Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Marie Johansen

Development of a customized remote access trojan (RAT) for educational purposes within the field of malware analysis

Master's thesis in MIS4900
Supervisor: Mass Soldal Lund
Co-supervisor: Geir Olav Dyrkolbotn
June 2022

NTNU
Norwegian University of
Science and Technology

Marie Johansen

Development of a customized remote access trojan (RAT) for educational purposes within the field of malware analysis

Master's thesis in MIS4900
Supervisor: Mass Soldal Lund
Co-supervisor: Geir Olav Dyrkolbotn
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Kunnskap for ei betre verd

Acknowledgement

I would like to take this opportunity to first of all thank my father and mother for always supporting and helping me. I could not have asked for better supervisors.

Mass Soldal, my external supervisor, and Geir Olav Dyrkolbotn, my internal supervisor, have also deserved my thanks and appreciation as well. Mass has taken his time to assure that I stay on track and has helped me out with all aspects of the project. Geir Olav gave me great guidance in the beginning and has also shared his thoughts and expertise within the topic. I am also grateful for being able to cooperate with the Norwegian Cyber Defence Academy that was willing to trust me with this project and taking their time to work with me.

Abstract

There are hundreds of thousands new malware samples registered each day by different vendors and institutes, and the numbers keep increasing for each year. Most people and corporations are possible targets and there has already been seen some most serious, sophisticated, and complex cyber-attacks that had devastating consequences. The fast growing numbers of malware has created an educational gap within the field of malware analysis for years. It is however predicted that the work within the cyber domain will be the 10th fastest growing occupation for the next decade. There is already a lot of courses provided by both the public and the private sectors and this thesis investigates the educational aspects of malware analysis and suggests using self-made malware as an educational approach.

The developed malware is adapted towards an educational setting and has thus configurations and functionalities directed towards learning objectives within the field. The developed malware was used as a mean to collect the data on the use of self-made malware in an educational setting and was thus also tested in an educational environment where students and educators were the targeted audience. Datta was collected through qualitative research using phenomenological approach and semi-structured interview as method. The development phase resulted in a remote access trojan (RAT) that encompasses the ability to communicate to an infected device, in addition to other common functions found in malware. Having the ability to execute further commands on a client makes it situational adaptable and can thus be used to mirror different threat actors all depending on what learning objectives wanted to cover. However due to its simplicity, the malware is though limited to illustrate less sophisticated threat actors and it is needed for additional malware samples to cover additional desired learning objectives in order to support knowledge and skill progress.

Sammendrag

Hver dag blir oppdages og registreres store mengder ondsinnet programvare i forskjellige databaser hos forskjellige institutter, selskaper og leverandører. Tallet på eksisterende skadevare har økt drastisk for hvert år som går, og de aller fleste personer og firmaer er ett potentielt mål. Opp gjennom årene har det allerede blitt sett flere store og svært skadelige, komplekse og sofistikerte skadevarer, og på grunn av den raskt-akselererende økningen i skadelige programvare og det allerede store antallet forskjellige typer som finnes, har det vært vanskelig å få dekket den arbeidskraften som kreves innenfor feltet. Arbeid innenfor data sikkerhet, som inngår blant annet skadevare analyse, er imidlertid spådd til å være en av de 10 voksende yrkene gjennom det neste tiåret og det finnes allerede flere kurs for grunnleggende kunnskapslæring på nett, hos universiteter og private sektorer.

Denne oppgaven undersøker de pedagogiske aspektene ved skadevare analyse og foreslår bruk av selvlaget skadevare som en pedagogisk tilnærming. Den utviklede skadevaren er tilpasset utdanning og har dermed konfigurasjoner og funksjonaliteter rettet mot læringsmål innenfor feltet. Data ble samlet inn gjennom kvalitativ forskning ved bruk av fenomenologisk tilnærming og semistrukturert intervju som metode. Den utviklede skadevaren ble brukt som et middel til å samle inn data om bruken av selvlaget skadevare i en pedagogisk setting og ble dermed også testet i et pedagogisk miljø der studenter og lærere var målgruppen. Resultatene viste at den utviklede skadevare omfatter flere fordelaktige aspekter ved læring ettersom den enkelt kan tilpasses basert på ønsket læringsutbytte. Imidlertid er den skadelige programvaren begrenset til å illustrere mindre sofistikerte trussel aktører, og det er behov for ytterligere skadevare eksempler for å dekke flere ønskede læringsmål og for å støtte en progresjon i læringen.

List of Contents

List of figures

List of tables

Chapter 1

1 Introduction

1.1 The importance of malware analysis

Within the Cyber security community, Malware analysis plays an essential role. It is the key process for identifying core functions, determining capability, and understanding the true purpose of a malware sample. According to the AV-TEST Institute, over 450,00 new malware samples and potentially unwanted applications (PUA) are registered each day [1]. McAfee reports a significant increase in several threat categories from Q3 to Q4 in 2020, including a grown of 199% in office malware, 69% growth in ransomware, 6% growth in new Linux malware, and 118% growth in mobile malware driven by SMS. In total, over 1,200,000 new malicious signed binaries were reported in Q4 of 2020 [2]. In addition to these numbers, it also worth mentioning how expensive cybercrime truly is. Numbers show that data breaches cost businesses an average of $3.62 million [3], and if it were considered as an economy, it would be the third largest in the world as it cost us almost $6 trillion annually in 2021 [4].

Facing these numbers is a challenge left to the everyday users of technical devices and systems and the people educated and eager to deal with it. On a daily basis, users and systems are at risk of being victims of vulnerabilities that cyber adversaries have chosen to take advantage of, while some do their best to protect these very same users and systems. Yet, the number of malicious software speaks for itself, and the ones working against it and trying to keep up are simply not enough. The number of people with appropriate expertise and experience is far below what's needed, which is a much bigger number than many realize. The 2021 (ISC)² Cybersecurity Workforce Study showed that the global cybersecurity needs to grow by 65% to effectively defend organizations critical assets [5].

According to the U.S Bureau of Labor Statistics, it is however believed that there will be a 32% growth in information security careers through 2028 [6]. They also predict that the work in the cyber domain will be the 10th fastest growing occupation for the next decade, where the grow rate is at 31%, which is 27% more compared to the average growth rate for all occupations [4]. This being the case, good educational options must be in place.

Analysing possible malware may help to identify unknown vulnerabilities which were used and exploited by the attacker, which often leads to the development of security patches as well as blocking of revealed indicators of compromise. Further, knowing how the malware works is of great importance for incident responders and security analysts. Having this knowledge contributes to better assessment and damage control, mapping out the severity and the extent of damage done, and figuring out how to contain and stop it from spreading further [7]. The source of the attack can be identified through malware analysis and additionally be used to get a better overview of the sample by comparing it to similar known samples from the same source. Intrusion detection rules and other signatures can also be developed in order to prevent future malware-based incidents and infections of same or similar nature together with other mitigation methods based on a particular sample.

As with everything else, having the skills to use detection software, making detection rules, recognizing, and analysing malware, collect and storing artifacts, and overall just handle the different aspects that follows when working within the field of information security requires practise and education. In today's society, this is most wanted skills due to the ongoing and constant battle in the cyber domain between attackers and those who work within security.

Teaching within malware analysis involves several stages and encompasses a broad range of topics within the field. Courses may include investigation strategies, some static and dynamic analysis, tools that can be used, and importantly, how to safely do a set-up of a testing environment. Ethical and legal considerations are also normally introduced to the students. While teaching malware, extra caution is important as it can lead to an unwanted infection that, based on the malware, may cause great damage. Nevertheless, the importance of teaching how to analyse potentially malware after a detected or suspected intrusion is crucial to keep up-to-date with hackers and cyber threats and keep information and data safe and secure.

1.2 Problem investigated in this thesis

The number of existing malware binaries that can be used as examples when teaching malware analysis are thousands, if not millions. Nevertheless, selecting the best-suited malware samples when teaching the basics of malware analysis can be difficult due to the amount. Malware samples vary in complexity, use of functionality, use of language, level of obfuscation, targeted system, etc. Thus, identifying and verifying samples that covers wanted learning objectives may be difficult and time-consuming.

The process of choosing what malware samples used in an educational setting needs to mirror the wanting learning outcomes. When finding something suitable during a search, the educator may want to do a manual analysis of the samples for further review. The review will provide an overview of the learning objectives covered by a specific malware sample, and the remaining desired functionality that needs to be covered by another sample. Based on the complexity of the malware found, it might also be expected to use time disabling unnecessary functionality that either is out of scope or might cause unwanted harm. Furthermore, suppose a sample is wanted to illustrate a scenario to give the students the possibility to learn the whole process from discovered infection to the analysis of the malware. In that case, a sample might also need to cover the attack vector. Also, researching malware needs to be met with caution as it is a most serious task.

1.3 Aim of the thesis and limitations

The topic in general is within the area of malware analysis for beginners and the thesis is done in cooperation with the Norwegian Defence Cyber Academy and the Norwegian University of science and technology (NTNU). When speaking of malware analysis in this thesis, the term will be not be limited to the malware analysis itself (static and dynamic analysis) but cover additional aspects within the field such as malware detection, and network traffic analysis etc.

In this thesis, the aim is to look at the use of self-made malware in an educational program within malware analysis and to investigate what is considered to be educationally friendly malware. The suggested approach will contribute in terms of providing a customized malware sample that covers some desired learning objectives. The developed malware will thus focus on some key functionalities and techniques used by different types of malware. Using self-developed malware allows for being in total control of the source code, to expand and add wanted functionalities, and customize the code to for example fit different scenarios or levels of skill-sets.

The malware to be developed is limited to some key functionalities and does focus on common techniques used to illustrate some learning objectives. Also, as it aims of being educationally friendly, implementing difficulty levs to in term of analysis to one of the functionalities will be added if time.

Investigating and implementing a suggested attack vectors is however out of scope, as the project does not aim at covering learning objectives within the infection phase.

1.4 Research Questions

In regard to the problem investigated, and the aim of the thesis, the following research questions are to be investigated and answered during this project:

High level:

- In what ways are developing customized malware, for learning purposes only, beneficial for a learning environment and how does this contribute to reaching desired learning objectives.

Low level:

- What kinds of malware are seen as educationally friendly, and why?
- What are some of the techniques and functionalities normally used by malware?
- Is there any significant differences in the learning outcome from the use of self-made malware versus real malware samples found online?
- In what ways is the developed malware beneficial?
- What malware frameworks does already exist?

The approach taken to answer these questions are a literature study, a development phase for a malware sample, which are to be tested on students in an educational setting, followed by a qualitative study with interview as the method. Each of these methods and how they were performed are elaborated further on in chapter 2.

1.5 Planned contribution

This master thesis aims to contribute with a customized piece of malware that will cover some of the learning objectives desired in an educational plan for malware analysis courses. The goal is to test the developed software in an educational setting in order to learn its pros and cons. As a result of a such self-made malware sample, and an analysis of lived experiences during the usage of a such sample in teaching setting, this thesis will provide insight into the benefits that comes with using a customized malware and how both the educators and students experiences it. The results of this project will hopefully provide additional knowledge and educational insight in the field of malware analysis and can be used as inspiration for future learning and educational programs.

The results of the development phase will contribute to a piece of malware with focus on key functionalities, scalability, and difficulty levels within functionalities. Scalability allows for adding additional wanted functions based on desired learning objectives. However, introducing difficulty levels and ideas on where it can be implemented is the main contribution in regard to software development. It is a component that emphasize educationally friendly malware and represents ways of learning. The goal is to design towards education, not towards the cybercrime domain. The thesis is done in cooperation with the Norwegian Cyber Defence Forces and the Norwegian Defence Cyber Academy and will because of this mainly contribute to their educational environment.

1.6 Background

1.6.1 Malware history

Malware is a collective term for all kinds of malicious software doing unwanted and damaging activities on an infected host, and there exist lots of different types. Depending on who made it and for what reason, the objective of a malicious program varies. Properties and functionalities vary, as does the level of sophistication. However, regardless of whether it is some generic piece of malware or some more sophisticated and targeted one, most malware is typically designed to gather some sort of information, have logging capabilities, do resource extraction, and paralyze or destroy files and systems. Most malware is also named or categorized by its key functionality, and depending on who you ask, there will be different answers regarding what the different types are. There are however some types that are more recognized and widespread than others, such as viruses, worms, trojans, ransomware, and spyware. This is illustrated in figure 1.

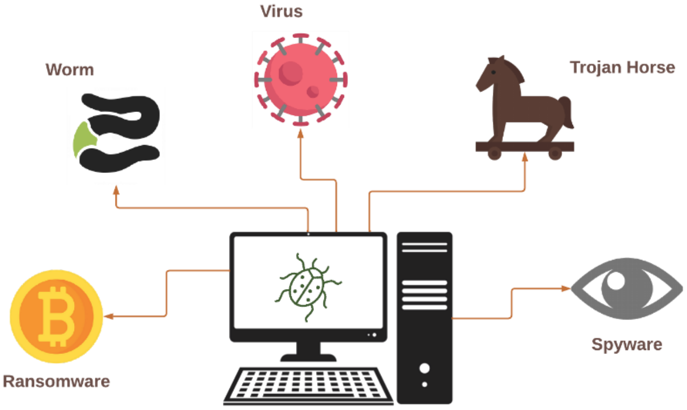


Figure 1: Illustration of common malware types that exists

A device, network, or system can be attacked by the use of different methods, and upon success/if successful, malware may deliver its payload resulting in infection. Some well-known methods includes hacking (which is a rather broad term and basically all forms of activity that helps a person gain unauthorised access to computer systems, networks, personal information, and other digital devices), SQL injection, zero-day attack, social media, social engineering, and phishing. Many of these methods and different types of malware are often combined and used at different stages of a cyber-attack. After either a successful or non-successful attack, the possible infected host is often thoroughly exanimated for further investigation.

The idea of what we know as malware today was first experimented with in 1949 by a computer scientists named John von Neumann. He wrote a paper named "Theory and Organization of Complicated Automata" where a program reproducing itself was the topic up for discussion. In 1970s, "Creeper Worm" which was a proof of concept and spread through the ARPANET (Advanced Research Project Agency Network), was written by Bob Thomas and is recognized as the first virus [8]. In 1982, the first Mac virus was introduced to the world and was named "Elk Cloner", and in 1986, "Brain" was first seen and is known as the first PC virus [9]. The virus was developed by two brothers from Pakistan. From here things slowly started to escalate, the Morris worm, which was a proof of concept, was created and as we know today, there exists malware in all imageable forms using all these different techniques and ways for reaching its objectives.

To mention some of the most famous malwares used in attacks, we have among others ILOVEYOU, WannaCry, Emotet, NotPetya, ZeuS, and Stuxnet [10]–[13]. They all had devastating consequences and did serious damage to their victims, causing trouble that was possible way out of their victims imagination. In table 1, some additional known malwares are mentioned as well as its type and year it was discovered. Additional to malware, tools that allows for customizing and have a multi-functional payload capability, and further features that provides different activity options has risen the last decade. Metasploit and Cobalt Strike are two of the most well-known tools, both of which were originally designed to assist penetration testers when looking for vulnerabilities in a targeted network.

Table 1: Famous malware, type, and release year

Malware	Type	Year
ILOVEYOU	Worm	2000
Mydoom	Worm	2004
Zeus	Trojan	2007
Stuxnet	Worm	2010
Pegasus	Spyware	2011
NotPetya	Ransomware	2016
WannaCry	Ransomware	2017
Emotet	Trojan	2018

1.6.2 Metasploit

Metasploit Framework (MSF) [14], is an open-source framework (or hack tool) that contains a suite of tools useable for finding security vulnerabilities, enumerate networks, choosing exploits, and dropping payloads. Even though Metasploit is not a framework for developing malware itself it is a hacking tool that can be used for deploying the malware. A sort of reconnaissance or vulnerability scan is done, exploits of identified vulnerabilities are run, and some level of access is provided, resulting in having the possibility to install malware. Some most known tools of MSF include Meterpreter, MSFconsole, and MSFvenom. Meterpreter is used for establishing communication and is a shell that is deployed using in-memory DLL injection stagers. It features command history, tab completion, channels, and so on [15]. MSFconsole provides a command-line interface to access and work with the MFS [16], whereas MSFvenom can be used for generating payloads[17].

1.6.3 Cobalt Strike

Another well-known exploitation tool which has been increasing in popularity the last years is Cobalt Strike. First release of the tool was in 2012 [18] and is commonly used by penetration testers and red team testers. Its Beacon backdoor provides ways of being configurated to match the needs and wanting of testers, and by this also provides among others the possibilities to execute commands, download and execute additional software, and emulate different sort of threats. The way this tool differ from Metasploit, which is open source, is that the source code was leaked, which led to the rise of its popularity among cybercriminals as well. Further, Cobalt Strike works in conjunction with the Metasploit Framework. However, as with Metasploit, it is discussed among security professionals the ethical aspect of building such tools as it is almost indistinguishable from actual hacker tools. But the need for pen-testing is still there, so is there a good argument saying that latest malware available on the marked should be used instead? [19] Instead of developing tools for white hat and grey hat hackers, that later, and most certain, are to end up in the hands of black hat hackers as it eventually gets out of control.

1.7 Related work within the field of malware analysis

This section will provide a discussion on resources relevant to this thesis. Resources of relevance is divided into two sub-sections: Resources related to malware development and resources related to malware analysis courses. For both sub-sections, the resources will be presented individually with an introduction highlighting the aim and/or main contribution followed by a summarize of the content discussed. In what way the resource is relevant and how it differ from the work presented in this thesis is then discussed.

1.7.1 Malware development

1.7.1.1 Resource 1: Malware Development for Red Teaming Using Metasploit

Petros Katritzidakis wrote in 2018 a master thesis to automate Metasploit procedures while developing a malware [20]. The purposed automation script, named "Pwnwr", aimed at assisting in Red Teaming operations by integrating existing tools and automating their manual handling. The end result was a modified version of the framework Metasploit.

Pwnwr is written in Ruby, as Metasploit, and uses Meterpreter as its payload when deploying. Metasploit Framework was the main software used when developing the script together with MSFconsole and MSFvenom. Windows was the targeted operating system (OS) and a personal computer was used for testing. The testing environment was powered by a virtual machine and consisted of six machines, three of which was Windows 7 and three of which was Windows 10, in order to simulate a small network in order to demonstrate how Pwnwr could move latterly inside it. Kali Linux was however the OS used for development, and VMware as virtualization software. Further two scripts was created, each one following attack graphs representing Red Teaming attacks. The first attack graph regards creation of malware targeting OS used, whereas the second attach graph regards the exploitation of any vulnerabilities found on the targeted OS.

The master thesis focused on helping out Red Teamers with automating their operation when using MSF. The project differs in the usage of the framework when creating malware and deploying it, and in testing environment set-up. However, as results are not discussed and there is little information presented on the actual malware, it is difficult to discuss possible similarities. It is all although a framework for malware and deployment of it, which is similar to what is purposed to be developed in this thesis. Technical specifications and objectives are however different.

1.7.1.2 Resource 2: Malware Obfuscation Techniques: A Brief Survey

Obfuscation is a widely used technique by malware writers for evading antivirus programs. In [21], Ilsun You and Kangbin Yim explore and discusses obfuscation techniques used by malware that are able to avoid detection. This includes encrypted, oligomorphic, polymorphic, and metamorphic malwares. Additionally, the paper discusses possible future obfuscation trends.

The authors of this paper aimed at briefly surveying existing malware obfuscation techniques and presenting these. Before some known techniques was discussed, an overview of malware that was capable of evading antivirus scanners was described. This included encrypted, oligomorphic, polymorphic, and metamorphic malwares, each better adapted than the former. In contrast to the others, metamorphic malware has the ability to evolve its body into new generations, making it the most difficult to the detect of the one mentioned. The obfuscation techniques discussed are dead-code insertion, register reassignment, subroutine reordering, instruction substitution, code transposition, and code integration. Each technique is presented with a sample code.

In dead-code insertion, instructions are added to change its appearance and is a rather simple technique. This can however be defeated by just deleting the instruction, an example of this is the "nop" instruction. Register reassignment is also simple as it only switches registers from generation to generation while keeping the behaviour the same. Though, wildcard searching makes this obfuscation technique useless. Subroutine reordering randomly order a code's subroutines, and the technique can generate n! different variants where n is the number of subroutines. Instruction substitution takes usage of instructions as well, but instead of only adding one, the technique replaces used ones with equivalent ones. Code transpositions reorder the sequence of the instructions. This can be done using two different methods. One which shuffles the instructions before recovering the original execution order by inserting the unconditional branches and jumps, and one which creates new generations by choosing and reordering the independent instructions that have no impact on one another. Lastly discussed, code integration is a most sophisticated technique as it integrates itself into the targeted program. To achieve this, the targeted program must firstly be decompiled in order for the malware to seamlessly add itself between manageable objects, and then the programs is reassembled into a new generation.

As obfuscation is a highly common technique found in malware as its simply makes the program harder to understand and detect, this paper is most relevant regarding malware development for learning purposes. Obfuscation is not a priority, based on the scope for this projection, but will be listed as "wanted functionalities", so if there are time, this is one type of behaviour that can be implemented if wanted.

1.7.2 Malware analysis courses

1.7.2.1 Resource 3: Finding Educationally Friendly Malware

"Finding Educationally Friendly Malware" [22] was a master thesis presented by Aleksander Bjørkhaug in June 2021 and was done in cooperation with NTNU. The purpose of the research was to understand what educationally friendly malware is and to develop a framework for finding it in order to help educators spend less time on the task.

In this thesis, the researcher wanted to help educators in finding malware samples suitable for malware analysis courses. The end result was a framework that could take an arbitrarily set of malware samples as input, and from this generate a subset of potential educationally friendly malware as output, including an accompanying report that explains what is found and where. Before starting the development of this program, a study on experts thoughts regarding the definition of "what educationally friendly malware is" was accomplished. Semi-structured interview were used as method where persons with extensive knowledge within the malware analysis field were the subjects. This was persons that either worked for a company or an institute and each subject was identified as an educator or company.

The findings implied that identifying usable samples varies with complexity but overall takes more time than wanted. "It might take a month or so to make the course content around it, making slides and writing notes. I am quite careful when choosing samples because it takes so much time", was said by one of the interview subjects. Further, it was important to stay away from samples with online network addresses, destructive capabilities, and the use of too many ransomware samples. The findings were used as indicators to locate potential samples that could be used in courses. Then, different workflows and possible ways of doing malware analysis were investigated. The idea was to discover which tools to use for finding educationally friendly malware. The book "Practical Malware Analysis (PMA)" was used as a base for finding which functionalities were of interest, and that was possible to use as indicators when developing the framework.

The framework did a static analysis of the given malware samples, as dynamic would require the malware to run which would result in infection. Picky, which is the name of the framework, had an average analysis time per subset of 100 out of 40 000 on 4,27 seconds. When testing, fictional cases given from an educator was used where the researcher was to find samples representing a specific technique. By using the framework, a maximum of 10 minutes was spent from start until finding a suitable sample.

This master thesis differ in the method used for looking into how a course within malware analysis can be improved. Bjørkhaug has proposed and presented a framework for speeding up the process of finding suitable malware samples for education, whereas in this thesis, it is suggested to use self-developed malware as an approach instead, or additional, of finding malware samples online. Both thesis are similar in doing a study where what educationally friendly malware looks like is investigated through interviewing people with knowledge within the area.

1.7.2.2 Resource 4: Review of Pedagogical Principles of Cyber Security Exercises

Mika Karjalainen and Tero Kokkonen [23] presented in 2020 their work on extended research on cyber security exercises where their main contribution was within pedagogical principles and aspects of such exercises. The article addresses pedagogical principles and aspects of cyber security exercises focusing on learning outcomes in perspective of the exercise lifecycle which includes three phases: a planning phase, implementation phase, and feedback phase. Prior to these three phases are further described in the article, cyber security exercises and pedagogical principles are discussed. Lastly, a discussion of assessing performance and results was presented.

The pedagogical principles addresses the importance of having a modular learning environment simulating realistic and expected behaviour in the cyber domain in which learning objectives are implemented carefully and in accordance with students existing knowledge. The authors further suggest and discuss how a cyber arena, such as cyber ranges, should reflect a complex entity consisting of different parts that interact with each other. In such environment, skills and competencies learning in practice should be applied.

Key elements of learning theories presented includes among other developing an understanding of unpredictability of the environment, different roles found in teams and responsibilities shared with those roles, resolving realistic problems in a realistic environment, having the opportunity to build a path of competence development, allowing the learner to update existing knowledge, and developing learning objectives based on the level of student’s existing skill-sett. Elements found in these theories, methodologies and principles presented, such as Ericsson's deliberated practices (DP), the Miller pyramid, and constructive methodology, were further categorized into three main pedagogical principles used as implementation vectors for the pedagogical framework of cyber security exercises.

The three main principles include behaviourist design principles, cognitivist design principles and constructivist design principles. The principles should further be present in all phases of the cyber security exercise life-cycle: the planning phase, implementation phase, and feedback phase. The effort put into the planning phase determines the effectiveness and outcome of the exercise and is viewed as a most critical phase. The implementation phase is where the target audience is placed in a scenario with given roles. Situation awareness training gives great value and current expertise of the individuals reflects their ability in decision making as the situation develops. There should also be room for guidance. It is the responsibility of the exercise management team to ensure that students work towards desired learning objectives and based on actions made by the students and guide them either through system adjustments or verbally explaining instructions and clues. Though the planning phase does set the standard for the exercise, the feedback phase is the most important regarding to the development of an individual's competence. A review of pedagogical goals is conducted with reference to the actions performed and events that occurred. In this way, students have a chance to reflect on what they have experienced throughout the exercise.

For assessing performance and results, the authors discuss the use of different evaluation methods and frameworks. Kirkpatrick four-level assessment framework is widely used and divides the evaluation into the four levels: reaction, learning, behaviour, and results. Communication monitoring is another option. This allows for better understanding of the behaviour of the exercise target audience and can be used as a parameter in measuring performance. In terms of assessing, it has also been suggested to look into why, what, how, who and when. The significance and implementation of these sections should be planned in advance in order to gain most out of the assessment. The purpose of feedback is to reduce the gap between existing and target competence. Continuous feedback is of relevance in larger and complex learning settings, such as the cyber range setting. The purpose of feedback can be divided into three sections: Feed up, feed back and feed forward. Each building on each other starting with continuously clarify learning objectives, motivate, and engage in pursuing these objectives. Following up with feedback on where a student is seen in relation with the set learning objections, while feed forward provides with information on what the next step should be.

The article focused on pedagogical principles and aspects of cyber security exercises and provided a discussion of several different pedagogical frameworks that can be used in each phase of the exercise. In order for students to constantly gain new knowledge and utilize the existing one as they progress through the exercises, each events in each operation must be well-planned and lead to a set of learning objectives. In this thesis, malware analysis has been defined to cover additional areas within the field and is not limited to static and dynamic malware analysis. This article will for this reason be most relevant as it is within an area that deals with cyber security where malware and malware analysis is key topics. Deciding what learning setting the developed malware will be used in is out of scope for this thesis. It is though highly possible that it can be used in a cyber range setting as one case scenario presented but it might as well be used in a class where the focus is entirely on malware analysis itself and not all other aspects that is part of a realistic scenario. However, it is not the aim of this study to investigate the pedagogical aspects within the field, but it is relevant as this study aims to provide a customized piece of malware that does cover some of the desired learning objectives that can be found within the field. Hence, the principles discussed does apply indirectly to this study even though how it is possible to achieve those learning objectives by looking at theory applied to the subject is not discussed.

1.7.2.3 Resource 5: A case study in malware research ethics education: When teaching bad is good

Malware research and ethical considerations has been discussed by Sullins. J in [24]. The case study presented aimed at highlighting ethics in malware studies. The author states that himself has been the ethical adviser for a course where students learned malware programming.

One way to learn how malware is built in order to uncover among other its objectives, techniques used, and capabilities is to reverse engineer samples of malicious software. Another way is to learn how to program malware. Interacting directly with the design of malware gives students the opportunity to develop a comprehensive understanding of what goes into the creation and deployment of malware. Yet, designing and creating malware in a teaching environment needs to involve ethical considerations. Teaching malware programming is however not seemingly accepted or understood within the field. The author further argues that moral thinking and sentiment is a skill that needs to be individually developed and trained. "Even when we are just trying to deconstruct a piece of malicious programming, it requires that we think like the criminal that wrote it in the first place." Students needs to be capable of applying learned malware related skills ethically, either it is learned from a reverse engineering or programming malware course. Another argument proclaimed by the authors co-worker, Professor George Ledin, says that "education in computer security without courses where the students were taught how to design malware, would be like a medical science that tried to invent cures without ever studying any diseases".

The author further states the importance of acknowledging that working with malware is not ethically neutral and addresses some of the ethical problems encountered in the study of malware. Human subjects in malware research are discussed as one of the major concerns as without their knowledge, researchers might violate the principle of CIA (confidentiality, integrity, and availability) with respect to those human subjects related to the information researched. Taken an example provided in the paper: if a subject owns a machine related to an illegal botnet being researched, it does not mean that the research where these machines is a part of are ethical, even though it is legal. Leaving the victims as they were and additionally conduct research on their behave is not desired from an ethical view, but again it is impossible to obtain consent before the research began. A second issue regards information communication and publicity. Should research and findings be shared if it may pose a threat to the very same they are trying to protect? This may for example include research on new vulnerabilities, if they are to be published before whom it might concern can make countermeasures, it might be in favour of cybercriminals. Also, if the research is funded by private of governmental entities, it might hinder the results of the research being communicated and shared due to restrictions even though it can be of others interest to learn what the research has resulted in. These issues emerges from the fact that an ethical framework has not been properly designed and provided for the field. The ethical issues discussed has been taken into considerations when teaching malware programming in the course that the author is involved with. It is thus concluded in that the individual programmer is the one that decides whether to put on the black or the white hat and from a learning perspective they can only provide the ethical concepts they need in order to reflect upon the ethical implications of the work the students are making.

The aim in this case study was to highlight ethical aspects of malware research and to discuss issues encountered when teaching within the field of malware, and more specifically malware programming (when teaching bad is good). This project does not provide guidance on how to program malware but it does involve programming malware, thus making ethical aspects and consideration both relevant and an important part of this thesis which is also further discussed in the section 1.9.

1.8 Justification, motivation, and benefits

The development of a customized malware sample and the ethics around this approach are important to discuss as malware is nothing to joke about. Even though malware development is in a grey zone when it comes to ethical and legal considerations as it may have a damaging effect on a system or cause other unwanted problems, it does in fact exists already and there is a lot of it. The objective of this project is not to create something that does not already exists, or to identify new vulnerabilities to exploit as a part of the project, or in any way "help" cybercriminals to evolve and expand their current knowledge. In contradiction, the objective is to make a customized program that behave like most malware and have same key functionalities to ease the task of a lecturer when it comes to finding samples fit for education, and then to look into whether both the lecturer and students finds is valuable. As already discussed, this may be a time-consuming task. Additionally, often more than one sample are needed as normally one type of malware does not cover all desires areas and stages for an optimized learning outcome. The motivation is to create a scalable program and framework with key functions with the possibility to easily add new functions later that covers other behaviours. By making a such malware, this can further be used as a base for investigating in what ways it may be beneficial in an educational setting. The more areas it may cover, the better it is from an educational point of view. There are however some functionalities that may be wise and reasonable to exclude such as reproducing capabilities and functions that havoc and destroys the device upon infection. Ransomware is also something to be careful with. Ethical issues regarding the development of malware will be discussed and addressed in section 1.9.

1.9 Ethical and legal considerations

The ethical and legal considerations for this project involves discussion regarding whether or not malware should be developed for teaching purposes, malware analysis/reverse engineering, and the purpose of the qualitative study.

The project aims at contributing to knowledge distribution within the field of malware analysis, by providing a program that can be used in a scenario where students are to figure out what happened and how in terms of an infection. When the infected device is located, it will then be possible to conduct an analysis of the malware. As already shortly discussed, the objective of this project is not to create something that does not already exists, or to identify new vulnerabilities to exploit as a part of the project, or in any way help cybercriminals to evolve and expand their current knowledge. To the contrary, the objective is to make a customized program that behave like most malware and have same key functionalities, which then can be used to determine, based on a qualitative study, whether or not it has value. "Can this program be beneficial and in what ways", that is what this project is about. Having this in mind, I believe that the ethical issues are minor/few regarding this project. There is no intention to use what I learn while programming malware to any other use then applying it to the work I will do as a security analyst. Also, only using already existing known functions and methods will not in any way contribute to evolvement within the cybercriminal society. There exists a risk of misuse of course, but again, there will really be nothing new or sophisticated findings within this program.

While analysing malware and looking into reverse engineering, there are however important to have ethical and legal considerations in mind. Learning this can be applied to other applications and software that is not recognized as malware, and this is a risk as well. Knowing the objective and intentions of a person is not possible, and it is therefore important to learn ethics while learning reverse engineering as well. As a security analyst, reverse engineering, and knowledge on how to do it is crucial. There has however been cases where instead of using it to combat for example malware, it have been used as a tool for theft of intellectual property, and the law of copyright has been violated [25]. Nevertheless, there are no laws against it, and seen in context with cybersecurity, it is crucial to keep up to date with the hackers.

A qualitative study aims at observing and recording reactions, feelings, thoughts, and opinions on a "lived experience". Within this field or research, ethical and legal considerations are especially important to have in mind. Regardless of the topic being researched, confidentiality and self-awareness in internal biases and interpretations is important factors when it comes to building trust and mutual respect and ensuring that the participants interest is taken seriously. It is argued that particularly sensitive topics can pose emotional and other risks, which in case there should be clear protocols in place for dealing with distress [26]. Nevertheless, a guide should be provided where participants are informed and ensured that they are allowed to quit or take a break at any time, and that the interview is on their terms. Personal information will not be publicly available, and it will not be able to trace data collected to participants. For this project, the aim is to investigate if a course taking usage of customized malware is seen as useful and/or beneficial, leaving out topics seen as sensitive end emotional.

Chapter 2

2 Choice of methods

Through this thesis, a process consisting of four steps (methods) have been conducted in order to answer the research questions stated in chapter 1. The process is presented in figure 2. Prior to conducting the research, literature related to it was reviewed. This was the first step. Planning the type of malware which was to be developed and starting the development phase was the second step. After providing the malware with desired functionalities and conducting some internal testing, the program was ready to be tested in an educational environment where students was the main target audience. Presenting the case to the students are considered to be the third step in the process of getting material to analyze. The fourth and last step taken was qualitative research with interviews as method. Each method is further presented in the following sub chapters.

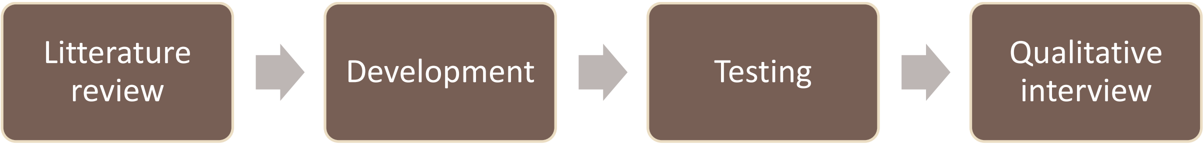


Figure 2: Illustration of process used to generate results

A modified approach of scrum was utilized throughout the whole project period. Measuring progress and knowing what to do, in what order, what is difficult, and to identify obstacles are important for being able to deliver a successful project to scheduled time. The following points were to be decided:

- Sprints of chosen duration
- A tool to keep track of tasks and everyday objectives
- Meetings at the end of each sprint with supervisor

Meetings held would include discussion of further work, what has been done, feedback, and possible adjustments if needed. By following this process, it will not only be easier to see progress and to identify problems during the project, but also to do adjustments before it is too late. Scrum specifications are presented in table 2.

Table 2: Scrum specifications

Sprint duration	Two weeks
Task tool	Notability
Meeting time	Initial every other Friday

2.1 Literature review

A literature review where previous related research on the topic of interest was presented and discussed in sub-chapter 1.7. The review included a summary of related resources and a discussion regarding it relevance. The review of published information contributes to avoidance of replicated research, support of theories, comparison, and inspiration. In the process of conducting a literature review, several steps were taken. Relevant literature can be found in a wide range of different sources and as stated by J. Rowley and F. Slack in [27], "One of the most intimidating aspects of a literature review is encountering the messy nature of knowledge". A lot of information are easily available today and locating and evaluating the information sources may be a time consuming task.

Using good search engines and have a clear opinion on what keywords and terms to use is an effective way of finding work of interest. Finding literature regarding malware techniques, malware development, and education was the main objective for this project, thus keyword like the one mentioned and similar ones was used in searches. For example, when looking into common malware techniques, the words "obfuscation techniques" and “malware” was one of the keywords used. In this case, the idea was to get knowledge about obfuscation techniques used by malware, which would further help in decision making of what techniques to implement in the development phase.

When searching, some different search strategies were applied. This included brief search and citation pearl growing. In brief search, few documents were quickly retrieved. In citation pearl growing, key terms of interest found in one document were used for further search of other similar documents. Both of which were mentioned in [27]. An additional approach used was to look at cited references in order to locate other work that might be of interest. Search engines used for this project included:

- Google
- Google Scholar
- IEEE

Evaluating information resources (papers, articles, journals etc) to conclude whether or not a resource is relevant is the next step in the process in finding literature of interest. Learning to efficiently read a resource is critical to avoid spending too much time and effort on reading all the details. A good and simple approach is the "three-pass" approach described in . The first pass gives a general idea of the paper. It is a quick overview where titles, abstract, sub-headings, theoretical foundations, and conclusion are given the attention. From this information, it should be easy to decide if more attention towards the resource in question is needed. If one does decide to take another look at the paper, the second pass is the next step. The paper should be read more carefully this time, but details such as proofs should still be ignored. Diagrams, graphs, and figures can be studied a little closer as well, and after this second pass, one should be able to summarize the main thrust of the resource. The third pass is there to fully understand the content of the resource, in depth. Most time would be spent on this part as this require attention towards all aspects of the resource. Getting the essence of a paper requires training, and this approach was the main method used when searching for related work and choosing what to spend time on, and what not to.

2.2 Development of the malware

The development of the malware, which is given the name "Exploding Kitten – an educational RAT" involved three main phase;. a planning phase, implementation phase, and a testing phase. The planning phase involves technical decision making. planning of desired design and learning objectives. However, as an agile approach is taken to the software implementation phase, there might be changes in the design and what is being prioritized. The implementation phase includes main develop steps, and a short description on internal testing. The testing phase includes a description on how the test was planned to be performed and what happened prior to infection, what happened after infection, and roles involved.

2.2.1 Planning phase

This master thesis aims to provide a customized piece of malware with certain functionalities that cover some of the learning objectives desired in an educational setting for malware analysis. In figure 3, a high-level diagram representing the design of a RAT is illustrated, which is the desired outcome of the development phase. To achieve this goal, it is beneficial to do thorough research on previous work and to carry out careful planning.

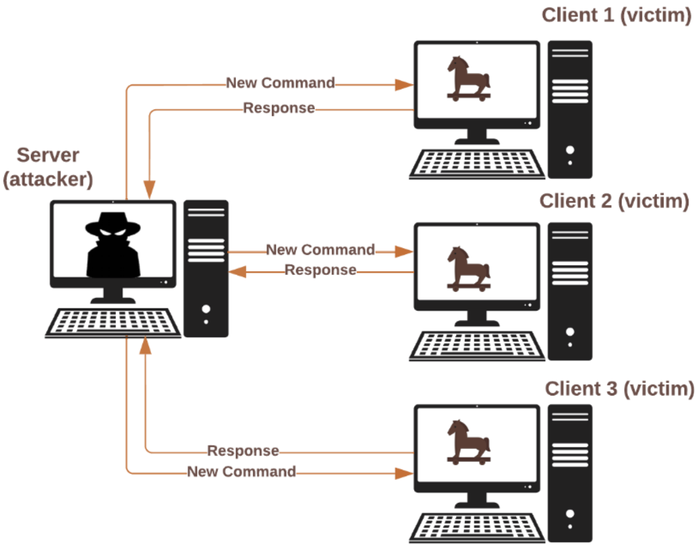


Figure 3: High-level design of a RAT

2.2.1.1 Choosing what type of malware to develop

The malicious software to be developed is a remote access trojan (RAT). This is very similar to legitimate remote access programs, but in a case of a malicious RAT, this is installed without permission. This is a most known method as it is often more difficult to discover and can go unnoticed for a long period of time. It does also allow for a number of different actions taken on the infected device such as further download of malicious payloads, stealing information, record keys strokes, take screenshots, real time movement of the cursor, privilege escalation, and installation of a fake AV.

2.2.1.2 Priority list of functions

To make sure that desired learning outcomes are met, a priority list in table 3 of wanted functionality has been made. The list is supposed to help identify the most important functions that is wanted to be included into this project, as well as functions that can be added if time. The functions presented are well known to be used in malware and are common to see in different type of trojans. A reverse shell where there is possible for a sort of command execution are the most essential function in a RAT, followed by different types of information collection functionalities. A RAT will also be in the need of a backdoor in case of the infected machine are rebooted, to make sure that connection is regained. Being able to handle multiple clients is also a realistic scenario, in case of success in the distribution phase of the malware. Scanning IP addresses allows for further investigation of the infected network and is a desired functionality in case of reconnaissance and lateral movement. Implementation of a mutex is a technique used to avoid reinfection and can be found in multiple different kinds of malware.

Table 3: Functional priority list

Priority	Function
1	Reverse shell (command execution)
2	download
3	upload
4	keylogger
5	persistence
6	obfuscation
7	screenshot
8	handling multiple clients
9	scanning
10	mutex

2.2.1.3 Learning objectives

Desired functionalities linked to learning objectives are presented in table 4. These learning objectives are functional specific.

Table 4: Learning objectives linked to functionalities

Objectives	Covered by what functionality
Students should be able to detect network traffic	reverse shell, upload, download
Students should be able to learn ways for malware to do reconnaissance	scan
Students should be able to learn ways for malware to stay persistent	persistence (registry modification)
Students should be able to learn ways for malware to avoid AV (MS defender)	reverse shell (use of excluded folders in MS defender)
Students should be able to use OSINT to determine which external tool that has been used.	download, reverse shell

In table 5, additional learning objectives that are directed towards how the students work, tools used, ability to reflect etc are listed.

Table 5: Learning objectives linked up to team and individual learning

Objective
Students are able to work on and improve their teamwork quality by getting a case that was possible to analyze rather quick and easily, which also included all members regardless their level of skills
Students are capable of using analysis tools independently
Students are able to make some thoughts about further possibilities/threats besides commands already ran.
Students are able to determine whether or not the malware looked targeted
Students should be able to determine by captured traffic what kind of actions that has been taken

Students should be able to recognize malware used based on it characteristics (typically functionalities of a RAT)

2.2.1.4 Choosing programming language

Python was the programming language used for the development of "Exploding Kitten". Python is an interpreted language, whose implementations execute instructions directly without previously being compiled into machine-language instructions. This gives the freedom to easily modify and implement code, and to test and debug it in runtime as things moves forward. Python is also a well-established program language with high popularity rate. It is a commonly used language, meaning it is commonly used among cybercriminals as well. It is easy to read, easy to write and has an extensive-third party library of scripts available.

2.2.1.5 Technical specifications of development environment

Technical specifications used in the implementation phase (see section 2.2.1.6) are listed in table x. Python was the programming language used and PyCharm the development environment, which is a python specific tool. AzureDevops was used to create a repository which was maintained in SourceTree, a GUI (graphical user interface) used for the source control.

Table 6: Development environment and technical specifications

Development environment	Software	Version
Programming language	Python	v 3.8.1
IDE (Integrated Development Environment)	PyCharm	v 2021.3.2
Git Repository	AzureDevops	v Dev18.M202.1
Git GUI	SourceTree	v 3.4.7
Platform	Windows	v 11

2.2.1.6 Functional scalability and learning level

The RAT should support the opportunity to add new functions. For some of these functions, it should also be possible to provide support in levels of difficultness, in term of analysis.

Scalability is essential within software development and needs to be considered at an early stage of the development phase. At what level will which function work together, and how easy is it to provide maintenance to these functions needs to be though through. Is it easy to make changes, or does it require fundamental changes in the infrastructure. Planned scalability leads to lower maintenance cost, better user experience and higher agility. For Exploding Kitten, it was desired to add new functions without having to make changes in the way the server and client side communicated.

Another idea was to make it possible to add different difficulty levels on some of the functions. This idea was proposed in order to highlight the educational aspects of the malware. Also, having this opportunity makes it possible to use the software in different educational settings as well. For example, it can be used on beginners, but also on a class with a higher level of skills. Looking at the functions added in the priority list, there was two functions believed to be a good fit for adding additional levels of difficulty to. Obfuscation and persistence. However, only one was looked into due to amount of time.

Obfuscation can be added on multiple levels. For this RAT, one option would be to add it to the communication between the server and client. This would give the students the opportunity to capture obfuscated traffic, and to have something to work with while the malware is still operational and conducting different actions on the infected device. Another option is to obfuscate the entire source code. This would give the students something additional to work on while analysing the extracted executable.

2.2.2 RAT Implementation

2.2.2.1 Starting point

The RAT "ExplodingKitten" was built upon an existing RAT called "BasicRAT" [28], which was found on GitHub. The source code from this RAT was cloned and used as a base for further development. Considering that this program was a mean and not the main objective for this thesis, it is reasonable not to reinventing the wheel, which is also a common practise within software development. With that said, the base was somewhat outdated and it required multiple fixes and updates in order to work properly and to meet desired learning objectives.

2.2.2.1 Main development steps

BasicRAT already had some desired and practical functions implemented. The code was however not up to date and modifications was required in order to reach the desired objectives. The following bullet points highlights the main action taken during the development phase.

- Cloned BasicRAT and created a GIT repository on AzureDevops
- Established implementation environment in PyCharm
- Upgraded and modified source code to run in newest version of Python
- Modified already existing commands
- Restructuring of code and files
- Added new commands
- Added obfuscation options in the communication between client and server
- Fixed bugs related to server-client connections
- Created build script
- Added command line options to set IP and port
- Testing and debugging with multiple clients

2.2.2.2 User commands

The user commands, that was implemented and which the attacker can perform on an infected client, are listed in table 7 and 8. The existing user commands presented in table 7 were modified, whereas the user commands presented in 8 was additionally implemented.

Table 7: Modified user commands

Command	Description
client	Connect to a client
clients	List connected clients
execute <command>	Execute a command on the target.
help	Show this help menu
kill	Kill the client connection

scan <IP>	Scan top 25 TCP ports on a single host
survey	Run a system survey

Table 8: Added user commands

Command	Description
download <file>	download file from client to server
upload <file>	upload file to client from server
persistence	Apply persistence mechanism
keylogger <arg>	Logging keystrokes. Arguments taken: <enable|disable>
screenshot <file>	Upload screenshot from client to server
encrypt <arg>	Encrypt traffic arguments taken: <none|base64|exploding>

2.2.2.3 Testing multiple clients

Debugging and testing are implicit parts of development and were frequently done throughout the whole implementation process. Most of the testing was performed between the server and client scripts in PyCharm.

In figure 4, testing two infected clients are illustrated. As seen in the figure, both clients calls home, letting the attacker know that they are ready to receive commands. Client 1 is chosen and connected to, a new command is sent, and a response is received. For this setup, two personal computers where used.

A personal computer acted like both the server and client 1. One testing scenario included using same principle as described above, both server and client script was launced in PyCharm. Second testing scenario included the use of executable files for both the server and client scripts. A second personal computer acted like client 2. The network connection was established through ethernet and required creation of a firewall rule in order to allow traffic on the decided port to be received.

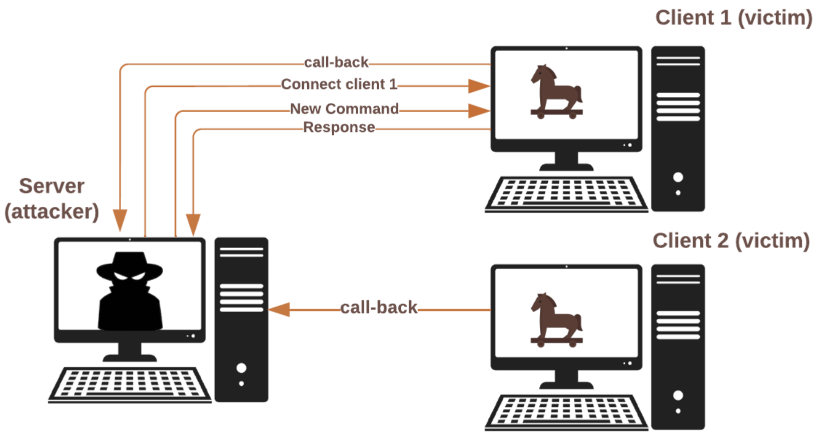


Figure 4: Illustration of multiple infected devices and communication between clients and server

2.3 Case scenario: Exploding Kitten - Testing in an educational setting

Exploding Kitten was tested in an educational setting where the case served as a basis for the interviews. The case was one out of several cases students in the Norwegian Defence Cyber Academy was presented to during an exercise.

2.3.1 Prior to the infection

Prior to the infection, an entry point for the malware, a case scenario, and indicators of compromise (IOC's) had to be decided. Deciding the entry point and how the malware managed to be installed on the infected devices was out of scope for this thesis but was still an important aspect of the learning objectives. For this part, it was decided to go forward with a combination of brute force and phishing attack. This was managed by the main educator in the Academy, alongside with all other aspects that might be relevant to a case scenario. Documentation of the case scenario "Exploding Kitten" was delivered for the supervisor of the students in case they needed any suggestions and clues. This can be found in appendix A. Furthermore, a plan of actions was written before the attack happened. This can be found in appendix B. The plan was tested in advance on an identical network to ensure everything would work properly. This has the advantage of reducing the chance of error during the case. The attack was planned to start after midnight, challenging their focus during night-time.

2.3.1.1 Scenario

A group of students is going to do an exercise within malware analysis where the developed malware will be used. This is a part of a course offered by the Norwegian Defence Cyber Academy where the students are presented with several cases involving some kind of cyber-attack. There are a lot of tasks included in this exercise that will not be discussed in this research project. The focus is on whether or not a custom-designed malware is fit for a learning environment and how the students, as well as the lecturer, experienced the malware, and not on the methods used for analysing and in general how the exercise is conducted.

The scenario presented to the students involving the developed malware represents that non-targeted attacks do happen, and sometimes cybercriminal, regardless the level of sophistication, get lucky. The scenario is based on randomness and is there to make some noise that will easily be detected and analysed. Further description of the scenario, IOC's, actions taken, and hopefully learning outcomes can be found in appendix A.

2.3.2 Roles

Two roles were involved in the case of Exploding Kitten: the attacker and the victim. Both roles were played by me, the author of this thesis. The tasks expected to be performed are listed in table 9 (there are no relation between the actions taken by the attacker and the victim).

Table 9: Roles and tasks included in the scenario

Tasks	Attacker	Victim
1	Plan a list of actions (commands)	Conduct normal user activity (allow for infection)
2	Execute the actions (commands)	Answer questions regarding what actions was performed during the night of the attack, asked by students investigation the incident.

2.3.3 Exploding Kitten has successfully infected a device

Upon infection, the activity started immediately. The attack was meant to be random and not very specifically targeted. Actions taken needed somehow to align with this scenario and it was decided to do a lot of activity right ahead before going silent for some hours and then makes some noise again. This pattern of activity was meant to demonstrate an attacker, that while still having established connection, tried to gain as much information as fast as possible. The next steps could be planned based on gained information, and students would have something to work with right away. Giving life signs hours after was meant to signalise that the malware is still present and further actions may occur.

2.4 Qualitative study

A qualitative study with the use of interview as method was used to collect data regarding the defined high-level research question. Qualitative research will be used due to the fact that the defined research questions relays on human experience, emotions, and opinions, whereas quantitative research involves "number-based" approaches and methods for collection and analysis which results in conclusions based on numbers and not feelings [29]. It is also important to notice that a qualitative study often gives an interpretive evaluation and understanding of the data. However, the purpose of a qualitative study is not to make any claims through interpretations and analysis, but to simply invite a participant to share in a lived experience. Trustworthiness is used in qualitative research as a term equivalent for internal validation, external validation, reliability, and objectivity [30], and can be achieved through for example credibility, transferability, dependability, and confirmability. To increase the trustworthiness in a study, standards such as member-checking and audit trails can be applied, each of which can be used in verifying the substance of what participants said, which in turn also reveals the degree of the interpretations done [31].

The study will take a phenomenological approach which is a form for a qualitative study where the focus is on "lived experiences". The approach gives data based on the experience of others, both in terms of what and how it was experienced. In other words, the approach can be defined as research that seeks to describe the essence of a phenomenon by exploring it from the perspective of those who have experienced it [32]. For increasing the trustworthiness of the study, member checking will be used as it gives the participants the possibility to give feedback on the accurateness of the analysed data. Also, an audit tail will be provided in order to give insight in decision making and which steps have been taken from start to findings and results. To represent the decision making, a table of how the themes were chosen can be found in appendix C. This also helps in establishing confirmability and that the findings are in fact based on response from participants and not how the researcher interpreted it [33].

2.4.1 Interview

The interview takes a semi-structured format and is the main method used in this study. Utilizing a semi-structured questioning format allows for both flexibility and more responsive interviews and is often preferred as it encourages both the participants and the interviewer to let the dialogue flow, making the setting feel more natural even though it is somehow guided and supplemented by follow-up questions. The data collected is open-ended, meaning feelings, thoughts, and beliefs are possible to explore further. As the questions are open-ended, the interview will be recorded in order to allow for better quality when transcribing [34].

The targeted focus group are students who study within the field of information security and have some or no knowledge about malware analysis. In other words, the targeted group are beginners. It is nevertheless important to include the main instructor and supervisors of the exercise as these persons has a saying in whether the software developed is fit for the desired learning outcomes or not. Because of this, even though the students are the main users, there will be a second focus group which will be the educators. There was no requirements regarding domestics, such as sex, age, etc. Regarding numbers of participants, it was ideally to interview a minimum of two students and one lecturer, and a maximum of four students and four educators. This number was due to the scope of the research project. An interview guide can be found in appendix D.

2.4.2 Analysis of collected data

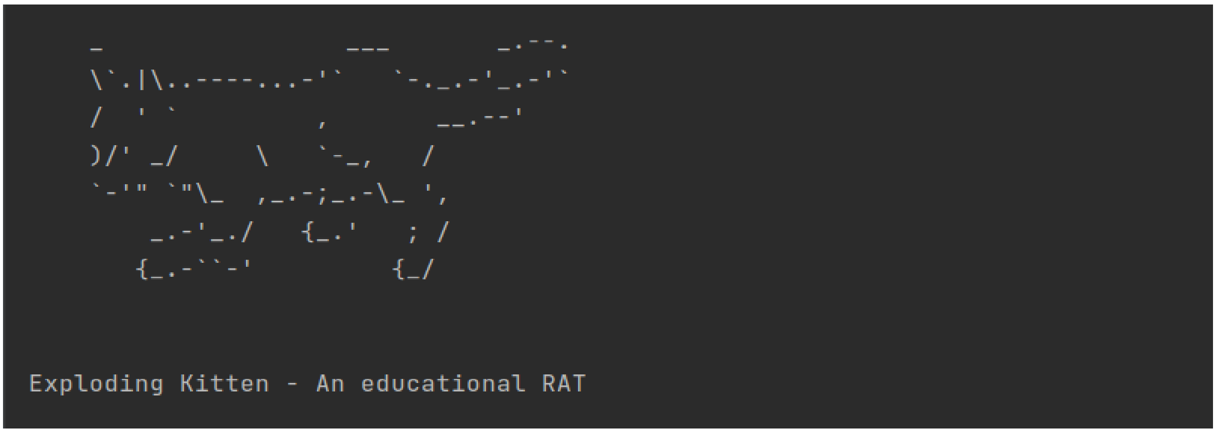
Thematic analysis will be used as method for analysing the data in order to identify patterns of themes across the data (transcripts). When analysing, normally a review of the collected data (familiarization) is the first step, then codes are derived from the data, which again are derived into sub-themes and reviewed. The sub-themes makes the main themes and each is named and presented in the analysis. The themes within data collected will be identified using both inductive and deductive approach. Inductive approach will give themes linked to data themselves and is a data-driven approach. It is also referred to a bottom-up approach. Deductive approach is the opposite. It is a top down approach with a set of predetermined codes and then find excerpts that fit those codes. Semantic approach will be used for identifying at what level the themes are derived from as this involves a progression from description where patterns is organized in semantic content and summarized to interpretation [35].

Chapter 3

3 Results

3.1 Exploding Kitten

The development phase of this thesis resulted in the malware "Exploding Kitten – an educational RAT". Exploding Kitten is a reverse shell that encompasses the ability to send defined commands to an infected client. It is scalable and introduces different difficulty levels to one of the implemented functions that can be used depending on the targeted audience in terms of analysis. Having a function that behaves according to given argument, in this case level of difficulty in terms of analysis, was implemented having learning objectives in mind and thus increasing its status as an educational RAT. Exploding Kitten is a simple RAT targeting beginners and does not include any solution for attack vector in terms of infection. A screenshot from the server terminal view is shown in figure 5.



```
    _                ___       _.--.
    \`.|\..----...-'`   `-._.-'_.-'`
    /  ' `         ,       __.--'
    )/' _/     \   `-_,   /
    `-'" `"\_  ,_.-;_.-\_ ',
        _.-'_./   {_.'   ; /
       {_.-``-'         {_/


Exploding Kitten - An educational RAT
```

Figure 5: Terminal view seen by the attacker (server side)

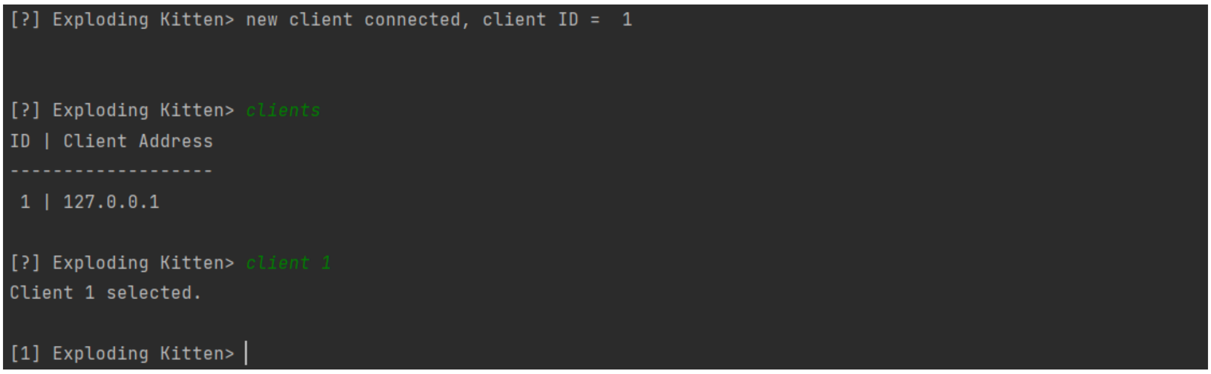
3.1.1 Functionalities of Exploding Kitten

Exploding Kitten have the possibility of conduction the following actions:

- Connect to a client
- List connected clients
- Execute a command on the target.
- Show help menu consisting of commands that can be used
- Kill the client connection
- Scan top 25 TCP ports on a single host
- Run a system survey
- Upload file from client to server
- Download file to client from server
- Apply persistence mechanism
- Enabling and disabling logging of keystrokes.
- Taking and uploading screenshot from client to server
- Obfuscate traffic choosing one of the three options available

3.1.2 Using Exploding Kitten

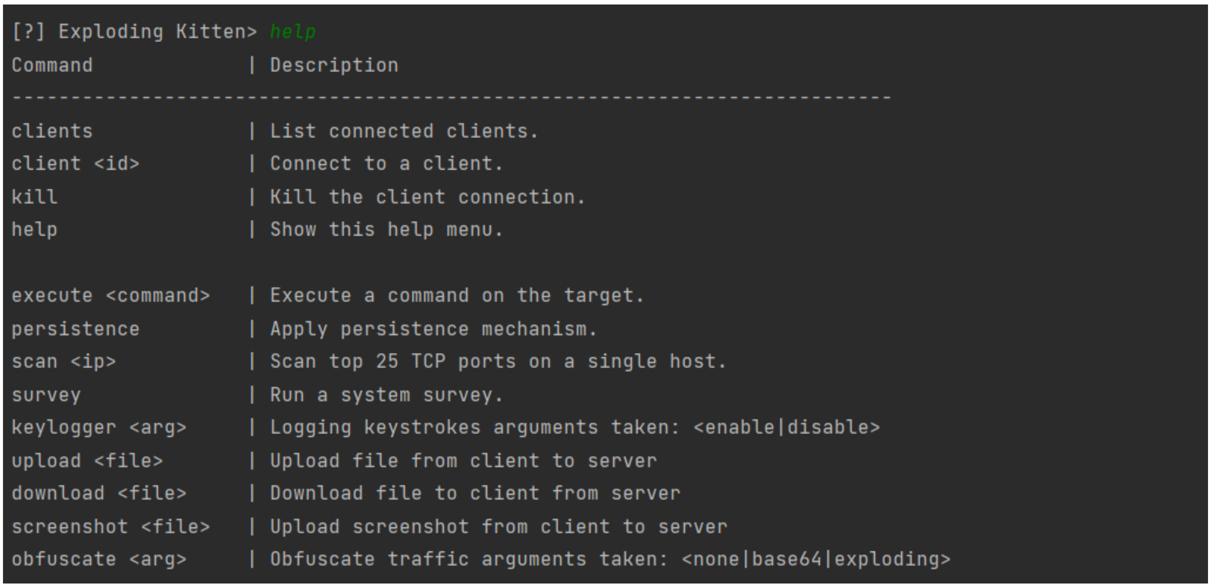
Exploding Kitten consists of an executable server and client file. The port and IP address are by default defined in the script before the executable is built. Though, it can be configurated to communicate to desired destinations when executing the executable client file as well. Once the executable files runs, the client makes contact to the server who is listening for incoming connections. The actions listed in the section above (3.1.1) apply for the attacker (server side) and further actions are described from the attacker's perspective (server side). By typing in the terminal "clients", a list of connections appear. After choosing one of the clients, for example by typing "client 1", the following commands used will be towards that client. A screenshot of the described approach is showed in figure 6.



```
[?] Exploding Kitten> new client connected, client ID =  1


[?] Exploding Kitten> clients
ID | Client Address
-------------------
 1 | 127.0.0.1

[?] Exploding Kitten> client 1
Client 1 selected.

[1] Exploding Kitten> |
```

Figure 6: Terminal view of client connection, and made connection to a client (server side)

By using the "help" command, a menu of further actions that can be taken by the attacker appears. How this appears for the attacker can be seen in figure 7.



```
[?] Exploding Kitten> help
Command             | Description
---------------------------------------------------------------------------
clients             | List connected clients.
client <id>         | Connect to a client.
kill                | Kill the client connection.
help                | Show this help menu.

execute <command>   | Execute a command on the target.
persistence         | Apply persistence mechanism.
scan <ip>           | Scan top 25 TCP ports on a single host.
survey              | Run a system survey.
keylogger <arg>     | Logging keystrokes arguments taken: <enable|disable>
upload <file>       | Upload file from client to server
download <file>     | Download file to client from server
screenshot <file>   | Upload screenshot from client to server
obfuscate <arg>     | Obfuscate traffic arguments taken: <none|base64|exploding>
```

Figure 7: Terminal view that shows "help" command being executed what it does (server side)

3.1.3 Structure of Exploding Kitten

Exploding Kitten has a server side and a client side. The clients connects to the server, and the attacker can use the server to execute commands on the client.

3.1.3.1 Server side

The server side is where new client connections are handled, as it listens for incoming connections. Furthermore, the server handle user commands. There are both user commands that concerns existing client connections and commands towards that client. Figure 8, represents the scripts involved on the server side and the functions that can be found in those scripts. As seen in the server script, the client connections are handled here. Client commands needs to be handled on both sides and are on the server side defined in the client_commands script. Each function (command) handles how the command is sent and received. The obfuscate script concerns how the message is sent and is therefore put into an isolated script that exists on both sides.

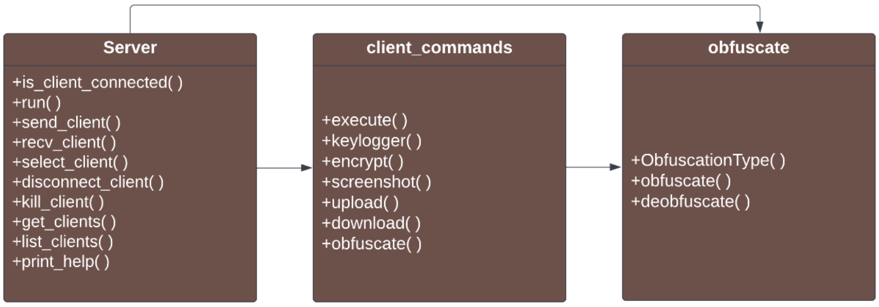


Figure 8: Server side scripts

3.1.3.2 Client side

Figure 9, represents the scripts involved on the client side and the functions that can be found in those scripts. The main function in the script client is where the attempt to connect the Exploding Kitten server is made. Also, the clients side handles received commands in the client_loop function found in the client script. Each of the commands are further defined in each of the additional scripts.

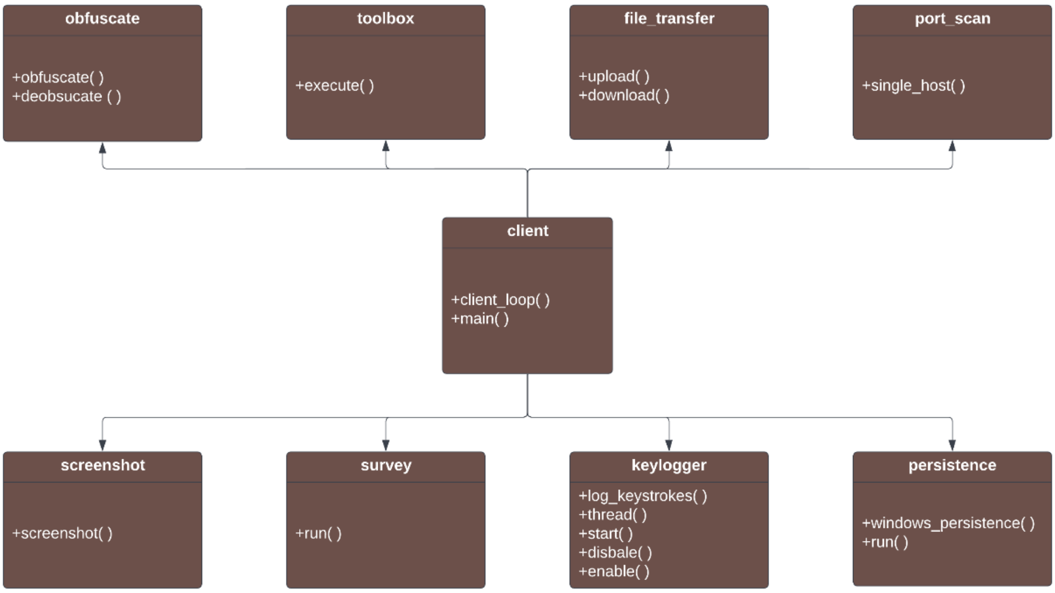


Figure 9: Client side scripts

3.2 Qualitative interviews

The collected data and the results found are presented in two main themes. The main themes encompass distinct, but closely related sub-themes describing the participants' views on the case Exploding Kitten and its benefits, and educationally friendly malware in general. The main themes found are illustrated in figure 10. Results identified in the themes are given a proper context and are summarized shortly before being supported with phrases or sentences said by participants.

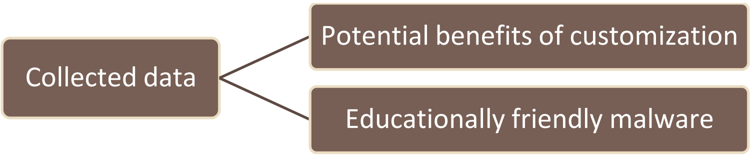

Collected data
Potential benefits of customization
Educationally friendly malware

Figure 10: Main themes found from the collected data

In table 10, an overview of the roles of the participants, if the interview took place online or onsite, time it took and date are presented. The participants were a total of five persons, of which two were students, one professor, one main instructor of the exercise (referred to educator in the table), and one which had the role as supervisor for the students during the exercise. All interviews took place onsite the following day of which this case was presented and the students were both participating on the "Exploding Kitten" case, as goes for the supervisor.

Table 10: Interview objects, place, and time/date

Subject Role	Onsite/Online interview	Time	Date
Student	Onsite	19:18	05.04.21
Student	Onsite	30:46	05.04.21
Professor	Onsite	16:00	05.04.21
Educator	Onsite	34:38	05.04.21
Supervisor	Onsite	24:54	05.04.21

3.2.1 Potential benefits of customization

Looking into potential benefits with the use of customized malware in an educational setting, and how a such program as Exploding Kitten can contribute is the main goal to investigate in this thesis. Main findings show that there are multiple beneficial and positive aspects of using a such malware. It allows for illustrating different scenarios based on the desired learning outcome, it can be used to boost both individual and team mastery feeling, and help improving skills in terms of effective response and case management. Figure 11, illustrates the main theme and encompassed sub-themes.

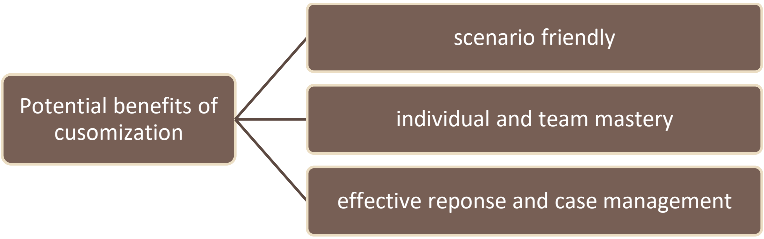

Potential benefits of cusomization
scenario friendly
individual and team mastery
effective reponse and case management

Figure 11: Theme 1 – Potential benefits of customization

3.2.1.1 Scenario friendly

Exploding Kitten did not provide a solution or suggestion for an attack vector, how it is intended to infect and potentially spread, and can therefore be used in various scenarios depending on which threat actor is wanted to be mimicked. The malware is though limited to represent less sophisticated and serious threat actors reasoning its lack of stealthiness and complexity. The scenario used in the exercise where Exploding Kitten was presented simulated a generic non-targeting phishing attack with some grade of credible content (all depending on how you define credible content). It was a scenario where user error was supposed to be the cause of successful infection.

From the point of view of the educator and supervisor participant, this is beneficial in terms of providing cases presenting different threat actors.

"Focusing on seeing the whole picture in regard to possible threats is important to illustrate. Mimicking various threat actors is for that reason important, so I think we managed that part quite well. Although we are most worried of the most sophisticated, it turns out that the less sophisticated threat actors can do harm as well due to user errors. So basic security measures must be in place".

"It is wanted that they experience the whole spectre of different possible threat actors that they may encounter in real life. Both easy and difficult to analyse. Which I believe they get through this exercise and by included a case such as Exploding Kitten".

The scenario could also present a form of insider threat or further spread itself through targets emails. As it was played for the students, there was a lot of activity in the beginning where the threat actor looked around and gather the information it could find before going silent for some time. Then there was a small amount of activity before the scenario ended from the attacker side. This could however also be adapted based on the wanted scenario to be presented and towards desired learning objectives. Possible further actions was also one of the aspects that one of the participants (student) reflected upon.

"One of the things I thought of right before my shift ended was the possibility of the RAT to move lateral in the network and attempting to infect for example the file server"

However, it was appreciated the way the scenario was presented as it did give the students a lot to work with after a short time period.

“The case (Exploding Kitten) itself was well planned. Was few quiet moments and plenty to work with throughout the whole shift. We were some afraid that it would be little activity as it was night-time but it only took half an hour before the first alarms appeared”.

“It was alright to get something to work with right away. Normally we have to wait for the forensics to make any conclusions on what we are dealing with”.

In regard to scenarios, it is also worth mentioning that while this case still was running, a case representing a more sophisticated threat actor started in the background.

"I think it was really fine just the way it was presented. The last case has been put in motion now, where they intend to stay rather discreet the first 24h, so a case that makes a little more noise while a stealthier one starts in the background will allow us to see how perceptive they are. How much will they pay attention towards what already is there and how much attention is paid on activity that is in the background".

An easier and noisier case that is ongoing while a more advanced case is starting may challenge their observation abilities. To play it differently, both scenarios could be executed by the same threat actor, where Exploding Kitten was a diversion in order to take away the student's focus while executing the real threat in the background.

3.2.1.2 Individual and team mastery

Exploding Kitten is made for being noisy, easy to analyse and to give a quick idea on what type of malware it is and some of its key functionalities. It is made for being simple and to include all students regardless of their skill-set. As the results showed in the previous section, introducing cases with different levels is a part of learning and providing a simple case cannot simply be too easy as stated by one of the participants (professor).

"Such an example can hardly be too simple. All students are to be included and everyone understands something that can be used as a foundation for further learning".

All students will also get the basics of how a RAT works in principle, as stated by one of the educator participants .

"I believe the students, all of them, will get a basic understanding of how a remote access tool works in principle, and the opportunities such a tool provides".

Besides including all students regardless of their skill level, being easy to analyse does also have the advantage of possible boosting both individual and team mastery feeling. A straightforward case may give a masterly feeling in terms of doing an overall efficient analysis, communicating well, and managing individual tasks without the need of clues in where to look and what to look for. When asking one of the participants (supervisor) about initial thoughts on the case Exploding Kitten, it was believed that a case like this, a win, was something they needed after having a more difficult one presented beforehand. It was also clear that it gave motivation and boosted their mastery feeling.

"What I got out of it, by observing, was that they really needed a case like that right now. They had a lot of difficulties with the previous cases. They met multiple obstacles, it was little clues in finding out what had happened and they needed a lot of help. Thus, getting a such attack gives a much needed motivational boost. Found all tracks themselves, easy to analyse, concluding in what happened, and considering consequences".

Both student participants also stated that even though they felt the case Exploding Kitten was too easy and they would expect something more, it was motivating having something easy and it is believed that having the whole group in mind that it was a good idea to give them something that they could immediately work with without too much thought into detecting, analysing, and figuring out what is actually going on.

"There was a lot of motivation in it, absolutely. Even though a lot of it was easy to understand. Maybe a little too easy compared to previous cases.

"But again, if there are no progress for a very long period of time it is easy to lose motivation. This was not the case for tonight (Exploding Kitten) and we had a lot of progress between 02 and 07 which was fun".

"I would like to highlight the teamwork. It was interesting to look at the way the communication worked compared to the days before".

Additionally, when asking about if it would be more interesting to get something involving encryption both stated that it would be interesting from theirs perspective, but on the other side it was also alright with something straight forward.

"It's some difficult to say now, after the event, as it was expected to get something more difficult in terms of encryption or detecting it. But that not being the case did actually give a lot of self-confidence boost. The whole process also got more efficient".

"I think it would be fun with a bit more challenge in terms of obfuscation in this case. Those cases is something I have not practises a lot on, it would be more demanding, but also more fun. The traffic could at least been in base64 instead of clear text, as you mentioned was possible. Thinking about it now, I think I would have preferred that, but at the same time it was alright with something easy and to get the tasks done quickly. Also having the rest of the team in mind, it was alright how it was done".

3.2.1.3 Effective response and case management

Another aspect of being easy and simple, is that it also allows for reviewing of effective response and case management. By providing a simple case that does not require little to no guidance, it is possible to look closer into how communication works, how individual task are performed, use of tool independently, and over all how the case is managed. Instead of using all energy on the difficult aspects of a case, the team can take a step back and practise their effectiveness and case management.

From observations made by one of the participants (supervisor), the students was effective while analysing and the report that was handed in was the most precisely yet.

"I think they were very effective with this case. Of course, the network traffic was in clear text, but still they managed to work efficient with following IOC's, analyse the information, and also they worked very effective internally and communicated well. As you experienced as well, as you played the role as victim, they called you the very same morning in order to ask you questions about the incident"

"Also, it has been the most precise report handed in yet".

One of the student participants also mentioned that it was nice to practice the process.

"It was nice to practice the process. Compared to the previous case where ransomware was run, there was a lot of difficulties and missed clues and it took quite a long time before we even understood where we should have been looking".

4.2.3 Educationally friendly malware

Looking into what educators within the field of malware analysis define as educationally friendly malware and why was relevant in terms of malware development. The results show that differences within learning environments and learning objectives affects what determines how educational-friendly malware is considered to look like. The results also indicate that it is not necessarily matter much to the students whether the malware has been developed to be used for learning or if it is a malware sample taken from the internet if it resembles a real-world threat scenario. The main theme and its sub-themes are illustrated in figure 12.

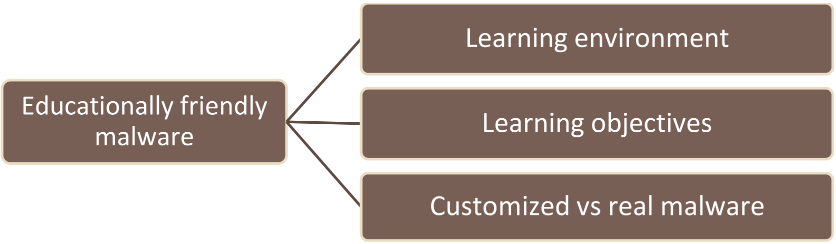


Figure 12: Theme 2 - Educationally friendly malware

3.2.3.1 Learning environment

In the cyber exercise where the case Exploding Kitten was presented in, all aspect of a possible scenario was implemented as a part of the exercise. The educational setting is a military setting and they aim at providing learning in all operations included in the cyber domain. Operating a SOC, and forensics analysis are two examples of their tasks during a such exercise. Making the exercises realistic and allowing students to be challenged with different roles are what they focuses on. Thus, what kind of malware used is not of much importance according to the educator participant.

"No, there are no limits in use of malware type. We are open for use of most types. We consider the entire threat landscape and it also important for us to look at the context and scenario when presenting cases".

However, they do want to enlighten threats specific towards the cyber domain related to military operation such as threats against intelligence.

"A very real threat to the military domain is a threat towards intelligence, so that is something we want to highlight during the cyber exercises".

As a part of the learning environment and simulated scenarios, there are also a red team that adapts their actions based on situation updates and actions done by the students, and execute further actions aligned with those updates.

"We do have a red team playing, real people that adapts actions based on the current situation, and the progress of that situation".

From the professor participant point of view, there are on the other hand some types of malware that is not desired for the students to analyse.

"It is not desired to use ransomware among others. I do not want for students to be put in a situation where they are exposed to malware that can do real and serious damage. The virtual learning environment used are supposed to be safe, but with so many students there are always some that does not follow instructions!".

"When we do find something usable, we then look into how we can neutralize it. What we do is that we take out the functionality that does the most damage. The term is called "defang".

3.2.3.2 Learning objectives

Learning objectives in a military educational setting included experiencing roles and the responsibilities that follow with those roles.

"There is no goal of becoming experts within one role with those specific responsibilities or another. This is a bachelor's degree so looking into the depth of different aspects is not of interest. Becoming experts happens when you choose what you want to specialize within".

One of the student participants also stated that by experiencing these different roles and tasks that followed with them, he/she was hoping to find out what kind of specialization that was of interest.

"We exchange tasks and responsibilities internally, so my goal was to figure out what I like to do"

In case for the professor participant, the learning objectives is aligned with the themes presented in syllabus which include basic static, basic dynamic, advanced static, and advanced dynamic malware analysis. In this case, malware analysis is also limited to those four themes and additional aspects within the field are not a part of the course.

"It is important to find malware samples that illustrate well each of the themes presented in syllabus, in order to cover desired learning objectives. Finding samples that give some sort of progression is waned, but it is not always easy to find good samples that provide this".

3.2.3.3 Customized vs live malware

When asking the students participants about usage of live malware samples and customized malware, both did not have much experience with analysing "real" malware but did however feel that the scenario presented and the malware used was realistic.

"I don't have much experience with analysing actually live malware but from what I experienced it felt really realistic. I did feel the gravity of the cases".

"I did not think it was not realistic. On the contrary, I thought that one user was probably exposed to phishing. It was also a dynamic situation update, and much more realistic than the case presented las semester".

When further asked if it would be any differences by using malware found online and malware developed for learning purposes one student participant believed that it might be as useful with homemade malware as long as it is carefully planned and aligned with those learning objectives.

"It is fine to use homemade malware if you think about what learning objectives you want to emphasize and if the malware presented are carried out according to those learning objectives".

Based on the comments from the professor participant, who does use live malware, it is a long and demanding process of finding good and educationally friendly malware samples. As mentioned above, it is desired to find samples that illustrate the learning objectives, and where it is difficult to find a good sample is especially for basic dynamic analysis.

"It is a long and demanding process. To give estimated time used is difficult to provide. I'm looking for malware to different learning purposes and settings. When trying to find malware examples that illustrates the themes in the syllabus, it might take days, weeks, or months to find the right samples. I rarely find an example that covers all themes".

"It is difficult to find a good sample illustrating basic dynamic analysis. Where we are to run live malware and see what it does to the system. Finding malware that do file and registry changes that is possible to analyse, and that leaves network traffic traces is not always easy. Especially the network communication is not easy to illustrate as we normally only have one side of the traffic, the client side, and not the server side. However, with the malware you have provided it is possible to illustrate the response wanted"

It is thus preferrable to have a self-developed program that provides full control over the source code as an introduction to the students.

"The principles we educate are valid for years, so to have something self-made that we have full control on is preferred instead of searching for live malware online that have the "correct" functions".

Progress is however important and to have a combination of something that shows basic principles and then provide the students with live malware found online would illustrate the differences and the progress desired.

"As we for now do not have something self-made, we jump right into it and uses real malware samples to illustrate the learning objectives which may make it difficult to find malware that illustrates the wanted progress".

Another challenge with real malware samples is that it may be too complex for students to understand.

"While real malware can provide students with some learning outcome, it can also often be too complex for them to fully comprehend".

Chapter 4

4 Discussion

4.1 Issue investigated and main results

This thesis explored the use of self-made malware in an educational setting and beneficial aspects with the approach. As discussed in the problem investigated (section 1.2), due to the vast number of malware it might be both time consuming and challenging to find samples that are not too complex nor too dangerous for students new to the field to investigate and analyse, and that overall mirror desired learning outcomes. The intention of this thesis was to provide with a customized piece of malware that covers some of those desired learning objectives. However, the ethics of programming malware can be discussed. Nevertheless, as stated in the sub-chapter ethical and legal considerations (section 1.9), the aim is not to discover and present new malware techniques that can be abused by cyber criminals, nor to develop zero-days attacks. Instead, already common functionalities and techniques was implemented to illustrate certain learning points. Moreover, educationally friendly aspects were the main focus during development. Thus, introducing the idea of having difficulty levels in terms of analysis that illustrates common techniques used by malware was the main contribution in regard to developing educationally friendly malware.

Results indicate that the use of customized malware is overall beneficial as it allows for flexibility and can be designed directly towards learning objectives. In this case, the malware represented a less sophisticated threat actor, made a lot of noise, and was easy to analyse. From the student participants point of view, the RAT was easy and straight forward to analyse. Having a such case gave a motivational boost and it felt good being able to perform the tasks independently without further guidance. Both the educator and supervisor participant also believed that the case provided with several positive learning outcomes. Presenting something simple allowed the students to rehearse procedures, improve team and individual skills, practise communication, and enlighten that human errors occur and that smaller incidents can be expected as well. As the supervisor states, the report on the case was the most precise yet and it was clear that they needed the motivation gained from the case. However, when discussing the deliberated practice theory in in section 1.7.2.2, the author pointed out that "According to deliberated practice theory, students do not benefit from the training if the tasks are at a level that they can perform routinely or if the goal setting of competence development has not been done with sufficient accuracy to mirror the student's level of competence". Thus, it can be discussed if this applies to the students experience as they did feel that the case Exploding Kitten was almost too easy. Nevertheless, practise is important and based on the collected data it does overall suggest that it almost cannot be too easy as long as the case is well planned and are in accordance with the existing competence. Consequently, in order to advance knowledge and skills, it is necessary to present additional cases with higher complexity.

Investigating what is considered to be educationally friendly malware was another important aspect of this thesis. On this matter, the result shows that this depends mainly on two factors: learning environment and learning objectives. In a virtual environment configured and controlled by experienced personal, there was no limitations in regard to the type of malware. It was rather important that the malware reflected desired learning objectives. In a virtual environment configured and controlled by students on the other hand, it was not desired to use too complex malware with destructive capabilities, such as ransomware, in order to prevent exposing students to unnecessary threats. The same term, "educationally friendly malware" was reflected upon and investigated in resource 3 (section 1.7.2.1). However the results in this case was used as indicators to the provided framework and educational content such as learning books was also considered when looking into the term.

4.2 Conducting the research

In chapter 2, methods used in the research was presented. The methods were presented as a process in which four steps were taken in order to answer the research questions stated in accordance with the issue investigated. There was a literature review phase, development phase, testing phase, and a data collection phase.

The literature review gave necessary insight in the topic and the resources found was used both as inspiration and a way of comparing relevant work to the issues investigated and suggested approach. Each paper presented different aspects of this thesis worth investigating: Further development of already existing frameworks, common malware techniques, locating and discussion of educationally friendly malware samples, discussing pedagogical principals within the field, and ethical consideration of malware programming. However, looking closer at the state of the art within malware techniques could have been beneficial in light of the making the malware more educationally friendly and directed towards even more specific learning objectives.

In the development phase, planning the malware and learning objectives based on provided functionality was of great importance. During implementation, frequent meetings was also in place to present progress during development and getting feedback on decisions. Based on the results, the malware did overall provide with the wanted functionalities and could be adapted to different case scenarios depending on the threat actor wanted to illustrate. The educational setting the developed malware was presented in did however emphasise the importance of illustrating different threat actors as a whole. Meaning providing an attack vector was essential for the scenario presented to the students. This was something that should have been considered during the development phase.

The developed malware was tested in a military educational setting. Deciding the testing environment and all details regarding how the malware was to be investigated, including among other tools used and aspects of the infection, was out of scope for this thesis. The focus was on how self-made malware can be beneficial in an educational setting and not in a specific educational setting. In advance of the exercise it was though possible to test the final product in an identical learning environment in order to identify possible issues, testing the communication between server and client, and to learn technical details of the environment. As the case scenario consisted of a server side (attacker) and a client side (victim) and was illustrating a phishing attack, two roles was required to be played; an attacker and a victim, both in which was played by me, the author of this thesis. This was both beneficial at the same time as it may reduce the realness of the scenario from a student point of view. While performing some actions as an attacker, there was also a need of simulating common user tasks on the infected machine. The actions performed by the victim could however easily align with the actions done by the attacker. To what degree a such scenario appears realistic was also a concern shared by the supervisor participant. Thus, the importance of carefully planning does applies to this phase as well, as how well an exercise is planned reflects all aspects of it.

The data collection phase consisted of five semi-structured interviews. Three of which was educators within the field. The educators represented two different learning environments, in which two (educator and supervisor) represented the testing environment. Two participants represented the students, in which both were present during the detection and initial analysis of the developed malware. The case happed during night-time and the interviews was conducted right after the students were done with their shift, while the memory was still fresh. This can however be some criticized as it did not allow for the students to fully reflect upon the case, and it might have influenced their ability to think clearly as well. All five interviews were furthermore performed before the case was finished being fully analysed and concluded in by the students. Student participants in the research did not include the ones collecting and examining the infected device. There are for this reason no collected data regarding the reverse engineering process and further thoughts on the malware itself and techniques used. Such data could have provided additional insight into the overall experience with the case and more specifically in what degree the forensic and reverse engineering process offered learning. Although, given the response from the participants included in the case, it can be speculated in that that it was as motivating and that it provided skill improvement in terms of routinely performance, and individual and team mastery feeling. This can however not be concluded in.

4.3 Design decisions

One of the low-level research questions involved considering common functionalities and techniques found in malware. By investigating different types of malware and based on the priority list defined in section 2.2.1.2, it was decided that a remote access tool would be representative for a lot of malware that is used by cyber criminals. A RAT allows for command execution which opens up for multiple further attack options, such as download additional malware and exploitation tools, information theft, later movement in network, establishing persistence etc. Thus, the customized malware "Exploding Kitten – an educational RAT" resulted in a remote access trojan, a scenario flexible tool consisting of common functions and techniques used by malware. When investigating common techniques used, resource 2 (presented in section 1.7.1.2) which discussed malware obfuscation techniques was used as inspiration.

From a teaching perspective, a RAT allows students to detect network traffic, capture packets, and to analyse the traffic, additionally to analyse the malware itself. Having malware that generated a lot of human-readable traffic was based on the results appreciated and motivating as it generated work-load possible to analyse right away but was also at the same time not very challenging. New functions can further easily be added to a RAT, making it a scalable project that can be adjusted, if wanted and programmed in sauch a way, to fit the targeted students level of skills. If the malware to be developed was for example ransomware, it would not be any flexibility in "choosing" what kind of encryption that should be used, as this program would not have any command and control feature. The trojan will also make it possible for the students to reflect upon further actions that may be taken.

Deciding on the malware type to be developed also included looking into existing frameworks such as Metasploit and Cobalt Strike as it could be used as an inspiration. As presented in resource 1 (section 1.7.1.1), an extension to Metasploit in terms of automating some of the process was provided. This could be used as an inspiration as well in terms of automating parts of the malware that was to be developed as well. However, both frameworks are developed though many years and are highly complex. If there is any inspiration to be taken from the frameworks, it would thus be its modularity.

The malware was programmed in Python. However, C/C++ would be good options as well, as these are as common languages. They are better in terms of compression and resourced taken when running, faster, and compiled by nature. This does on the other side makes it possible more difficult to analyse as a beginner. Though, as Python is an interpreted language, it is relative easier when it comes to reverse engineering compared to compiled machine code. In terms of malware, Python does require the interpreter needed to be installed in the victims computer, or the malware can be packed into a single executable file. In the case of "Exploding Kitten", option two was used. When the malware is then located on the infected machine, the students will need to unpack the malware file and extract the different scripts inside. With the use of compiled code, the code provided from reverse engineering will, because of the way machine and assembly code works, not be exactly the same.

There are however a lot of improvements that can be applied into Exploding Kitten in order to increase its status as educationally friendly malware. Implementation of common malware techniques in different variants, each representing a different degree of complexity is one suggestion. This has already been applied to one function which allows for deciding whether to obfuscate the network traffic or not, and at what level. Obfuscation and different levels of it could also be added to the source code itself which would provide a challenge to students in terms of reverse engineering the program.

Exploding Kitten does not provide any form of an attack vector. There are numerous different strategies that can be used depending on the threat actor wanted to mimic. As with the above, including different levels of difficulty can also be included in the infection phase. The scenario that the malware represents is an important aspect of an educational setting such as the one Exploding Kitten was presented in. Thus the type of attack vector used has significance for the scenario. Providing a such example of attacking vector was however out of scope for this thesis.

However, in order to stay educational friendly there should not be a goal in it-self to develop a full-scale framework such as metaplot or cobalt strike which also as discussed are used by threat actors. Keeping the difficulty levels to a minimum should be enough to illustrate some learning points.

Chapter 5

5 Conclusion

5.1 Conclusion of the thesis

This thesis investigated the educational aspects of self-made malware and the extent to which it can be beneficial in a learning environment. The suggested malware "Exploding Kitten – an educational RAT" was developed with the intention to contribute towards educational objectives and resulted in a flexible tool that encompasses several beneficial functionalities and supports situational adaptability. To conclude in what degree and how the malware contributes with achieving learning objectives, it was placed in an educational setting and presented as one of the case scenarios in which students were to investigate. How the malware was experienced by both educators and students was thus investigated through semi-structured interviews. Furthermore, the term "educationally friendly malware" was explored in the interviews where an additional external participant with knowledge within teaching malware analysis contributed on the topic. The result of the analysis is two-parted: one in which highlights the beneficial aspects of self-made malware and one in which investigated what is put into the term educationally friendly malware.

The findings indicates that whether the malware used is considered to be educationally friendly or not, differ within learning environment and learning objectives. The case Exploding Kitten was experienced to be beneficial in terms of providing motivation and feeling of team and individual mastery, as it represented a less sophisticated threat actor. Exploding Kitten did however represent only one specific threat actor. In terms of learning, it is important to provide the opportunity for knowledge and skills progression, thus other malware samples are required to cover additional aspects of wanted learning outcomes. The collected data did further indicate that from the student's point of view, it might not be of great relevance if the malware is self-made or not as long as it represents the desired learning objectives well. On the other hand, from an educator's view, self-made malware such as Exploding Kitten allows for highlighting some specific desired learning objectives and being in total control of source code. As a result, use of customized malware as an approach in an educational setting allows for flexibility.

5.2 Future work

Exploding Kitten RAT is a first version program with a lot to offer. There is yet room for improvement, and several suggestions have been provided for in the discussion chapter. As a first version, there was no solution to attack vectors. It was instead a focus on functionality and providing levels of learning for those functionalities. Both are aspects of the RAT that can be developed further. Adding a solution for the infection phase makes the RAT more suitable for other learning environments, such as its tested environment. If the learning objectives are limited to the malware analysis itself, this would not be important. Consequently, for a learning setting where the scenario is of great relevance, including the attack vector would improve its overall quality in regard to the set learning objectives. Adding additional learning levels will strengthen its status as educational friendly as well, and it can be considered to provide learning levels to the following aspects of the RAT: Obfuscation of the malware binary, persistence, and attack vector.

In regard to further research on educationally friendly malware with the use of customized malware, it could be interesting to look at the use of the malware in a different learning environment and look into the differences in the learning experiences. Another aspect of the thesis that would be interesting to investigate further is the differences between the usage of self-developed malware and live malware found online. The aspect was briefly discussed with the student and professor participants, but apart from speculation, no further data was collected on this matter.

Bibliography

[1] AV-TEST, “Malware Statistics & Trends Report.” https://www.av-test.org/en/statistics/malware/ (accessed Dec. 10, 2021).

[2] C. Beek *et al.*, “McAfee Labs Threats Report, April 2021,” 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-apr-2021.pdf

[3] Tulane University, “Four Reasons the Cybersecurity Field Is Rapidly Growing.” https://sopa.tulane.edu/blog/four-reasons-cybersecurity-field-rapidly-growing (accessed May 11, 2022).

[4] P. Jathanna, “Top 3 Reasons Why Cybersecurity Careers are the Future,” Nov. 30, 2021. https://emeritus.org/blog/cybersecurity-careers-always-in-demand/ (accessed May 11, 2022).

[5] (ISC)^2, “(ISC)2 CYBERSECURITY WORKFORCE STUDY, 2021,” 2021. Accessed: May 10, 2022. [Online]. Available: https://www.isc2.org//-/media/ISC2/Research/2021/ISC2-Cybersecurity-Workforce-Study-2021.ashx

[6] City University of Seattle, “Demand for qualified cybersecurity workers is soaring | The Seattle Times,” Nov. 15, 2021. https://www.seattletimes.com/sponsored/demand-for-qualified-cybersecurity-workers-is-soaring/ (accessed May 10, 2022).

[7] Inc. Solutionary, “How Malware Analysis Benefits Incident Response.” Accessed: Dec. 10, 2021. [Online]. Available: https://informationsecurity.report/Resources/Whitepapers/51e831f9-aeef-41a4-b2e9-5162a2ac5f65_How%20Malware%20Analysis.pdf

[8] V. Saengphaibul, “A Brief History of The Evolution of Malware | FortiGuard Labs ,” Mar. 15, 2022. https://www.fortinet.com/blog/threat-research/evolution-of-malware (accessed May 12, 2022).

[9] M. Landesman, “A Brief History of Malware,” Mar. 09, 2021. https://www.lifewire.com/brief-history-of-malware-153616 (accessed May 12, 2022).

[10] T. Shareef, “The 8 Most Notorious Malware Attacks of All Time,” Sep. 16, 2021. https://www.makeuseof.com/most-notorious-malware-attacks-ever/ (accessed May 29, 2022).

[11] D. Panduru, “10 Malware Examples: Most Famous And Devastating Cases In History - ATTACK Simulator,” Aug. 09, 2021. https://attacksimulator.com/blog/10-famous-malware-examples-in-history/ (accessed May 29, 2022).

[12] F. Melnyczuk, “Famous Virus Attacks | Antivirus.com - Cybersecurity, Data Leaks & Scams, How-Tos and Product Reviews,” Oct. 27, 2021. https://antivirus.com/2021/10/27/famous-virus-attacks/ (accessed May 29, 2022).

[13] Norton_team, “The 8 Most Famous Computer Viruses of All Time,” Feb. 22, 2016. https://uk.norton.com/norton-blog/2016/02/the_8_most_famousco.html (accessed May 29, 2022).

[14] Rapid 7, “Metasploit.” https://www.metasploit.com/ (accessed Nov. 29, 2021).

[15] Offensice security, “About the Metasploit Meterpreter - Metasploit Unleashed.” https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/ (accessed Dec. 01, 2021).

[16] Rapid 7, “Metasploit Framework | Metasploit Documentation.” https://docs.rapid7.com/metasploit/msf-overview/ (accessed Dec. 01, 2021).

[17] Offensive security, “MSFvenom - Metasploit Unleashed.” https://www.offensive-security.com/metasploit-unleashed/msfvenom/ (accessed Dec. 01, 2021).

[18] SophosLabs, “Sophos 2022 Threat Report Interrelated threats target an interdependent world,” Nov. 21AD.

[19] Malwarebytes Labs, “Cobalt Strike, a penetration testing tool abused by criminals | Malwarebytes Labs,” Jun. 01, 2021. https://blog.malwarebytes.com/researchers-corner/2021/06/cobalt-strike-a-penetration-testing-tool-popular-among-criminals/ (accessed May 11, 2022).

[20] G. Ioannou, “Malware Development for Red Teaming Using Metasploit Petros Katritzidakis,” 2018. Accessed: Nov. 29, 2021. [Online]. Available: https://core.ac.uk/download/pdf/236205153.pdf

[21] I. You and K. Yim, “Malware obfuscation techniques: A brief survey,” *2010 International Conference on Broadband, Wireless Computing Communication and Applications, BWCCA 2010*, pp. 297–300, 2010, doi: 10.1109/BWCCA.2010.85.

[22] A. Bjørkhaug, “Finding Educationally Friendly Malware,” NTNU, Gjøvik, 2021. Accessed: Nov. 29, 2021. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2786107

[23] M. Karjalainen and T. Kokkonen, “Review of pedagogical principles of cyber security exercises,” *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 5, pp. 592–600, 2020, doi: 10.25046/AJ050572.

[24] J. P. Sullins, “A Case Study in Malware Research Ethics Education: When Teaching Bad is Good,” 2014, doi: 10.1109/SPW.2014.46.

[25] NC State University, “Ethics in Computing.” https://ethics.csc.ncsu.edu/intellectual/reverse/study.php (accessed Dec. 01, 2021).

[26] M. Sanjari, F. Bahramnezhad, F. K. Fomani, M. Shoghi, and M. Ali Cheraghi, “Ethical challenges of researchers in qualitative studies: the necessity to develop a specific guideline,” *Journal of Medical Ethics and History of Medicine*, vol. 7, p. 14, Aug. 2014, Accessed: Dec. 01, 2021. [Online]. Available: /pmc/articles/PMC4263394/

[27] J. Rowley and F. Slack, “Conducting a literature review,” *Management Research News*, vol. 27, no. 6, pp. 31–39, Jun. 2004, doi: 10.1108/01409170410784185/FULL/PDF.

[28] A. Jackson and S. Curtis, “GitHub - wisoez/RAT-Python.” Sep. 11, 2017. Accessed: May 15, 2022. [Online]. Available: https://github.com/wisoez/RAT-Python

[29] D. Booth, “LibGuides: Research Methods: What are research methods?,” Dec. 15, 2020. https://libguides.newcastle.edu.au/researchmethods/home (accessed Nov. 17, 2021).

[30] S. of E. and H. D. University of Miami, “How is reliability and validity realized in qualitative research? – STATS-U.” https://sites.education.miami.edu/statsu/2020/09/22/how-is-reliability-and-validity-realized-in-qualitative-research/ (accessed Nov. 18, 2021).

[31] R. L. Jackson, D. K. Drummond, and S. Camara, “What Is Qualitative Research?,” *https://doi.org/10.1080/17459430701617879*, vol. 8, no. 1, pp. 21–28, 2007, doi: 10.1080/17459430701617879.

[32] A. Teherani, T. Martimianakis, T. Stenfors-Hayes, A. Wadhwa, and L. Varpio, “Choosing a Qualitative Research Approach,” *Journal of Graduate Medical Education*, vol. 7, no. 4, pp. 669–670, Dec. 2015, doi: 10.4300/JGME-D-15-00414.1.

[33] Cohen D and Crabtree B, “RWJF - Qualitative Research Guidelines Project | Audit Trail | Audit Trail,” Jul. 2006. http://www.qualres.org/HomeAudi-3700.html (accessed Nov. 18, 2021).

[34] Cohen D and Crabtree B, “Qualitative Research Guidelines Project,” 2006. http://www.qualres.org/HomeSemi-3629.html (accessed Nov. 29, 2021).

[35] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006, doi: 10.1191/1478088706qp063oa.

Appendix A – Exploding Kitten Case Scenario

Case3: Exploding Kitten – an educationally friendly RAT

Scenario

A phishing campaign towards among others the .mil domain has successfully come through. The campaign is not targeted and one of the victims happens to believe the content to be valid. The source and content of the malicious mail was masquerading "Skatteetaten". As a consequence, the machine gets infected by the Exploding Kitten RAT.

- Email distributed through a spamming campaign by atl4mhh05.registeredsite[.]com
- One dummy extracts the zip file into Download folder which happens to be an excluded folder by Microsoft Defender AV
- Dummy runs the exe
- The RAT establish connection with the c2 server

IOC

C2 callback address	172.25.193[.]77
Domain	atl4mhh05.registeredsite[.]com
Excluded folder by Defender AV	C:\Downloads
Persistence path in registry	Software\Microsoft\Windows\CurrentVersion\Run
Tool used	log.exe (seatbelt)

About the RAT

Language written in	Python
Shell access	Cmd and powershell
Client functionality	Persistence, keylogger, download/upload files, network traffic obfuscation/deobfuscation, screenshot, scan ip, execute command
Server functionality	Kill connection to client

Hopefully learning outcome:

- Students was able to work on and improve their teamwork quality by getting a case that was possible to analyze rather quick and easily, which also included all members regardless their level of skills
- Students was capable of using analysis tools independently
- Students was able to capture traffic that made it possible for them to get an idea of what commands that was ran and could based on this quickly conclude in what kind of threat the user/organization was exposed to. This also includes determining the capability of the malware.
- Students used OSINT to find out what kind of tool that was downloaded (seatbelt.exe)
- Students were able to make some thoughts about further possibilities/threats besides commands already ran.
- Students was able to understand why Windows Defender did not stop execution of additional tools used (exclusion folder)

- Students was able to locate registry modifications based on command ran, and what this registry modification was used for (persistence)
- Students was able to determine whether or not the malware looked targeted

Timeline and actions

- After midnight, Exploding Kitten was infected on a single host after a successful phishing attempt
- C2 activity started right after infection
- Persistence and keylogger was enabled before the threat actor started to take a look around in the system
- Information disclosure vulnerability in Defender allowed attacker to exploit any folders, files, path, or processes left out by AV. The Downloads folder was for some reason excluded which made it possible to download further tools without being stopped by defender.
- The tool seatbelt was used to extract different kind of information
- The attacker also learned that W: was available, which was shared files.
- Different kinds of files were uploaded from client to server, this included files from shared files, logged keystrokes, screenshot taken, phone list, and logs created by seatbelt.
- To hide its tracks, extracted information stored in new files was uploaded and deleted
- Some network mapping was also done using among other the scan command and the DC was among other discovered.
- The attacker god tired and went to bed
- The morning after the attacker checked in to see if connection was still up
- The attacker then went silent while planning next steps of attack

Links

- https://borncity.com/win/2022/02/11/microsoft-fixt-wohl-heimlich-schwachstelle-im-defender-unter-windows/
- https://github.com/GhostPack/Seatbelt

Appendix B – Plan of actions

Plan of actions before infection

1. Build client executable file, change default port and IP before building
2. Prepare some documents that can be upload from client to server
3. Create a folder exclusion. From powershell as admin: Add-MpPreference -ExclusionPath "C:\...."
4. Right-click on file, go to properties and uncheck the “unblock” checkmark
5. Execute infected file from this folder

Plan of actions during infection

- Apply persistence mechanism
- Start key logger
- Gather information available
- Take a look around, see if there are any shared folders
- Any IP addresses of interest?
- Use of Seatbelt

Initial commands

survey
persistence
keylogger enable
execute dir
execute net user
execute net start
download seatbelt.exe
seatbelt.exe -group=all -full -outputfile="log.txt"
upload log.txt
screenshot
execute ipconfic /all
scan <ip>

Appendix C – Details of the interview analysis process

Initial themes:

- Educationally friendly malware
- Customized malware

Identifying themes related to malware development:

- Exploding Kitten – case presented for the students and result of the developed malware

Words, phrases, and sentences	Codes	Sub-themes	Themes
"A lot of motivation"	Motivation		
"Motivational boost"		Individual and team mastery	
	Easy		
"Easy to anlyse"			
	Noisy		Potential benefits of customization
"Generated a lot of events"			
"want to highlight team work"	collaboration		
"self-confidence boost"			
	Mastery feeling		
"the whole process got a lot more effective"			
		Effective response and case management	
	Effective team work		
"no challenge in clear text"			
"good communication"			
	Little challenge		
"prefer encryption"			
"easy to adapt scenarios"	Communication		
"feeling of mastery"			
	Easy		
"felt realistic"			
		Scenario friendly	
"including"	Mastery		
"everyone teaches something"			
"important to illustrate different threat actors"			

Identifying themes related to malware analysis courses:

- Educating malware analysis – what is considered to be educationally friendly malware

Words, phrases, and sentences	Codes	Sub-themes	Themes
“cyberdomain in a militarey setting” “time consuming task to find samples that illustrates the wanted learning objectives” “defang functions that does more harm than nessecarry” “only a bachelor, spesialization comes after” “security operation center, SOC” “it is important to give the students a form for learning progression” “do not have limitation regarding the malware it self” “threat intellegece is one of the biggest threats in a military operation” “beneficial having cases both readable and obfuscated” “Want to present malware easy to analyse as well as more complex samples not as easy to figure out” “want to highlight the whole therat ladnscape” “do not want to use malware too comples or having destructable capabilities” “virtual lab enviornment” “difficult to show how c2 traffic looks like with only having the client side”	Military enviornment Finding malware samples takes a lot of time and energy Malware covering different learning objectives SOC Levels of difficulty Malware types What type of malware used is not of importance, it is more important that certain learning objectives are highlighted, not how. Represent different learning objectives Threat landscape Virtual Real malware samples found online Malware samples found online does not cover some of the learnings objectives such as c2 traffic	Cusomized vs real malware Learning objectives Learning environment	Educationally friendly malware

Appendix D – Interview guide

(This section will be in both Norwegian and English, as the interviews will be held in Norwegian).

Før vi starter intervjuet ***//Before starting the interview***

Formålet med intervjuet er å undersøke hvorvidt ett eget designet malware egner seg for undervisning sammenlignet mot noe som har blitt brukt i virkeligheten, og hva slags lærerutbytte dette gir. Ønsker du å ta pause, har noen spørsmål eller noe annet på hjertet så bare si ifra. Hvis du ikke ønsker å svare på noen spørsmål er dette også greit, og du kan forlate intervjuet når enn du ønsker.

//The purpose of this interview is to investigate whether or not a customized malware sample is beneficial for an educational course, and to look into what learning outcome it does provide. If you want to pause, have questions or any other problems, just let me know. You are also not obligated to answer the questions if you don't want to, and free to withdraw from the interview at any time.

Spørsmål for studenter fra Cyeberingeiørskolen ***//Questions for students in the Norwegian Cyber Defence Academy***

Del 1 *//Section 1*

Generelle spørsmål *//General questions*

- Hva er bakgrunnen din innenfor cybersikkerhet feltet? Hadde du noen tidligere erfaringer som kom godt med under øvelsen? *//What is your background within the field of cybersecurity? Did you have any previous experience that beneficial during the exercise?*

- Hadde du på forhånd noen tanker/forventninger rundt øvelsen? Hva håpet du på å lære? *// What was your pre expectations/thoughts for the exercise? was there anything specific that you wanted to learn?*

- Tror du det er noen forskjell på å analysere malware laget for en øvelse enn malware som har blitt brukt i virkeligheten? *// Do you think it is any difference in studying malware developed for an exercise/course versus malware samples taken from real life?*

Del 2 *//Section 2*

Spørsmål rettet mot ExplodingKitten og om arbeidsoppgaver *//Questions about the students work tasks related to the ExplodingKitten RAT and the RAT itself*

- På hvilken måte syntes du skadevareprogrammet «ExplodingKItten – an educational RAT» ga deg læringsutbytte? Også da i forhold til de andre skadevarene dere jobbet med å analysere? *//In what ways do you think the malware «ExplodingKitten» gave you educational outcome? Also when comparing it to the other cases given during the exercise?*

- Kan du fortelle litt om hva din jobb var under denne spesifikke casen (ExplodingKitten)? Hva var dine arbeidsoppgaver og ansvarsområder? *//Can you talk about your work tasks and responsibilities during this case?*

- Hva var det som gjorde det lett/vanskelig å analysere ExplodingKitten? *//What made it easy/difficult to analyse ExplodingKitten malware?*

Del 3 *//Section 3*

//Questions about the exercise in general

- Overårdnet, hvordan syntes du øvelsen gikk? *// Overall, how do you think the exercise went?*

- Kan du si en positiv og en negativ ting om kurset? *// Name one positiv and one negative experience with the course.*

- Var det noe som kunne blitt gjort bedre? *// Was it anything that could have done the course better?*

Spørsmål for undervisere *//Questions for educator*

Del 1 *//Section 1*

- Hva ser du på som studievennlig skadevare? *//What is educational friendly malware for you?*

- Hva er deres tidligere erfaringer med slike øvelser? Kan du fortelle litt om koseptet? Hva har funket og hva har ikke funket? *//What are your previous experience with this kinds of exercises? Can you say something about the concept? What has worked and what has not worked?*

- Hvilke områder eller elementer ønsker dere å ha i focus under analysebiten? *//What areas of malware analysis or elements do you want to focus on during the exercise?*

Del 2 *//Section 2,*

- Hvordan synes du den utviklede skadevaren fungerte i praksis? *// Overall, how do you think the malware worked for this exercise?*

- Har du noen tanker rundt fordeler og ulemper med ExplodingKitten? Også sett mot de andre skadevarene dere brukte? *//Do you have any thoughs on the benefits/disadvantaes by using malware like ExplodingKitten?*

- Malware programmet er laget slik at det skal være lett å legge på ny funksjonalitet, er dette noe som du/dere fortsatt vil bruke/utvikle for å kunne bruke det igjen til senere øvelser? *Is this RAT something you would like to further use/develop? And to use in next years exercises?*

- Hva kunne blitt gjort bedre/annerledes? *//What should have been done differently?*

- På hvilke måter dekket programmet det læringsutbyttet som dere ville ha? *//In what ways did the malware cover the desired learning outcome?*

Spørsmål for ekstern underviser ***//Questions for external educator***

- Hvor mye tid (estimert) blir lagt inn i å finne skadevare som kan bli brukt for undervisning? *//How much time (estimated) do you use on finding malware fit for the course?*

- Hva ser du etter når du skal finne skadevare for kurset, og er det noe du unngår? *//What are you looking for when finding malware samples? Any types that you are avoiding?*

- Hva var ditt første inntrykk av programmet? *//What was your first impression of this program?*

- Er dette noe som kan brukes i ditt kurs? *//Is this something that can be used in your course?*

NTNU

Kunnskap for en bedre verden