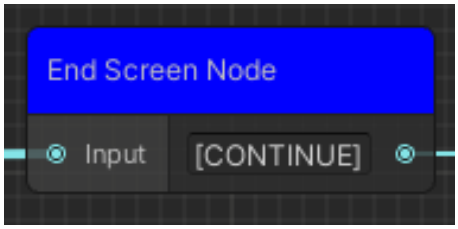


Figure 4.14: The end node.

4.7.2 End Screen Node

This node can be shown at the end of the game. It shows the player's score and presents a link to The Fake News Game, which it cites as an inspiration source. The end screen node and the corresponding screen can be found in figure 4.15.



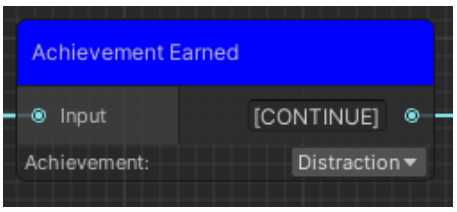
(a) The end screen node.

(b) The screen of the end screen node.

Figure 4.15: The end screen node and the corresponding screen of an end screen node.

4.7.3 Achievement Earned Node

The achievement earned is used every time the player earns an achievement. It is possible to select which achievement the player earns (see figure 4.16a). The game holds the record of which achievements are already earned. The first screen of the achievement earned shows which achievement the player earns and shows some explanatory text (see figure 4.16b). These explanatory texts are translated texts from the works of Stajano and Wilson [73, 74]. The second screen shows which achievements the player has already earned (see figure 4.16c).



(a) The achievement earned node.

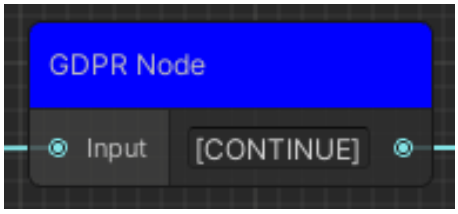
(b) The first screen of the achievement earned node.

(c) The second screen of the achievement earned node.

Figure 4.16: The achievement earned node and the corresponding screens of the different stages.

4.7.4 GDPR Node

The GDPR node can be used at the beginning to make the user accept the General Data Protection Regulation (GDPR) statement. The screen contains a link to the privacy statement. The player can not continue until he accepts this privacy statement. The GDPR node and the corresponding screen can be found in figure 4.17.



(a) The GDPR node.

(b) The screen of the GDPR node.

Figure 4.17: The GDPR node and the corresponding screen of a GDPR node.

4.7.5 Property Changer Node

The property changer node does not show anything on the screen but changes the value of an exposed property. This can be used to customize the user experience of the player. For example, the user could choose between two persons. With the property changer, it is possible to set the exposed variable to the value of that person's name and use it in the following conversations. The property changer node can be found in figure 4.18.

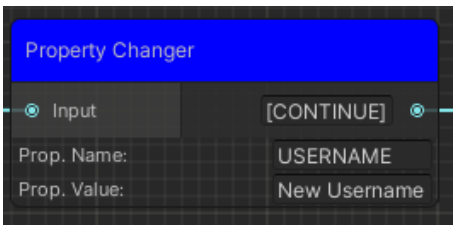


Figure 4.18: The property changer node.

4.7.6 Score Changer Node

The score changer node does not show something on the screen but increases or decreases its score. This node can be placed after a correct or an incorrect answer to change the score accordingly. The score changer node can be found in figure 4.19.

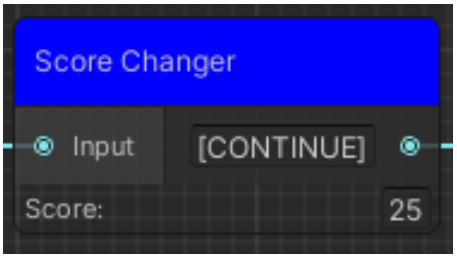


Figure 4.19: The score changer node.

4.7.7 Credibility Changer Node

The credibility changer node is working similar to the score changer node. However, this node alters the credibility meter instead of the score. It also has an extra variable, which can be used to hide or show the credibility meter. This is used in the SES to only show the credibility meter in the second part of the game. The credibility changer node can be found in figure 4.20.

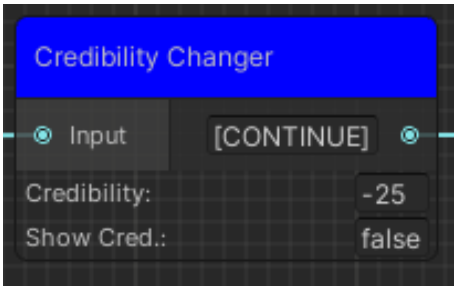


Figure 4.20: The credibility changer node.

4.7.8 Conditional Node

The conditional node does not show something on the screen but can alter the game flow based on an exposed property or the score. This can be used to show specific prompts for specific choices the player made or on the player's level of score at that point in the game. This works by filling in the property name in the node and the specific values in the answers. If the property has a value that is not present in the answers, it will continue with the answer with the `[CONTINUE]` value. To use the score as a conditional value, it must fill in `[SCORE]` as the property name. The answers, the score value and conditional operators ($<$, $=$, $>$) can be filled in. The conditional node can be found in figure 4.21.

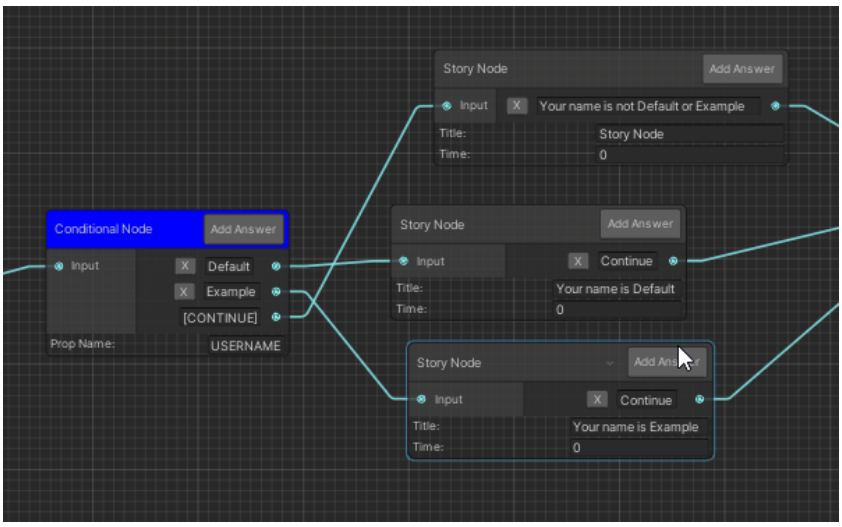


Figure 4.21: The conditional node.

4.8 Technical Design

4.8.1 Unity

The game is a web-based application, which runs on most desktop web browsers. It is developed in Unity, a game engine that can be used to create interactive media [30]. Unity offers support to build to various platforms, including web, Windows, Mac, Android, and iOS. Unity works with a visual Integrated Development Environment (IDE), where elements used in a game are called game objects. All game objects are put together in a scene, which can be seen and adjusted visually in the Unity IDE. Game objects can be a wide array of things, such as text objects or images, but can also be empty. Their behaviour is altered via components. It is also possible to add custom scripts as components. These scripts are written in C#.

4.8.2 Scene

The scene is a visual representation of everything visible on the screen of the game, and this called the User Interface (UI). A UML diagram of the scene used in the SES can be found in figure A.1 in appendix A. Everything is contained in the Main Scene. The main scene contains a few invisible game objects, which hold essential components. The Event System handles inputs given by the user. The Audio Manager handles all audio produced by the application. The Managers object holds all other custom managers, like the Game Manager, UI Manager, and Behavior Log Manager. The functionality of these managers will be further explained in section 4.8.3. The Main Content object holds all UI elements used in the game. This object holds the background, which contains an image and a vignette, and the timer. The achievement overview and pop-up are, most of the time, invisible. The Main Content also holds two containers; the Sidebar and the Game Content. The Sidebar contains the score text and other UI elements that are static and should always be visible. The Game Content is divided into two areas; an area for answers to be displayed and an area where the main game elements can be shown. The different types of content are explained further in section 4.4.

In Unity, it is also possible to save game objects as prefabs and instantiate them in the scene during runtime. These prefabs are the various pieces of UI that form the game. For example, when a standard question prompt needs to be shown, a text prefab is loaded in the content area, and different answers prefabs are loaded for each answer in the answer area. When an entire email needs to be shown, the email prefab is loaded, and the content inside the email is altered depending on the content loaded from the story.

4.8.3 Scripts

The custom made scripts are the backbone of the game. They completely decide the behaviour of the game. There are two types of scripts in the SES; editor scripts and behaviour scripts. Editor scripts are script that runs in the background of the IDE and has no direct influence on the game behaviour. Via these scripts, Unity allows developers to make their own editor windows in the IDE. These scripts are used to make the scenario editor. The second type of scripts is behaviour scripts. These scripts are called during runtime and directly influence the game. These scripts can also be added as components to game objects. First, the editor scripts shall be elaborated more. A simplified UML diagram of the editor scripts can be find at figure B.1 in appendix

B. After the editor scripts, the game behaviour scripts shall be explained. A simplified UML diagram of the game behavior scripts can be found in figure C.1 in the appendix C.

SESStoryEditor

This class inherits from the Unity class `EditorWindow`, which can be inherited to create custom editor windows. This class is the actual window seen in the IDE and holds the story graph (the flow chart itself), the toolbar, the minimap, and the blackboard. The story graph is the class `SESStoryGraphView` and shall be elaborated in section 4.8.3. The toolbar is used to clear the graph and load and save data. This is done by using the static class `GraphSaveUtility`, which is elaborated in section 4.8.3. The minimap is a small build-in feature that helps the user navigate the graph view. The blackboard holds all exposed variables, which is populated by the graph view.

SESStoryGraphView

This class is the flowchart itself and can be seen as one of the editor scripts' most important classes. It inherits the experimental `GraphView` Unity class, which can be used to make graphs. It holds a list of exposed properties, which will be further explained in section 4.8.3. It also creates a search window on start, which will be elaborated in section 4.8.3. The class has a reference to the blackboard, which it populates and starts, and gives the possibility to add extra properties. It is responsible for creating and adding nodes. When a node is created, first, a base node is made, next depending on the node type, extra features are added to the node, such as the possibility to add answers or a title text field. To create a node, only the position and the node type has to be given; the class will handle the rest.

SESNodeSearchWindow

`SESNodeSearchWindow` is a small helper function that opens a search window when the right mouse button is pressed in the scenario editor. In the search window, it is possible to select the node that has to be created. These are sorted into different levels. It has a reference to the `SESStoryGraphView`, which it calls when a node needs to be created. The search window gives the position of the right-click and the selected node type to the graph view.

Node Classes

The node classes hold all information of the node. All nodes inherit the `SESBaseNode`, which inherits the Unity class `Node`. There are two types of nodes; the story nodes and the principle nodes. The principle nodes inherit `PrincipleBase`, which also inherits `SESBaseNode`. The base node holds information all nodes should have, such as a bool, if the node should have a title, what that title should be, and a bool indicating if it has to be connected. Because of the way Unity serializes its data, these nodes can not be serialized directly and need to be serialized into `SESNodeData` classes; this is done by the `GraphSaveUtility`, which will be elaborated in section 4.8.3. Each node has a section where the node can be linked to another node; these are where the answers for the prompts are created. These are called ports. These ports can also have names. Each node also has a unique, Globally Unique Identifier (GUID). Thus, the GUID of a node, together with a port name and a targeted GUID of a node, form the links between nodes, and these are saved as `SESNodeLinkData` classes.

SESExposedProperty

Exposed properties are variables that can be used and altered in the game. They are shown in the scenario editor on the blackboard. They make text in the game less static. An exposed property has a name and a value. For example, when the variable' username' is added, and the input node alters this variable, the next node can have '[username]' in it. That variable will be changed with the filled-in value of the input node.

GraphSaveUtility and SESStoryContainer

Unity has its own serialization system, which can be very useful. It automatically serializes a lot of data. However, it also has a few downsides. For example, it is not possible to serialize abstract classes. It is also hard to serialize objects in a way that items can be used during runtime. A solution for this is to serialize objects to `ScriptableObect` classes. These are custom scripts, which can be seen as objects, that can be accessed during runtime, which are not attached to game objects in the scene. However, because the nodes inherit the `Node` class, which is necessary for the graph to work, they can not be serialized automatically. This is where the `GraphSaveUtility` comes in. This class can save nodes to and load nodes from a `SESStoryContainer`. The `GraphSaveUtility` can be statically accessed, but when accessing, it has to be given the `SESStoryGraphView`.

A SESStoryContainer is a scriptable object, which means it can be serialized by Unity and accessed during runtime. The container holds the following things: A list of SESNodeData, a list of ExposedProperty, a list of SESNodeLinkData and the GUID of the start node. Because Unity can not serialize abstract or inherited classes, all types of nodes are saved as a SESNodeData class.

When saving a graph, it first creates a SESStoryContainer. Next, it creates a SESNodeData for each node and populates that class, and adds all those objects as a list to the container. For each answer port in a node, the base GUID together with targeted node GUID and the port name is saved as SESNodeLinkData classes. It also saves the GUID of the first node. After that, it adds all the exposed properties in a list to the container. Finally, the container is saved as a Unity asset, a generic file type for files loadable in Unity.

When loading a graph in the scenario editor, the GraphSaveUtility first clears the graph and loads the story container. Next, it lets the graph view create nodes for each node data in the container. After that, the nodes are connected by looking at the node links. Finally, it adds the exposed properties to the blackboard.

GameManager

The GameManager is during runtime, the class that controls the flow and behaviour of the entire game. Recall from section 4.8.2 that there is an empty game object in the scene which holds all the manager scripts. To make it possible to add a script to a game object, the script has to inherit the Unity class MonoBehaviour. Every MonoBehaviour has four different methods that are called at specific times. When the game starts, the 'Start' method is called, and directly after that, the 'OnEnable' method is called. When the game is paused, for example, by minimizing the game, the 'OnDisable' method is called. When the game resumes again, the 'OnEnable' method is called again. Every tick, and thus each frame, of the game, the 'Update' method, is called.

In the manager object in the scene, selecting the game manager's start story is possible. On start, the game manager will say to the EventManager to load the SESStoryContainer asset file, which was created by the GraphSaveUtility. The EventManager is a scriptable object, which will be further explained in section 4.8.3. When a new story is loaded, or the event manager calls the answer inputted event, the game manager loads the next node. If this is a principle node, the game manager gives full control to that principle until the 'principle done' event is called. If it is not a principle, it will check if it is an end node and either load the next story or end the game. Else, it calls the 'update UI' and 'question started' event. When a node has a timer, the game manager will also control the timer and move to the answer if the time is up.

EventManager

The EventManager is a scriptable object and thus not attached to the 'managers' object in the scene. It can be seen as a separate object that runs in the background of the game. This means this object can also be 'given' to the other managers, so all managers work with the same event system. The EventManager holds the current story the game is on and which node the story is currently on. Furthermore, it has delegate callback methods that function as an even system that other classes can subscribe to. For example, if the UIManager get the input of a selected answer, it calls the 'InputAnwer' callback, which the GameManager is subscribed to, so it can handle what the game should do next.

UIManager

The UI manager is responsible for all UI on the screen. It is subscribed to the 'UpdateUI' callback called by the GameManager. It looks at the current node provided by the EventManger and updates the UI accordingly, and fills the UI's content from that node. For each answer it adds on the screen, it adds a listener for the answer. If the answer is clicked, the 'InputAnswer' event is called. The GameManger is subscribed to the 'InputAnswer' callback, so it updates the game when the answer is clicked.

Prefab Scripts

The prefab scripts are straightforward scripts that are added to the prefabs in Unity. In this way, they can be created programmatically. They all inherit from the base PrefabData script.

Principle Scripts

The principle scripts have a bit more bit body than the prefab scripts. They can be seen as small games within the game, which is why they can get full control from the GameManger. They control their own game flow and

their own UI, so they do not give acrshortui callback to the `UIManager`. All principle script inherit from the `BasePrinciple` class. When the principle is done, they give control back to the game manager. They do this by calling the 'PrincipleDone' callback. If this callback is called, they give `PrincipleData` with it, which are the results from the principle, such as the number of correct answers. This is used by the `BehaviorLogManger` to log the results correctly.

BehaviorLogManager

This class is responsible for logging the behaviour of the server and writing the results to the server. Because Unity is a multi-platform engine, and the game runs on a web server, it is not possible for the game to directly write to a file. For this reason, the game writes its data via PHP. This will be further explained in section 4.8.4. This class is subscribed to the answer and principle done callbacks. When an answer is submitted, this class writes all interesting information to the server. It does this by looking at the `SESNodeLinkData`, which was submitted by the answer done callback, and the `PrincipleData`, which was submitted by the principle. It also times the time between answers and writes that to the server.

4.8.4 Data Model

As said before, Unity can not directly write to a file itself because the game can build on multiple platforms. To solve this, the game writes to the server via a simple PHP script. At the beginning of the game, the `BehaviorLogManager` makes a CSV file with the current date, precise till the millisecond, as a name. Every time an answer is submitted, this CSV file is updated, so even if a user does not finish the game, the data is still saved. After the testing phase, all CSV files can be loaded in a Jupyter Notebook and perform an analysis on it. Because of how the log manager works, it is effortless to add extra parameters that should be logged.

Chapter 5: Method

5.1 Experiment Design

An experiment is used to test the efficacy of using active gamified inculcation in the form of the SES against social engineering risks. The experiment is a quasi-experimental design, as described by Kirk [44]. A quasi-experimental design means the participants are not randomly assigned. The experiment is a three-group pretest-posttest design. Where one group (group 1) plays the SES, and two groups (group 2 & 3) are the control groups. Their resilience against social engineering was tested before playing the game in the pretest and again after playing the game in the posttest. The participants in the control groups do not know they are participating in an experiment, so the participants can not be randomized. If the participants would be randomized, and participants from group 1 would talk to participants of group 2 and 3, the integrity of the experiment could no longer be guaranteed. An overview of the experiment can be found in table 5.1.

	Group 1	Group 2	Group 3
	Experiment	Control	Control
Pretest	Yes	Yes	Yes
Plays the game	Yes	No	No
Posttest	Yes	Yes	Yes

Table 5.1: An overview of the design of the experiment.

5.1.1 Participants

All participants are employees of the Dutch Armed Forces. The three groups in the experiment are three separate units. The units are comparable to each other and do the same work. Group 1 and group 2 have around 40 employees each, and group 3 has around 90 employees. The different units work in separate locations and do not speak or cross each other. Because the units work at different locations, there is no chance of cross-contamination.

5.1.2 Pretest

The pretest uses the scenario of Alberto Stegeman in his show Undercover in Nederland (section 3.3.1-3.3.3). An outsider gains access to a military base, where he can walk around the living and diner quarters but has no access to the more high-security areas. While the outsider is there, he is able to perform the "QR codes" scenario (section 3.2.9). In this experiment, this was simulated by secretly hanging posters on the locations of the three groups. For privacy and security reasons, no personal information was used in the experiment. This is why on the site, the QR code refers to the user is asked to fill in a unique code, which can be used to link the persons from the pretest with the posttest. The techniques used in the pretest are *need and greed* and *distraction*. The poster offered the participants small benefits that could be used within the organization. The posters have hung for five days, and after the period, they were removed. On locations of the Dutch Armed Forces, posters are often used for intern communication, so participants are familiar with this communication method. However, the content of the poster only contains information an outsider might also know.

5.1.3 Playing The Game

Five days after the pretest, the experiment group received an invitation to play the game. They were given a week to play the game. They were asked to take 60 minutes for the game and play it until the end. In the

results, it is also interesting to look at the behaviour in the game. One thing to look at is the answers for the examples at the beginning and end of the game. Better answers at the end of the game might indicate a positive effect of the game. It is also possible to compare that effect between the two versions of the game.

5.1.4 Posttest

The posttest was held four days after the last player played the game. The idea of the posttest is the same as the prettest, although different posters were used. The posters focused on the same techniques: *need and greed* and *distraction*. Because some users filled in a unique code, it was possible to identify if the same users scanned the posters in the pretest as the posttest. The posters hung again for five days and were removed after.

5.1.5 Debrief

After the posters were removed during the posttest, the participants were debriefed via email. In this email, the idea of the experiment was explained. Also, the posters used in the experiment were shown. If the participants had any questions, they could ask the researchers.

Chapter 6: Results

6.1 Results From The Experiment

6.1.1 Pretest

The results of the pretest can be found in table 6.1. This experiment tested the willingness of the participants to scan and click a malicious link from a QR code. To see whether the groups in this experiment behaved the same before the experiment, the following null hypothesis should not be rejected:

> **Null hypothesis:** There is no statistical difference between the different groups' behaviour during the pretest.

The hypothesis will be tested using a Chi-square test. The results of the Chi-square test ($\chi^2 = 2.48$, $df = 2$, $p = .29$) showed no statistical significant difference between the groups, therefore the null hypothesis can not be rejected. There is no statistically significant difference in the behaviour of the groups before the experiment.

	Clicks during pretest		
	Clicked	Did not click	Total
Group 1	9 (22.5%)	31 (77.5%)	40 (100%)
Group 2 (control 1)	4 (10%)	36 (90%)	40 (100%)
Group 3 (control 2)	16 (20%)	64 (80%)	80 (100%)
Total:	29 (18,13%)	131 (81,88%)	160 (100%)

Table 6.1: Number of clicks during the pretest per group.

6.1.2 Posttest

The result of the posttest can be found in table 6.2. To see an effect of the intervention, there should be a statistically significant difference in the group's behaviour during the posttest. The following null hypothesis should be rejected:

> **Null hypothesis:** There is no statistical difference between the behaviour between the different groups during the posttest.

The hypothesis will again be tested using a Chi-square test. The results of the Chi-square tests ($\chi^2 = 0.95$, $df = 2$, $p = .62$) show no statistical significant difference, therefore the null hypothesis can not be rejected. There is no statistically significant difference in the behaviour of the groups after the intervention. This means there is no evidence that the intervention positively affected the behaviour of the participants. However, it is important to note that not all participants in the experiment group could play the game, which might have flawed the experiment.

	Clicks during posttest		
	Clicked	Did not click	Total
Group 1	10 (25%)	30 (75%)	40 (100%)
Group 2 (control 1)	7 (17,5%)	33 (82,5%)	40 (100%)
Group 3 (control 2)	20 (25%)	60 (75%)	80 (100%)
Total:	37 (23,13%)	123 (76,88%)	160 (100%)

Table 6.2: Number of clicks during the posttest per group.

6.2 Results From The Game

6.2.1 General Results

Twenty-two participants played the game. When looking at the game's general results, the game was not perceived as a fun game, with an average of 5.32 out of 10. The game was deemed to be educational, with an average of 5.95 out of 10. The participants, however, do think they are safe from future social engineering attacks, giving themselves an average of 3.57 out of 10 chance of getting caught in a social engineering attack. The general results can be found in table 6.3. Because the group of participants largely consisted of males, the gender of the participants was not asked to ensure the privacy of the participants. Therefore, gender shall not be taken into account for this experiment.

Variable	Average Answer
Age	26.19
Educational value	5.95/10
Fun value	5.32/10
Chance of falling for a social engineering attack after playing the game.	3.57/10

Table 6.3: The general results from the Social Engineering Simulator (N=22).

6.2.2 Example Results

The following section is about the results from part 1 and part 3 of the game. This is the part where the player was presented seven examples of possible social engineering attacks and had to choose whether these are safe, suspicious or dangerous. For clarity, the seven examples presented in part 1 of the game will be called the first round of examples. The seven examples presented in part 3 of the game will be called the second round of examples.

The answers given per example can be found in appendix D. Not all players received the same prompts, so the number of participants can vary per example. For instance, if a player deemed a conversation to be 'dangerous', he would not receive the next prompt, while a player who answered 'safe' will receive the next prompt of that example. The most interesting aspect to look at is the performance of the first round of examples versus the second round. The examples of the first round and the second round were designed to be of the same difficulty and focus on the same techniques. Better performance in the second round might indicate a positive effect of gamified active inoculation on the resilience against social engineering attacks.

To look at this effect, it is required to only look at the final answer of an example. For instance, if an example is a phone conversation and has four prompts, the participants might give three correct answers at the beginning of the conversation, but give an incorrect answer at the last prompt. If this happens, the participant has made three correct answers for this particular example, while he only made one incorrect answer for this example. Still, this should be considered as an incorrect answer. This is prevented by only looking at the final answer of an example. This also makes sense; the final prompt of an example is when a participant makes the final decision in a conversation and makes either the right or the wrong choice.

To compare the effect, the number of incorrect answers are juxtaposed. The incorrect answers are chosen instead of the correct answers because they are the most important variable. Every wrong answer represents a potentially successful social engineering attack and should be as low as possible. The average percentage of wrong answers in the first round, compared to the percentage of wrong answers in the second round for both legit and malicious examples is presented in figure 6.1. To test the statistical significance, dependent T-tests are used. The answers given in the first round ($M = 4.32$, $SD = 1.2$) compared to the answers given in the second round ($M = 2.55$, $SD = 1.4$) show indeed a significant effect $t(21) = 4.9, p < .001$. On average, the participants performed 25.11% better in the second round than in the first round. Some participants performed worse in the second round than in the first round. No clear reason was found for this behaviour.

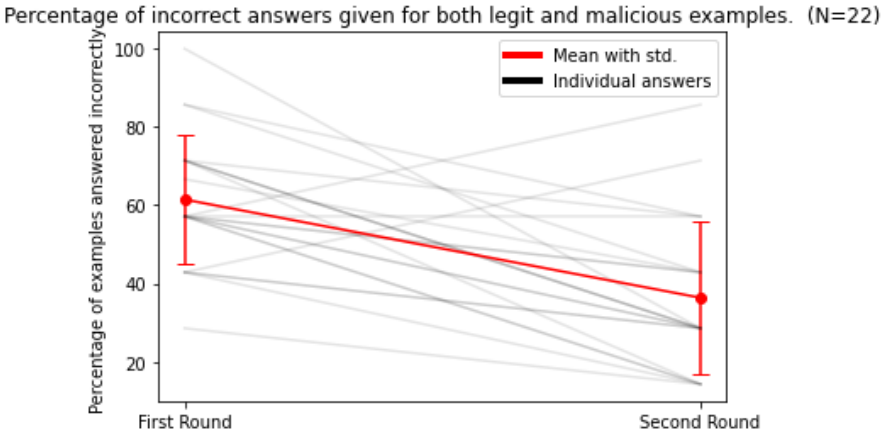


Figure 6.1: The percentage of wrong answers given in the first round of examples (part 1), compared to the second round of examples (part 3), for both legit and malicious examples of the game.

A positive effect of the game can be seen for all examples. However, it is also interesting to see the performance of only the malicious examples and only legit examples. The performance of the malicious examples is a benchmark of how participants perform with actual attacks, their level of resilience against it and how this changed during the game. The performance of the legit example is a benchmark of the level of scepticism and how this changed.

The percentage of wrong answers in the first round, compared to the percentage of wrong answers in the second round for only the malicious examples can be found in figure 6.2. There were four malicious examples in the first round and five in the second round, so the percentage of wrong answers will be compared. Comparing the percentage of wrong answers of the first round ($M = 68.26$, $SD = 20.3$) to the percentage of wrong answers of the second round ($M = 41.82$, $SD = 23.8$), shows indeed a significant effect $t(21) = 4.0, p = .001$. On average, participants performed 26.44% better in the second round when looking only at the malicious examples.

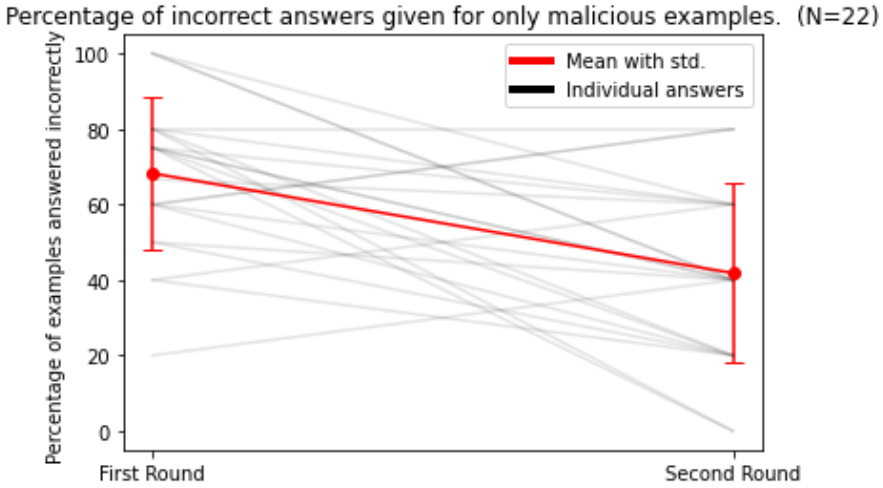


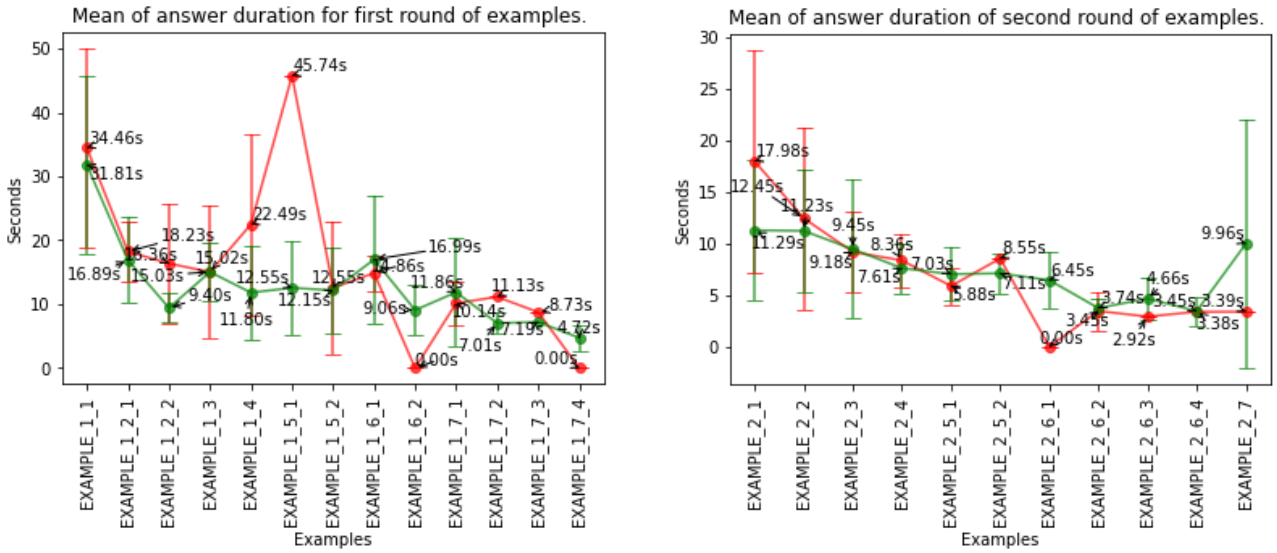
Figure 6.2: The percentage of wrong answers given in the first round of examples (part 1), compared to the second round of examples (part 3), for only malicious examples of the game.

The first round of examples contained three legit examples, and the second round contained two legit examples. When comparing the percentage of wrong answers in the first round ($M = 49.24$, $SD = 31.1$), with the percentage of wrong answers in the second round ($M = 40.91$, $SD = 33.2$), no significant effect can be seen $t(21) = 0.9, p = .392$. Thus, the participants did not become significantly more sceptical towards legit examples.

6.2.3 Duration Results

Another metric that might say something about the performance of a participant is the amount of time someone takes for each of the examples presented in part 1 and part 3. When looking at the duration per example, all prompts of an example should be taken into account because all show the participant's behaviour. Something to look for is the difference in duration between a correct and a wrong answer. This might help say something

about the behaviour of the participants. The average duration per prompt in the examples can be found in figure 6.3. Per example, the average duration of participants who gave a correct answer is compared to the average duration of participants who gave an incorrect answer. An independent T-test tests the statistical significance between the duration of the correct answers and the duration of the wrong answers. Comparing the duration of the correct answers in the first round ($M = 13.93$, $SD = 9.5$), with the duration of the wrong answers ($M = 20.47$, $SD = 13.2$), shows a significant effect $t(143) = -4.0, p < .001$. Comparing the duration of the correct answers of the second round ($M = 7.44$, $SD = 6.1$), with the wrong answers of the second round ($M = 11.02$, $SD = 8.8$), also shows a significant effect $t(163) = -3.2, p = .002$.



(a) The average duration per example in the first round.

(b) The average duration per example in the second round.

Figure 6.3: The average duration per example in the first and second round.

This is an interesting effect. An explanation for this might be that participants were unsure which answer to pick and eventually picked the wrong answer. At the same time, people who knew the answer made their decision quicker. This might also indicate that taking a longer time to look at suspicious communication, which is often recommended, might not always be as effective and might even be counterproductive. This is something that might be researched further in the future.

The difference between the duration of answers between the first and the second round might also have a significant effect. However, the most logical cause for this is that the participant better mastered the game's controls. This is why this effect is not further looked into.

6.2.4 Technique Section Results

Need and Greed Technique.

From the *need and greed* technique section, it is possible to look at which information participants were willing to give away. A histogram of the persons willing to give away information can be found in figure 6.4. As shown in the figure, the amount of participants willing to give away information for money is relatively high. Six people were even willing to give away their work email and password. Because of ethical and privacy reasons, the information itself was not collected. So, it is not possible to check whether the participants filled in the information truthfully.

Example six of the second round of examples (figure D.4a - D.4d) is a similar situation, where a student starts by asking for simple information and gradually asks for more sensitive information. When looking at the results of that example, only two participants answered 'safe' on the first sensitive question, and only one participant answered 'safe' on the last sensitive question. This indicates that the participants learned from this section and are better aware of this method of stealing information.

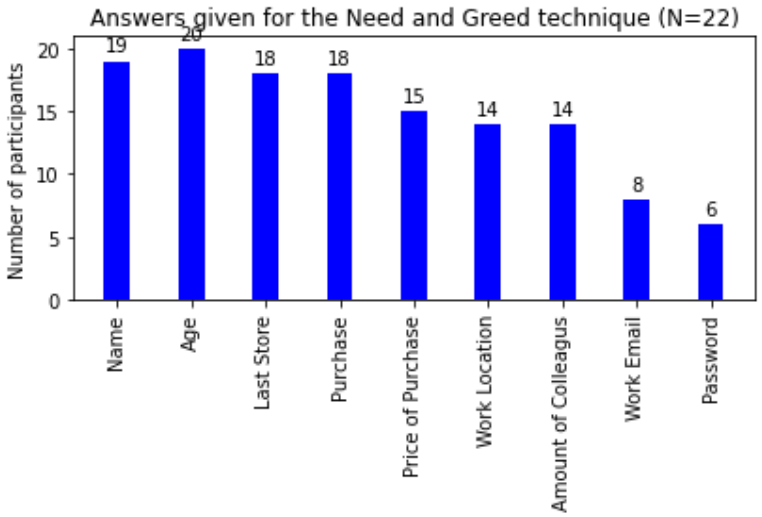


Figure 6.4: The amount of participants that were willing to give away certain information during the *need and greed* section.

Herd Technique

During the *herd* technique section, the players had to choose between two answers and were mislead by what they thought were other players. A histogram of the given answers by the participants can be found in figure 6.5. In this figure, the red labels represent questions where most other players went to the wrong answer; green labels represent questions where the majority went to the correct answer. An apparent effect of the *herd* technique can be seen after question 6. The sixth question reinforced the trust in the other players. In the seventh question, 19 participants followed the other players. This gradually becomes lower in the following questions. The direct effect of this technique is complex to see in the examples, because the *herd* technique is hard to simulate in those examples.

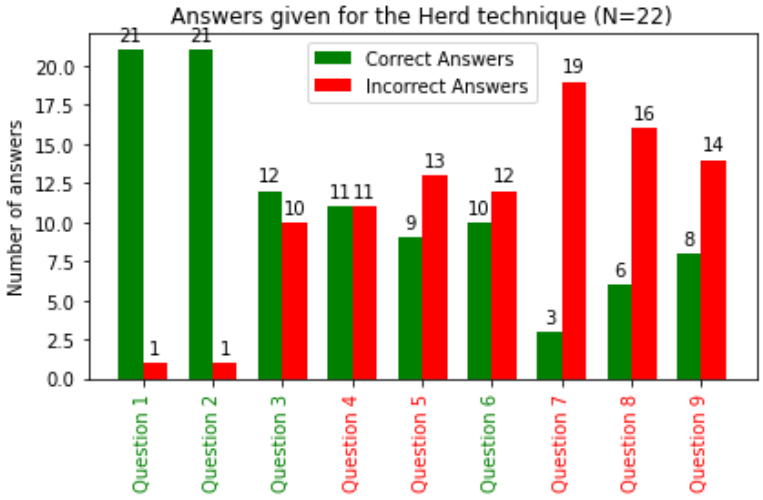
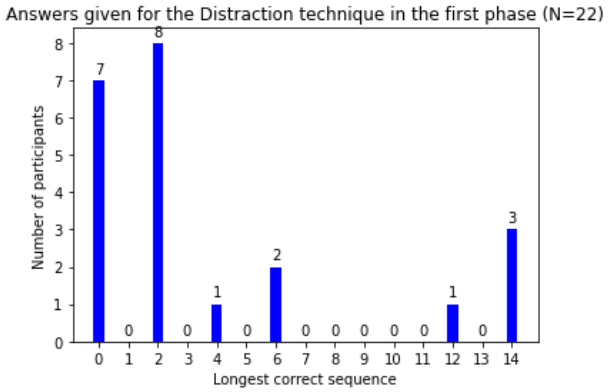


Figure 6.5: The answers given during the *herd* section.

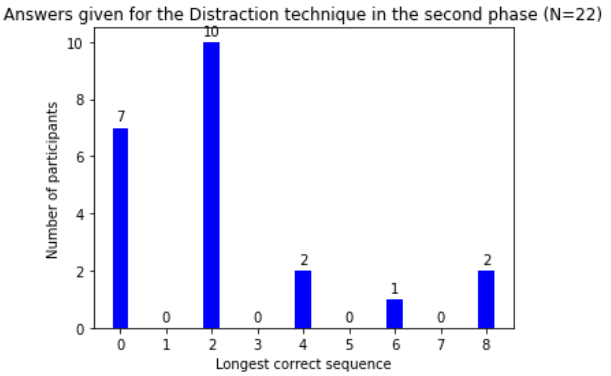
Distraction Technique

During the *distraction* technique section, the participants had to remember a sequence of numbers in the first phase. In the second phase, they had to do the same but were distracted by having to do a math sum after a few numbers. A histogram of the longest correct sequence given by the participants in the first and second phase can be found in figure 6.6. Comparing the longest sequence per participant of the first phase ($M = 3.91$, $SD = 4.7$), with the longest sequence per participant of the second phase ($M = 2.27$, $SD = 2.4$), indicates a significant effect $t(21) = 2.11, p = .047$. This indicates that the participants indeed performed less in the second phase of the game than the first. Looking at the histogram, a fair amount of players had in both the first and second round a longest sequence of zero. This might indicate they either did not understand the game or found

it too hard. The effect of this section is hard to pinpoint because the *distraction* technique is always combined with other techniques and is hard to simulate.



(a) The longest correct sequence given in the first phase of the *distraction* technique section.

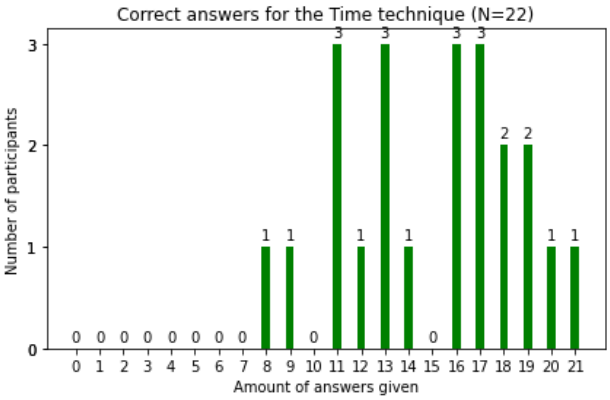


(b) The longest correct sequence given in the second phase of the *distraction* technique section.

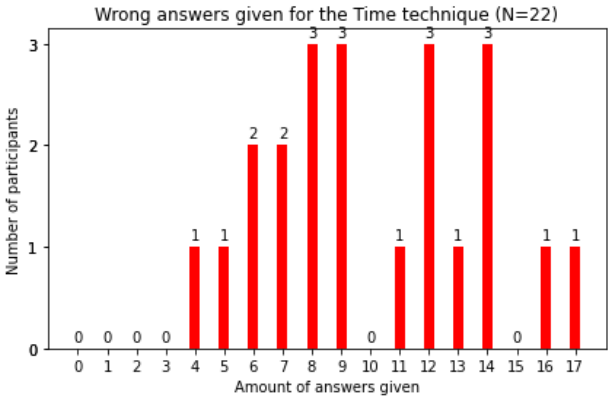
Figure 6.6: The longest correct sequence given in the first and second phase of the *distraction* technique section.

Time Technique

During the *time* technique section, the participants had to look at tiny mistakes at passports while under time pressure. A histogram of the wrong and correct answers can be found in figure 6.7. The average of correctly answered answers is $M = 14.95$.



(a) Histogram of the amount correct answers given for the *time* technique section.



(b) Histogram of the amount wrong answers given for the *time* technique section.

Figure 6.7: Histogram of the amount correct and wrong answers given for the *time* technique section.

6.2.5 Quantifying Cyber Awareness

In an ideal situation, an awareness intervention can be used to train personnel and indicate the personnel's awareness level. In the literature, there is no clear cut way of quantifying cyber awareness. This section will explore a method of quantifying the level of cyber awareness from the results of the SES. This could be a setup for future research into the topic of quantifying cyber awareness.

To get an objective measurement of the level of cyber awareness, the level is mapped to an integer from zero to a hundred. This way, the levels of all participants can be compared, and an objective result can be given to the participant. The ultimate goal would be a tool where a company can let its entire personnel play the game and get a value for each employee. This way, the company knows which staff needs more training. The cyber awareness level is calculated as follows:

$$Awareness = \lfloor EV * 0.8 + NaG * 0.1 + GS * 0.05 + A * 0.05 \rceil \tag{6.1}$$

where:

EV = The value per malicious example.
NaG = The amount of given answers in the *need and greed* section.
GS = The game score in part 2 of the game.
A = The answers given to information questions.

Example values (EV)

All values are mapped on a scale from zero to hundred, so the total value always has a maximum of a hundred. The value per example determines for the largest part the level of cyber awareness. This value is inspired by cyber risk assessments, where the risk is determined as:

$$Risk = probability * impact \tag{6.2}$$

To mimic this, each example in the SES was given a probability value and an impact on a scale of one to five, where one is the lowest, and five is the largest. The probability value determines how big the chance is that personnel will face such an example in real life. The impact value determines how big the impact for the company would be if the participant would fall for such an example. This way, the company can also give extra weight to examples they find the most important. The example value would then be calculated as:

$$EV = \frac{\text{Total example value} - \text{Participants example value}}{\text{Total example value}} * 100 \tag{6.3}$$

where:

$$\text{Total example value} = \sum_{n=1}^{\text{All examples}} \text{Example's impact value} * \text{Example's probability value} \tag{6.4}$$

$$\text{Participants example value} = \sum_{n=1}^{\text{Examples participant answered incorrect}} \text{Example's impact value} * \text{Example's probability value} \tag{6.5}$$

Need and greed answer values (NaG)

Because a relatively large amount of participants gave away information in the *need and greed* section, and this could have serious consequences, these values are taken into account when calculating the level of cyber awareness. Participants are given points when they did not fill in any information and loose points when filling in information. The participants were asked nine questions. This value is calculated as:

$$NaG = \frac{9 - \text{Amount of need and greed answers filled in}}{9} * 100 \tag{6.6}$$

Game score value (GS)

The game score earned in the second part of the game indicates how well a participant scored in the second part of the game. This indicates how serious someone took the game and an indication of his ability to estimate social engineering risks and attacks. This is why this game score is taken into account for a small part. The total score a player could earn in the second part was twelve hundred. This is calculated as:

$$GS = \frac{\text{End score part 2} - \text{Begin score part 2}}{1200} * 100 \tag{6.7}$$

Information questions value (I)

The last value that is taken into account is based on the answers given to two information questions. The first question was how safe someone ranks their internet behaviour on a scale from one to ten. The second question was how big the participant thinks the chance is they will fall for a social engineering attack after playing the game. These two values can be an indication of how the participant rank their own performances. However, it is hard for someone to indicate their own performances truly, so this value will only be weighted for a small part. The value is calculated as:

$$I = \frac{\text{Answer of internet safety} + (11 - \text{Answer of social engineering chance})}{20} * 100 \tag{6.8}$$

Results

For all participants, their quantified cyber awareness value was calculated. A histogram of the values can be found in figure 6.8. The lowest value was eighteen, and the highest value was seventy-four. Most values lay between forty and sixty, which would indicate an average level of cyber awareness. Future research is needed to see whether these values are a correct representation of the level of awareness. A future addition of the SES could be showing this value to the participants after playing the game.

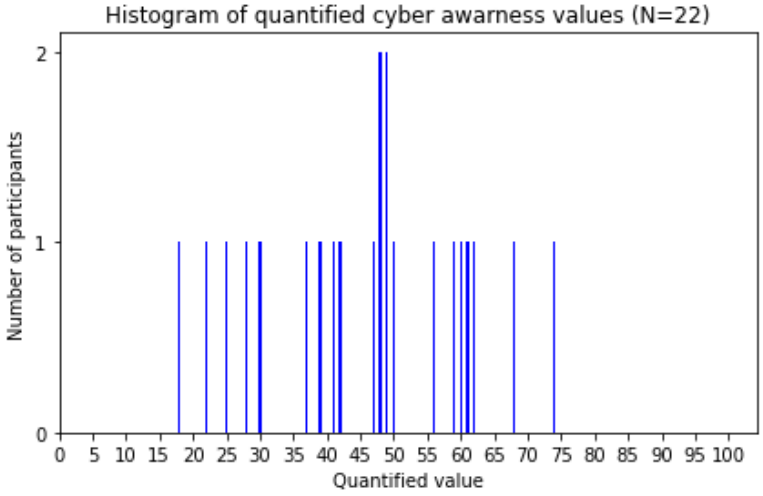


Figure 6.8: Histogram of the quantified cyber awareness values.

Chapter 7: Conclusions

7.1 Discussion

This thesis is one of the first works to use gamified active inoculation as an intervention method against social engineering risks. The intervention places the participant in the shoes of a social engineer and lets him actively engage with various social engineering techniques. This way, the player is 'vaccinated' with different social engineering techniques, and the body learns to protect itself against it by itself.

The effectiveness of the intervention was tested in an experiment with QR codes. No evidence was found that the intervention was effective in this experiment. However, from the data from the intervention itself, evidence was found that active inoculation reduced the chance participants fell for social engineering attacks. Participants did not score significantly better or worse on legit conversations, which shows that participants did not become more sceptical towards legit conversations.

7.1.1 Research Question 1

> *How effective is an intervention in the form of gamified active inoculation against social engineering attacks?*

Answering the first research question, it can be argued that gamified active inoculation has a moderate effect as an intervention against social engineering risks. The effect corresponds with the Fake News Game results and other works where inoculation is used against persuasion [7, 70]. Future research is needed to have a more precise number, where the examples before and after the game should be randomized to have better and more equal research. This work can not exclude some questions in the first round being harder or easier than in the second round.

Although the intervention had a statistically significant effect, it can not be seen as the panacea against social engineering risks. No solution can take away all risks, so this intervention should always be used in combination with other controls. Also, some malicious social engineering attempts were not widely spotted by the participants. For example, looking at the first example of the second round (figure D.3a). The text of this phishing mail used a harsh, compelling tone and contained a link to a malicious site. But the spelling error in the site was hard to spot (`vattenfall.nl` vs `vattenvalll.nl`). The participants learned that criminals liked to put time pressure on their victims, which this email did. However, only 6 participants were able to spot this malicious attempt. This indicates that only focusing on the psychological social engineering techniques itself might not be enough. There should also still be a focus on spotting the technical tricks. Concluding, the intervention had a moderate effect, but more research is needed to affirm this and to state a definitive number.

7.1.2 Research Question 2

> *Can physiological techniques and principles used by social engineers be simulated in a gamified active inoculation environment?*

In the SES, interactive sections were used to simulate a social engineering attack's psychological effects. During the *need and greed* section, many participants gave away information for a small amount of money. A small group of participants even gave away their passwords. Unfortunately, it is impossible to check if this information was correct due to ethical and privacy reasons. However, this shows that participants did believe this section. Later in an example comparable to this section, only one person was willing to give away sensitive information. This shows that this indeed had a positive effect.

For the *time or scarcity*, *herd* and *distraction* sections, it is harder to see the direct effect of the section on the results in the third part of the game. In the examples in part 3, it is hard to simulate those techniques. However, the results of the *herd* show that players were indeed compelled to follow, what they thought were, the other

players. In the *distraction* an apparent effect of the distraction can be seen when comparing the results from the first phase and the second phase. Concluding, it can be said that psychological techniques and principles used by social engineers can indeed be simulated within the game presented in this thesis.

7.1.3 Research Question 3

> *Can social engineering risks be derived, and the level of resistance against social engineering attacks be quantified from a gamified active inoculation environment?*

It became clear that most participants had some basic knowledge of social engineering risks and knew about some attacks. For example, the 'friend in need' Whatsapp scam has been in the news often, and most people spotted that scam in the first round (figure D.1f). However, the harder to spot phishing emails were more challenging. Most participants fell for the difficult to spot phishing mail in the first round (figure D.1e). In the second round, most participants still fell for a more challenging to spot phishing mail in the second round (D.3a).

This work presents a method for quantifying cyber awareness from data collected by the intervention itself. Participants were asked to look at different conversations and asked to spot the malicious ones. The examples of possible social engineering attacks were given a probability value of such an attack happening and an impact value of the impact of such an attack. This approach is based on risk management. Also, in this way, the organization can give weight to risks they find the most important.

Other metrics that gave weight to the cyber awareness levels were results from the game. Most participants scored average on this newly presented scale. More research is needed on whether this cyber awareness level is an accurate presentation of the actual level of cyber awareness. This method could be used as a starting point for other scholars.

Another metric that might say something about the analysis of the risks of the personnel is the time participants needed to answer the questions. Answers filled in correctly were significant answered quicker than answers filled in wrong. This metric was not considered for quantifying cyber awareness because it is not clear whether this effect is also present in a real-life scenario. This suggests that more research is needed into the correlation between the time people take to look at conversations and their ability to spot social engineering attacks. This might help with the development of new solutions. One can think of a tool that gives an extra warning when a user looks more than a specific time at a flagged potentially dangerous email. Concluding, a method of quantifying the level of resistance is presented in this thesis. More research is needed to test how accurate this number is. From the results of the intervention, some specific risk can be derived.

7.1.4 Other Findings

This thesis not only developed an intervention but also tested this intervention within a large organization. For companies conducting interventions against social engineering, risks are crucial and can prevent many losses. However, deploying such an intervention can be more complex than one thinks, especially in large organizations. This can be explained with the COM-B model of Michie et al. [52]. *Capabilities* were present by having both the knowledge to perform an intervention and having the intervention itself. Within the organization, people knew the importance of protecting employees against these risks, so the *motivation* was present. However, the policies for practising and executing these kinds of interventions, combined with an experiment, were not present. Thus the *opportunity* was lacking. Even though the employees' behaviour might be changed by the intervention, behaviour change should also be created within the organization. As long as the opportunity is not there, the desired behavioural change will lag behind. This is something organizations, and companies should take into account.

7.2 Limitations

7.2.1 Experiment Design

The experiment had to be changed throughout this research a few times due to security and privacy concerns. In the original design of the experiment, a phishing mail would be used instead of a poster with QR codes. Also, the user would be asked to fill in its work email address and password, from which the email addresses would be saved. However, the Dutch Armed Forces have stringent security policies. The phishing mail would require employees' email addresses to be stored on an external server, which was not allowed. Also, using email addresses and passwords in an experiment raised both security and privacy concerns. This is why the choice was made for posters and unique codes. This way, no sensitive data had to be stored. Although scanning,

visiting and interacting with a malicious website is potentially less harmful than giving away login credentials, it still could have significant consequences. This is why this experiment is still deemed to be representative of an actual attack.

Not all people in the experiment group could play the game due to time constraints on their side. This means some people were present at the location of the experiment group who not did receive the intervention. This might have caused people to click during the posttest who were counted as the experiment group, but did not have received the intervention. In the original experiment design, group 2 would also play the game and not function as a control group. This group would have played a version of the game without the interactive parts. However, due to the time constraints of this group, they were not able to play the game.

7.2.2 Corona Virus

This research took place during the global corona pandemic. This crisis also affected this research and its experiment. The participants of this research were chosen because they still worked on location and not remote. However, one can imagine that the work environment looked different during the corona crisis than during normal working operations. For example, work locations are visited less often and by fewer people. Strangers on the location are thus less common, which makes them easier to spot. The impact of the corona crisis on the results of the experiment are hard to predict and are thus unknown. During the earlier phases of this research, it was also suggested to experiment with a physical confrontation. However, this turned out to be difficult due to corona restrictions.

7.2.3 Time Constraints

Partly by the corona crisis, some time constraints were laid on the experiment. In an ideal scenario, the experiment existed in multiple pretests and posttest. Also, it would be interesting to look at the decay of resilience over time. This could be measured by doing multiple posttests spanning over a longer time. Inoculation theory also states that the protection against persuasion is improved by doing multiple interventions over time. This was not possible for this thesis but is a recommendation for further research.

7.3 Further Research

The results of this thesis are promising. However, future research is required to explore the effects of active inoculating on reducing social engineering risks. The experiment of this thesis had a relatively small sample size. Future research could look into more extensive and more diverse groups. Also, this research used posters as a vector for the experiment. Future works could look into the effect of inoculation on various social engineering attacks, such as phishing and psychical approaches.

The posttest of this research was executed a week after the intervention. The time decay of the intervention is thus shallow. Future works can look at how the effectiveness of the intervention decays over time. This can be done by doing multiple posttests over a more extended period.

This work used fixed examples before and after the game to test effectiveness within the game. In future works, randomized examples can eliminate the possibility of examples being harder or easier. Future work could also look into the exact effect of the interactive technique sections. All in all, active inoculation as an intervention method is a promising new method and could be helpful for different kinds of interventions.

This work also presents a method for quantifying the level of cyber awareness of the participants. This is a new research area, and other scholars are encouraged to look into this area more. This work could be used as a starting point for further research.

Bibliography

[1] Abagnale, F. W. *The Art of the Steal: How to Protect Yourself and Your Business from Fraud–America's# 1 Crime*. Currency, 2002.

[2] Abawajy, J. User preference of cyber security awareness delivery methods. *Behaviour & Information Technology 33*, 3 (2014), 237–248.

[3] Ahmad, T. Corona virus (covid-19) pandemic and work from home: Challenges of cybercrimes and cybersecurity. *Available at SSRN 3568830* (2020).

[4] Ancher, M., V. d. K. R., and Leukfeldt, R. Studenten treden in voetsporen cybercrimineel om meer inzicht te krijgen in sociaal engineering. *Informatiebeveiliging Magazine 19* (2019), 26–33.

[5] (ANP). Alberto stegeman moet boete betalen voor nepbom op kazerneterrein. *NOS* (2020).

[6] Arthur, C. Virus phone scam being run from call centres in india. *The Guardian* (2010).

[7] Banas, J. A., and Rains, S. A. A meta-analysis of research on inoculation theory. *Communication Monographs 77*, 3 (2010), 281–311.

[8] Barth, S., Hartel, P. H., Junger, M., and Montoya, L. Teaching empirical social-science research to cybersecurity students: The case of\" thinking like a thief\". *IEEE Secur. Priv. 17*, 3 (2019), 8–16.

[9] Basol, M., Roozenbeek, J., and van der Linden, S. Good news about bad news: gamified inoculation boosts confidence and cognitive immunity against fake news. *Journal of cognition 3*, 1 (2020).

[10] Bowen, B. M., Devarajan, R., and Stolfo, S. Measuring the human factor of cyber security. In *2011 IEEE International Conference on Technologies for Homeland Security (HST)* (2011), IEEE, pp. 230–235.

[11] Brewster, T. Lying sextortion scammers score $250,000 after sending victims their own hacked passwords. *Forbes* (2018).

[12] Bullee, J.-W., and Junger, M. How effective are social engineering interventions? a meta-analysis. *Information & Computer Security* (2020).

[13] Bullée, J.-W. H., Montoya, L., Pieters, W., Junger, M., and Hartel, P. H. The persuasion and security awareness experiment: reducing the success of social engineering attacks. *Journal of experimental criminology 11*, 1 (2015), 97–115.

[14] Cialdini, R. B. Influence: The psychology of persuasion.

[15] Compton, J. Inoculation theory. *The Sage handbook of persuasion: Developments in theory and practice 2* (2013), 220–237.

[16] Compton, J., Jackson, B., and Dimmock, J. A. Persuading others to avoid persuasion: Inoculation theory and resistant health attitudes. *Frontiers in psychology 7* (2016), 122.

[17] Cone, B. D., Irvine, C. E., Thompson, M. F., and Nguyen, T. D. A video game for cyber security training and awareness. *computers & security 26*, 1 (2007), 63–72.

[18] Conti, G., Babbitt, T., and Nelson, J. Hacking competitions and their untapped potential for security education. *IEEE Security & Privacy 9*, 3 (2011), 56–59.

[19] Dimkov, T., Van Cleeff, A., Pieters, W., and Hartel, P. Two methodologies for physical penetration testing using social engineering. In *Proceedings of the 26th annual computer security applications conference* (2010), pp. 399–408.

[20] Dolan, P., Hallsworth, M., Halpern, D., King, D., Metcalfe, R., and Vlaev, I. Influencing behaviour: The mindspace way. *Journal of Economic Psychology 33*, 1 (2012), 264–277.

[21] Dolan, P., Hallsworth, M., Halpern, D., King, D., and Vlaev, I. Mindspace: influencing behaviour for public policy.

[22] Eagly, A. H., and Chaiken, S. *The psychology of attitudes.* Harcourt brace Jovanovich college publishers, 1993.

[23] Edgar, T. W., and Manz, D. O. *Research methods for cyber security.* Syngress, 2017.

[24] enisa. Eu cyber cooperation the digital frontline.

[25] enisa. Cybersecurity culture guidelines: Behavioural aspects of cybersecurity.

[26] Fogg, B. J. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology* (2009), pp. 1–7.

[27] Gardner, B., and Thomas, V. *Building an information security awareness program: Defending against social engineering and technical threats.* Elsevier, 2014.

[28] Gragg, D. A multi-level defense against social engineering. *SANS Reading Room 13* (2003).

[29] Granger, S. Social engineering fundamentals, part i: hacker tactics. *Security Focus, December 18* (2001).

[30] Haas, J. K. A history of the unity game engine.

[31] Hadlington, L. The "human factor" in cybersecurity: exploring the accidental insider. In *Psychological and Behavioral Examinations in Cyber Security.* IGI Global, 2018, pp. 46–63.

[32] Herjavec, R. Cybersecurity ceo: Don't let coronavirus fears distract your employees from phishing scams. *Cybercrime Magazine* (2020).

[33] Hof, C. Social engineering. *TNO* (2013).

[34] Irvine, C. E., Thompson, M. F., and Allen, K. Cyberciege: gaming for information assurance. *IEEE Security & Privacy 3*, 3 (2005), 61–64.

[35] Isacenkova, J., Thonnard, O., Costin, A., Francillon, A., and Balzarotti, D. Inside the scam jungle: A closer look at 419 scam email operations. *EURASIP Journal on Information Security 2014*, 1 (2014), 4.

[36] ITU-T. Recommendation x. 1205: Overview of cybersecurity, 2008.

[37] Janczewski, L. J., and Fu, L. Social engineering-based attacks: Model and new zealand perspective. In *Proceedings of the international multiconference on computer science and information technology* (2010), IEEE, pp. 847–853.

[38] Johnson, A. Francophoned – a sophisticated social engineering attack. *Broadcom* (2013).

[39] Junger, M., Montoya, L., and Overink, F.-J. Priming and warnings are not effective to prevent social engineering attacks. *Computers in human behavior 66* (2017), 75–87.

[40] Kamp, R. Corona: phishing en oplichting. *Consumentenbond* (2020).

[41] Kaspersky. aspersky lab identifies operation "red october," an advanced cyber-espionage campaign targeting diplomatic and government institutions worldwide. *Kaspersky* (2013).

[42] Kepinski, W. Rsa geeft nieuwe securid tokens uit. *Computable* (2011).

[43] Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., and Weippl, E. Qr code security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia* (2010), pp. 430–435.

[44] Kirk, R. E. Experimental design. *Handbook of Psychology, Second Edition 2* (2012).

[45] Krombholz, K., Hobel, H., Huber, M., and Weippl, E. Advanced social engineering attacks. *Journal of Information Security and applications 22* (2015), 113–122.

[46] Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L. F., Hong, J., and Nunge, E. Protecting people from phishing: the design and evaluation of an embedded training email system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), pp. 905–914.

[47] Lastdrager, E., Gallardo, I. C., Hartel, P., and Junger, M. How effective is anti-phishing training for children? In *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)* (2017), pp. 229–239.

[48] Levin, M. Virtual kidnapping: The latest in an endless stream of scams. *F5* (2017).

[49] Lodhi, N. Beware of microsoft tech support scammers. *Business 2 Community* (2014).

[50] McGuire, W. J. Resistance to persuasion conferred by active and passive prior refutation of the same and alternative counterarguments. *The Journal of Abnormal and Social Psychology 63*, 2 (1961), 326.

[51] McGuire, W. J. Inducing resistance to persuasion. some contemporary approaches.

[52] Michie, S., Van Stralen, M. M., and West, R. The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implementation science 6*, 1 (2011), 42.

[53] Microsoft. Protect yourself from tech support scams. *Microsoft.com* (2020).

[54] Mitnick, K. D., and Simon, W. L. *The art of deception: Controlling the human element of security.* John Wiley & Sons, 2003.

[55] Mouton, F., Leenen, L., Malan, M. M., and Venter, H. Towards an ontological model defining the social engineering domain. In *IFIP International Conference on Human Choice and Computers* (2014), Springer, pp. 266–279.

[56] Mouton, F., Leenen, L., and Venter, H. S. Social engineering attack examples, templates and scenarios. *Computers & Security 59* (2016), 186–209.

[57] Nelson, R. Methods of hacking: Social engineering. *the Institute for Sistems Research, University of Maryland.(http://www. academia. edu/4903480/Methods_ of Hacking-social Engineering), diakses 10* (2001).

[58] Omroep Gelderland. Kamervragen over nepbom, minister wil procedures echt gaan uitvoeren. *Omroep Gelderland* (2018).

[59] Pars, C. Phree of phish: The effect of anti-phishing training on the ability of users to identify phishing emails.

[60] Pfleeger, S. L., Sasse, M. A., and Furnham, A. From weakest link to security hero: Transforming staff security behavior. *Journal of Homeland Security and Emergency Management 11*, 4 (2014), 489–510.

[61] Politie. Cybercrime. *politie.nl*.

[62] Politie. Whatsappfraude (vriend-in-noodfraude). *politie.nl* (2020).

[63] Politie. Criminaliteit 2020: minder inbraak, meer cybercrime. *politie.nl* (2021).

[64] ANP. Oplichters sturen mails over coronavirus 'namens WHO'. *RTL Nieuws* (2020).

[65] (Betaalvereniging Nederland). Pas op voor valse sms, zogenaamd van uw bank. *https://www.veiligbankieren.nl/* (2019).

[66] (Europol). Beyond the pandemic how covid-19 will shape the serious and organised crime landscape in the eu.

[67] Thinkfun. Distraction card game of memory and hilarious diversions. *thinkfun* (2011).

[68] Rivner, U. Anatomy of an attack. *RSA [database online]* (2011).

[69] Roozenbeek, J., and Van Der Linden, S. The fake news game: actively inoculating against the risk of misinformation. *Journal of Risk Research 22*, 5 (2019), 570–580.

[70] Roozenbeek, J., and van der Linden, S. Fake news game confers psychological resistance against online misinformation. *Palgrave Communications 5*, 1 (2019), 1–10.

[71] Scheeres, J. W. Establishing the human firewall: reducing an individual's vulnerability to social engineering attacks. Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ..., 2008.

[72] Security.nl. Ceo-fraude kost bioscoopketen pathé 19,2 miljoen euro. *Security.nl* (2018).

[73] Stajano, F., and Wilson, P. Understanding scam victims: seven principles for systems security.

[74] Stajano, F., and Wilson, P. Understanding scam victims: seven principles for systems security. *Communications of the ACM 54*, 3 (2011), 70–75.

[75] Stegeman, A. Undercover in Nederland - Season 4, episode 2, 2008.

[76] Stegeman, A. Undercover in Nederland - Season 10, episode 3, 2016.

[77] Stegeman, A. Undercover in Nederland - Season 15, episode 8, 2018.

[78] Streetlab. Streetlab - kun je zomaar een weg open breken?, 2015.

[79] Timmermans, J. The relation between the organizational information security climate and employees' information security behavior.

[80] Treglia, J., and Delia, M. Cyber security e-noculation. In *NYS Cyber Security Conference, Empire State Plaza Convention Center, Albany, NY, June* (2017), pp. 3–4.

[81] Van der Linden, S., and Roozenbeek, J. Psychological inoculation against fake news. *The Psychology* (2020).

[82] Von Solms, R., and Van Niekerk, J. From information security to cyber security. *computers & security 38* (2013), 97–102.

[83] yeung, J. Chinese students in australia are being scammed into faking their own kidnapping. *CNN* (2020).

[84] Young, H., van Vliet, T., van de Ven, J., Jol, S., and Broekman, C. Understanding human factors in cyber security as a dynamic system. In *International Conference on Applied Human Factors and Ergonomics* (2017), Springer, pp. 244–254.

[85] Zelster, L. Malware infection that began with windshield fliers. *SANS ISC InfoSec Forums* (2009).

Appendix A: Scene Overview

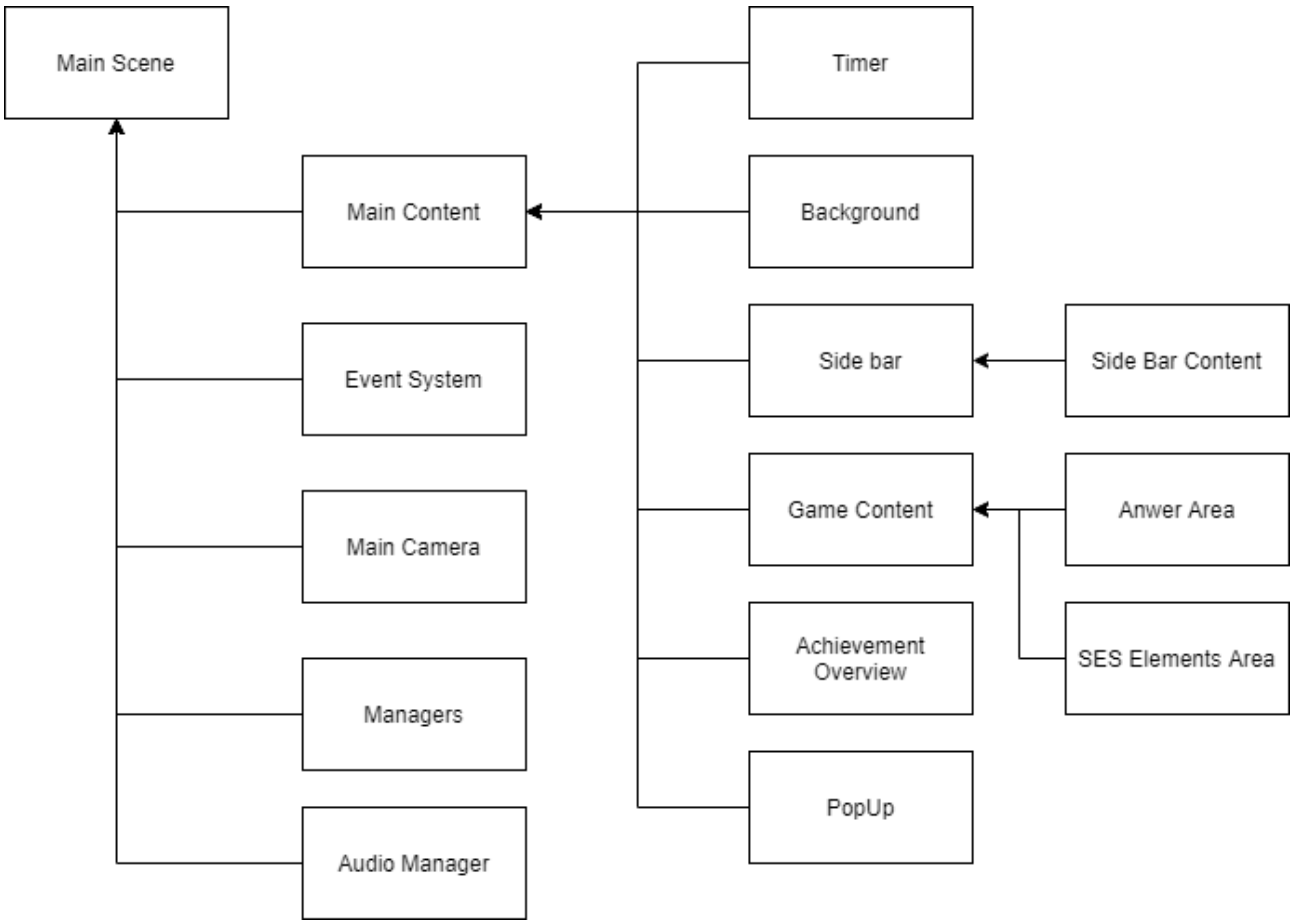


Figure A.1: UML Diagram of the Scene used in the Social Engineering Simulator (SES).

Appendix B: UML of scenario editor scripts

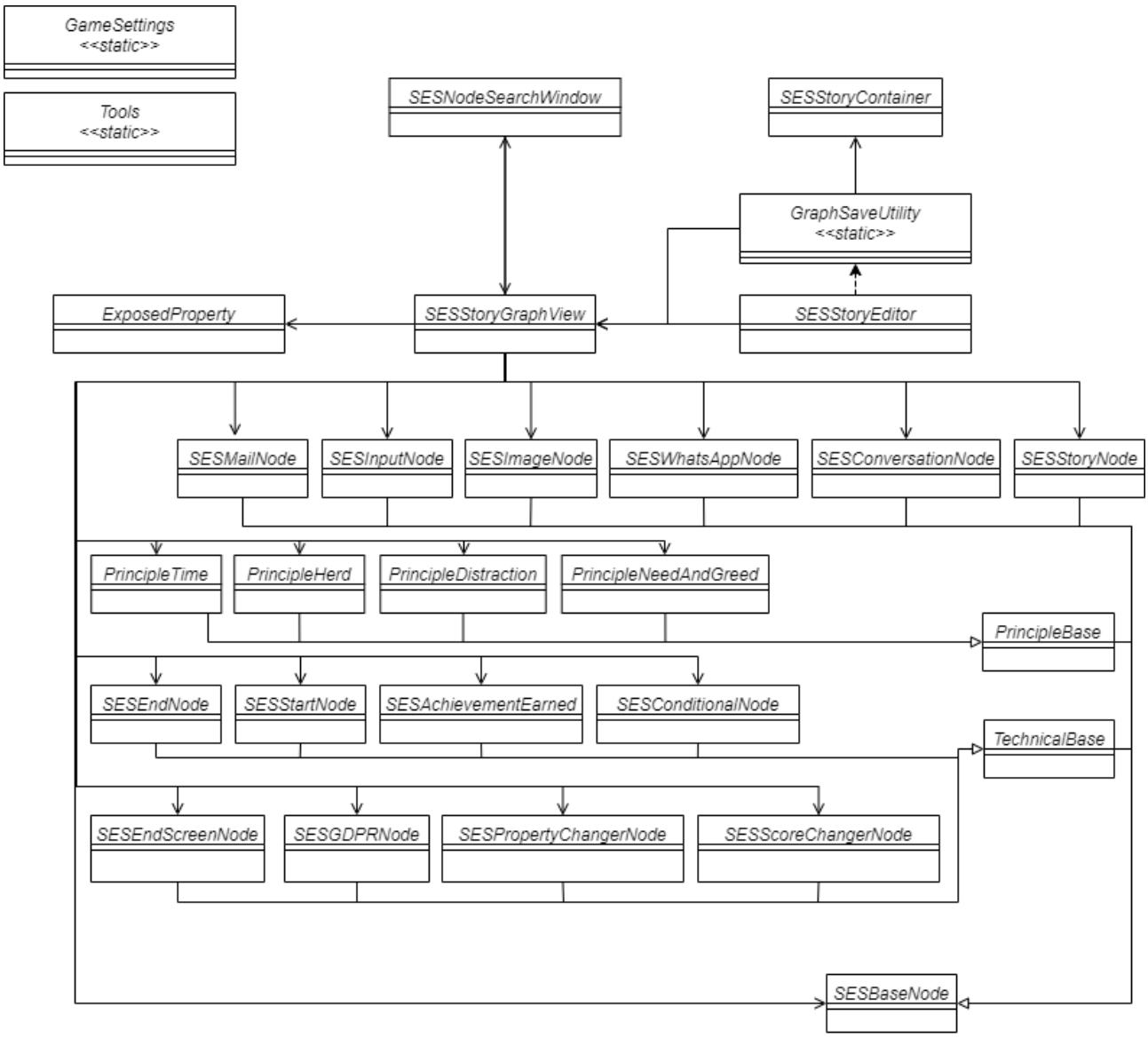


Figure B.1: Simplified UML Diagram of the scenario editor scripts in the Social Engineering Simulator (SES).

Appendix C: UML of game behavior scripts

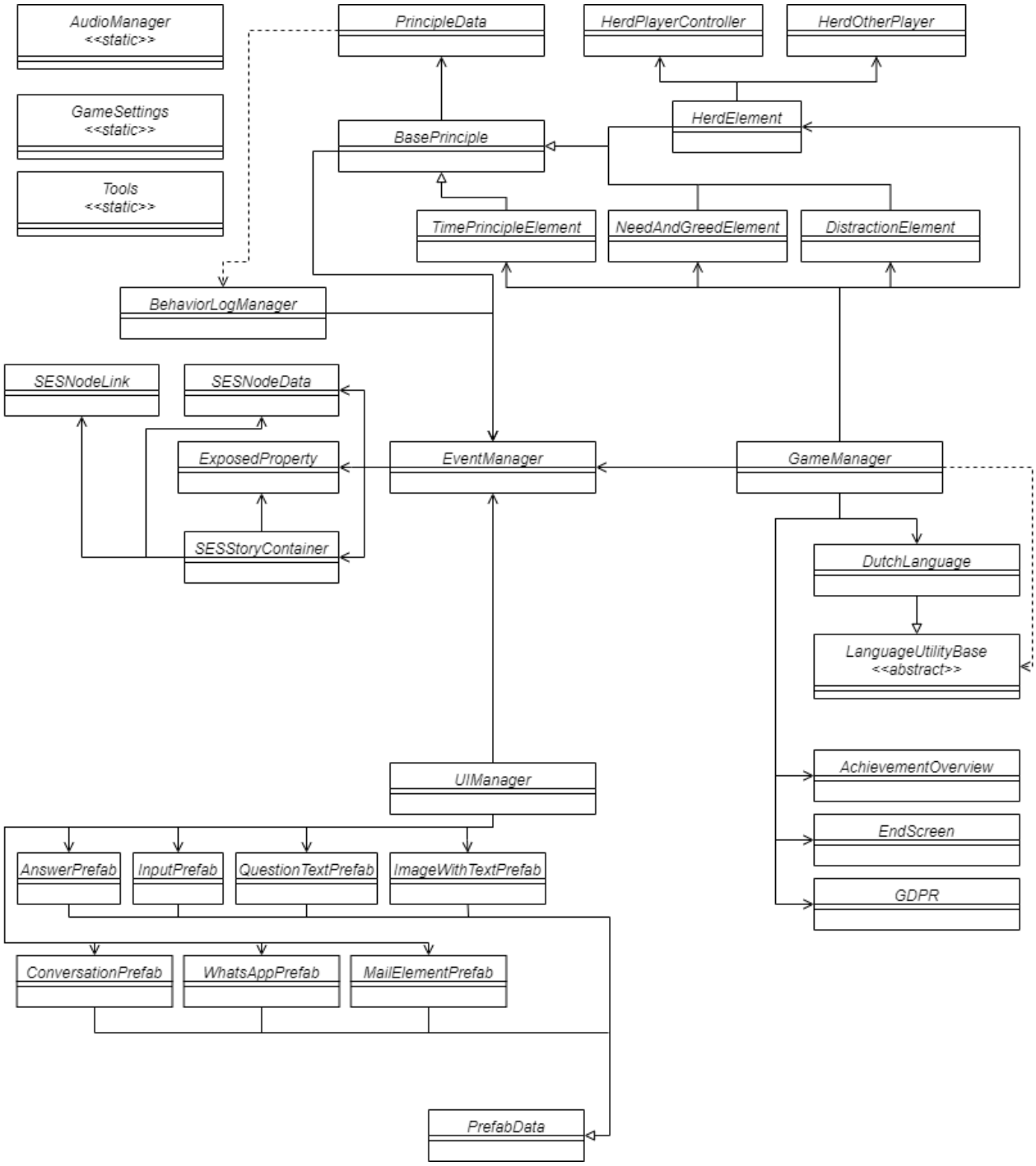
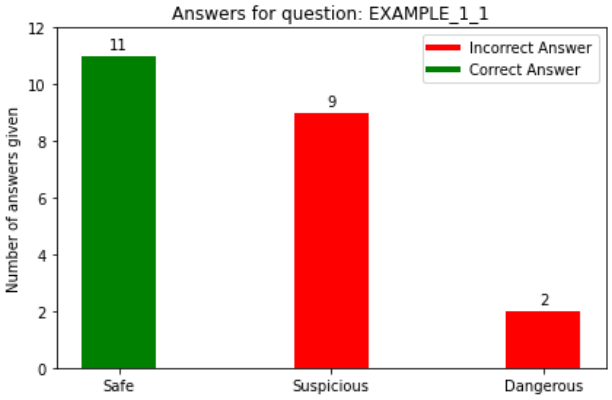


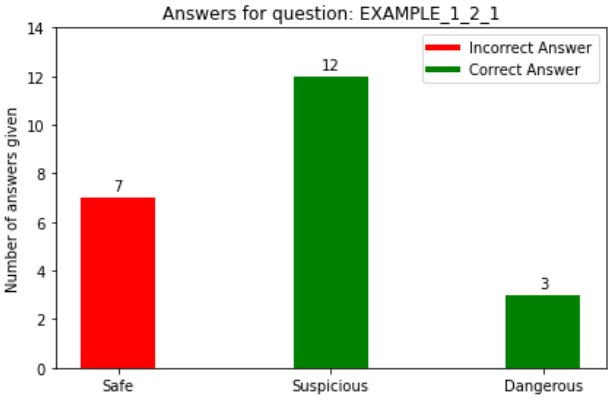
Figure C.1: Simplified UML Diagram of the game behavior scripts in the Social Engineering Simulator (SES).

Appendix D: Answers given for the question of the Social Engineering Simulator

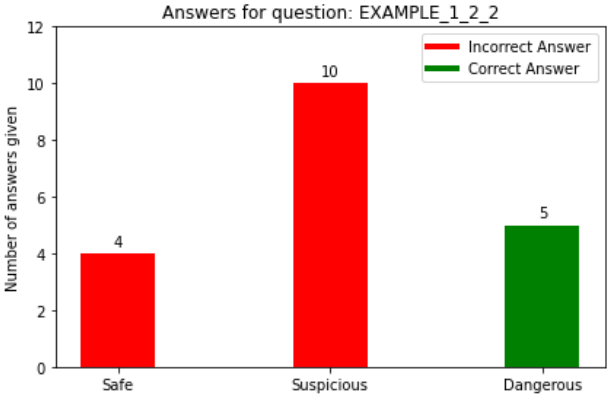
D.1 First Round



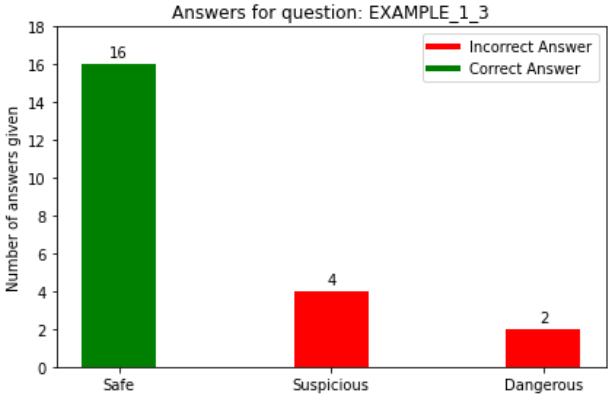
(a) Answers for example 1, first prompt.



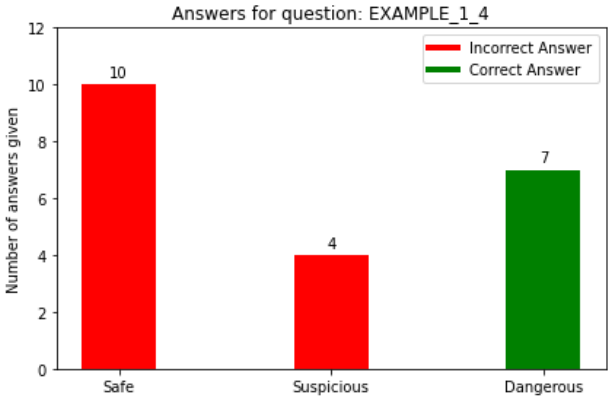
(b) Answers for example 2, first prompt.



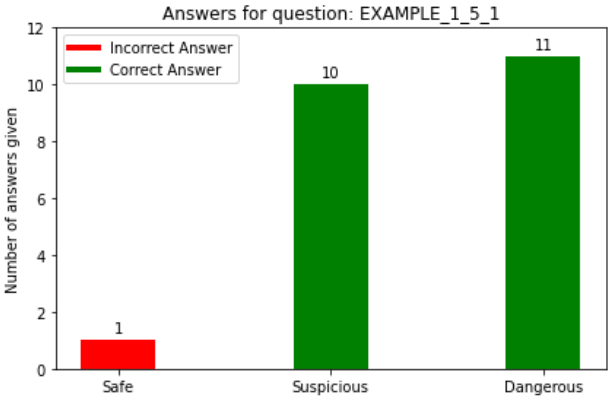
(c) Answers for example 2, second prompt.



(d) Answers for example 3, first prompt.

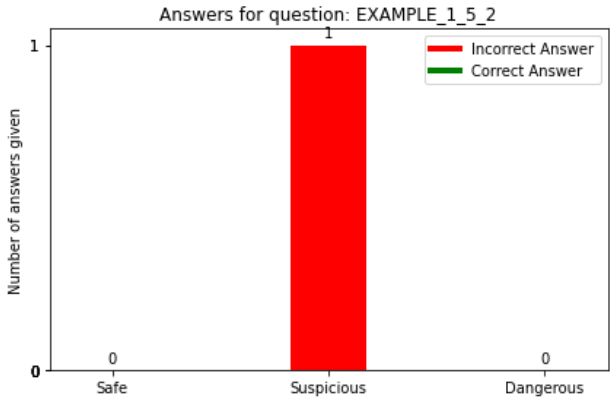


(e) Answers for example 4, first prompt.

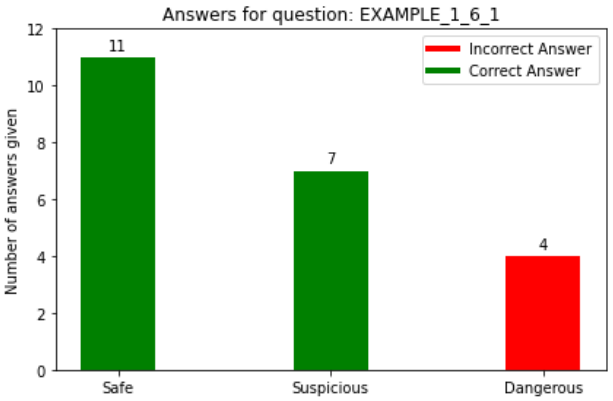


(f) Answers for example 5, first prompt.

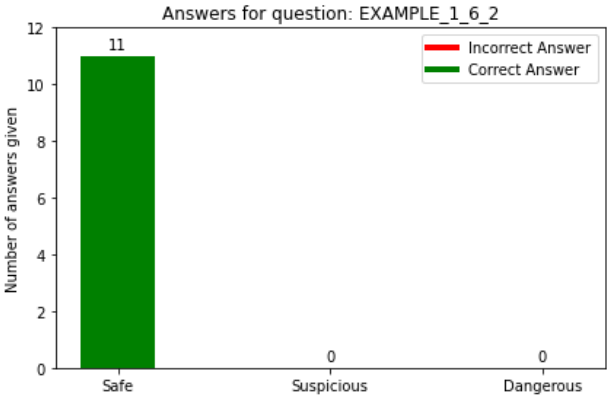
Figure D.1: Answers given for question 1 till 5 of the first round of examples of the Social Engineering Simulator.



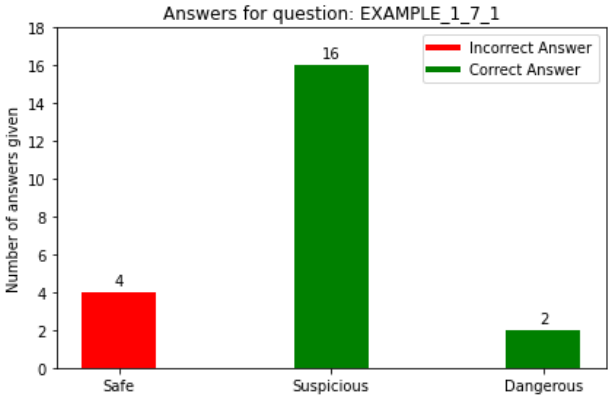
(a) Answers for example 5, second prompt.



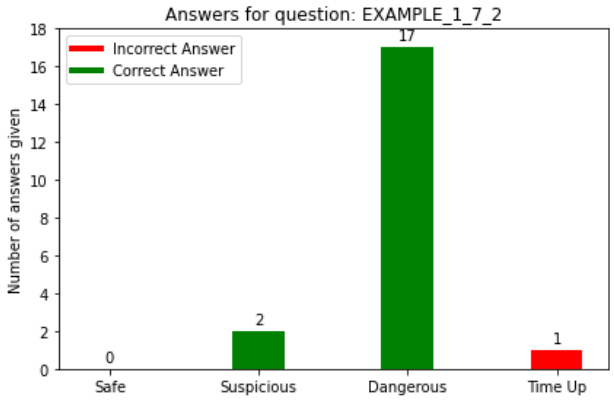
(b) Answers for example 6, first prompt.



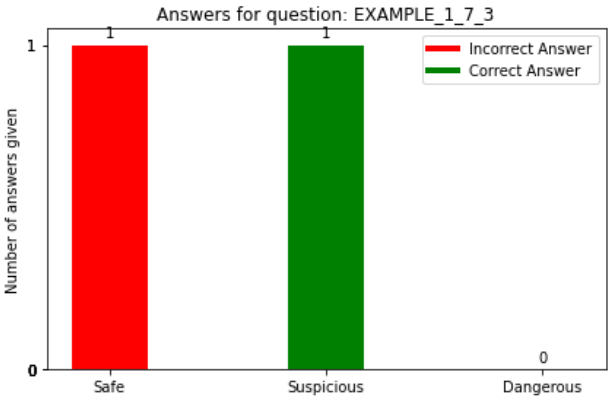
(c) Answers for example 6, second prompt.



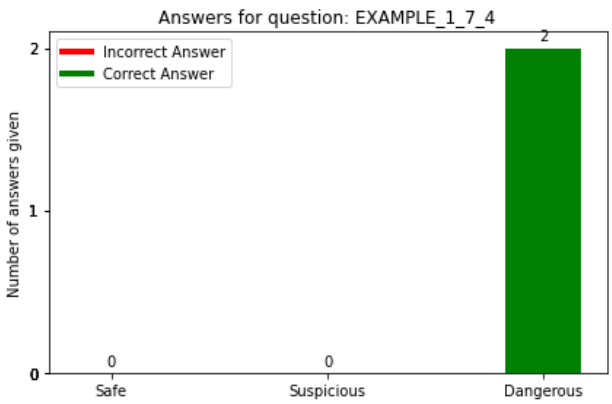
(d) Answers for example 7, first prompt.



(e) Answers for example 7, second prompt.



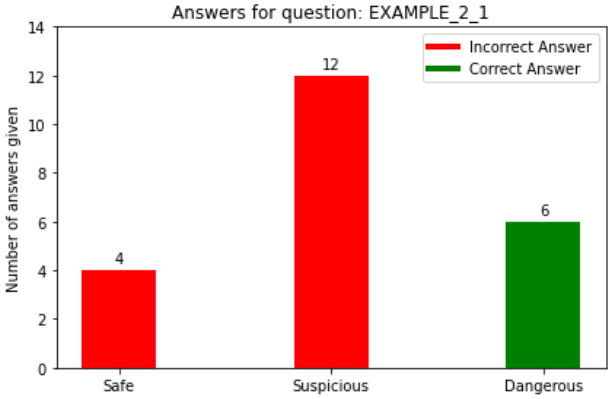
(f) Answers for example 7, third prompt.



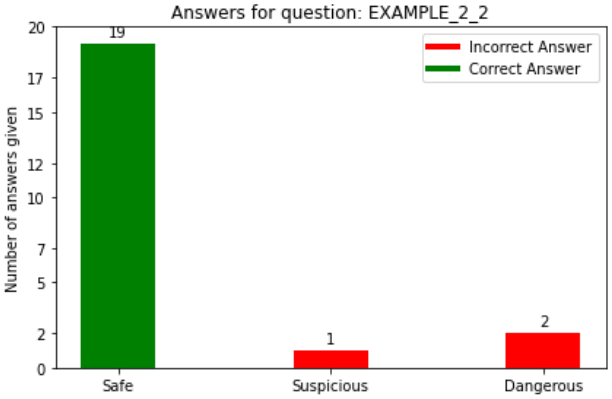
(g) Answers for example 7, fourth prompt.

Figure D.2: Answers given for question 5 till 7 of the first round of examples of the Social Engineering Simulator.

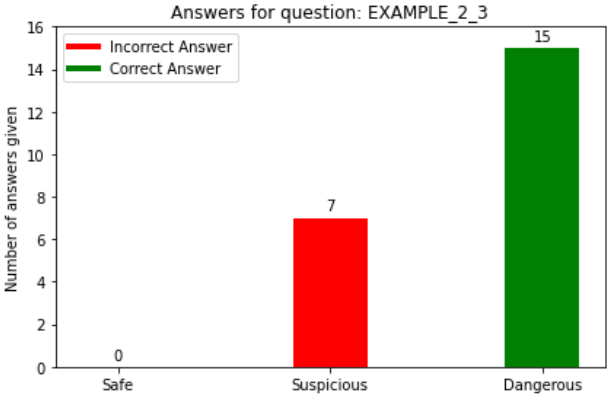
D.2 Second Round



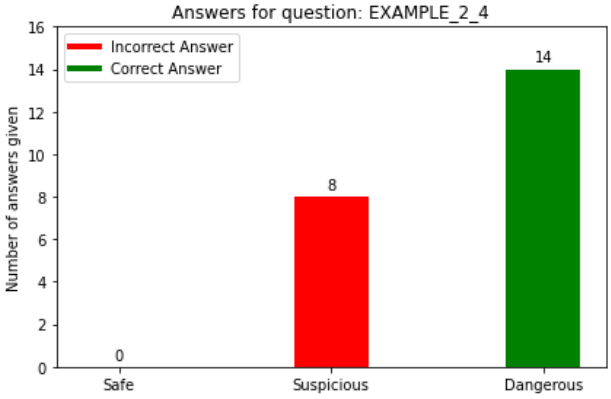
(a) Answers for example 2, first prompt.



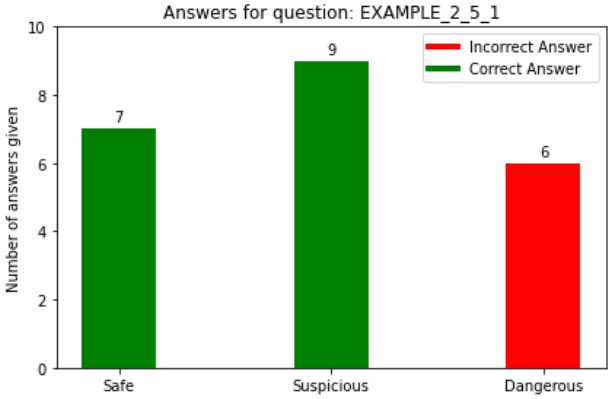
(b) Answers for example 2, first prompt.



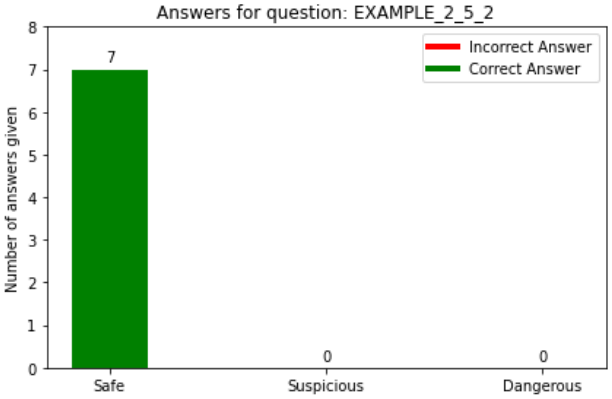
(c) Answers for example 3, first prompt.



(d) Answers for example 4, first prompt.

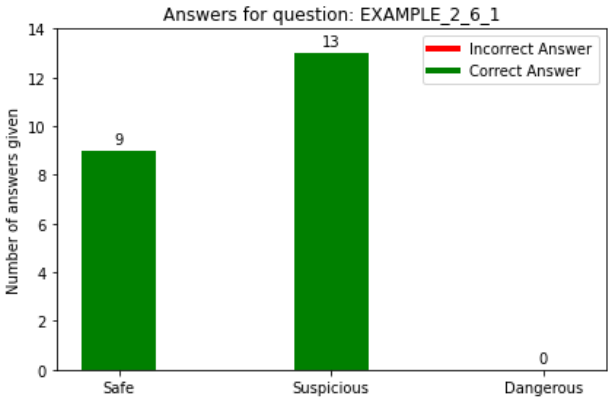


(e) Answers for example 5, first prompt.

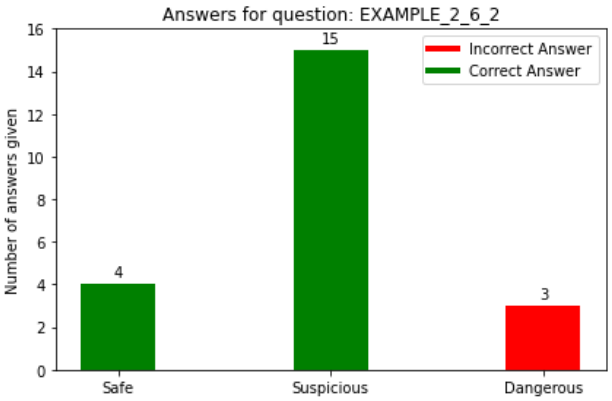


(f) Answers for example 5, second prompt.

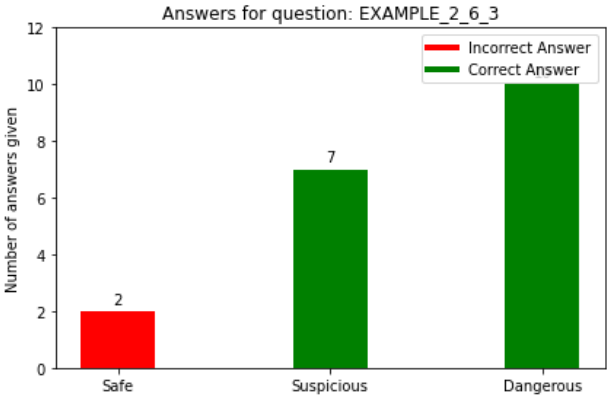
Figure D.3: Answers given for question 1 till 5 of the second round of examples of the Social Engineering Simulator.



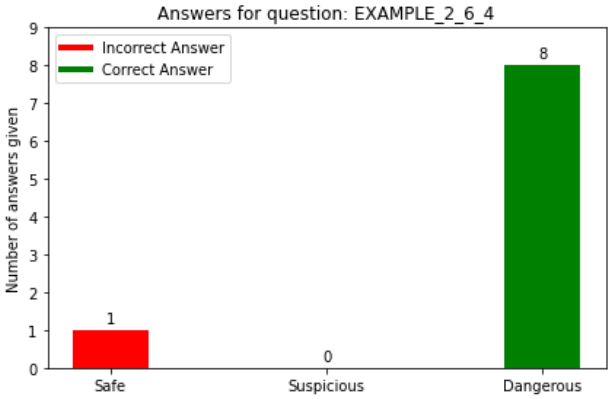
(a) Answers for example 6, first prompt.



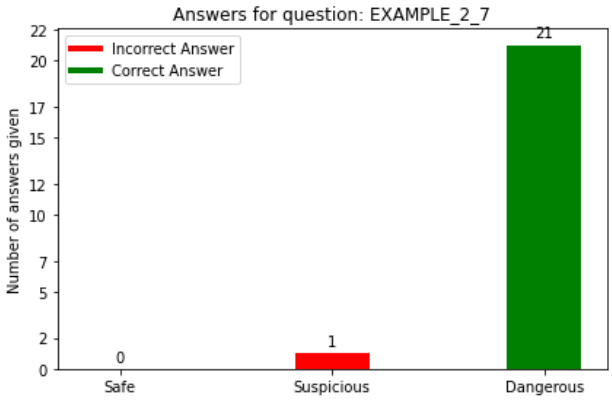
(b) Answers for example 6, second prompt.



(c) Answers for example 6, third prompt.



(d) Answers for example 6, fourth prompt.



(e) Answers for example 7, first prompt.

Figure D.4: Answers given for question 6 till 7 of the second round of examples of the Social Engineering Simulator.