AAU Honeypot

AALBORG UNIVERSITY
STUDENT REPORT

Study Board of Electronics and IT
Fredrik Bajers Vej 7 DK - 9220 Aalborg East
Phone +45 99 40 86 90
electronics@sict.aau.dk
www.sict.aau.dk/electronics-and-it

Title: AAU HoneyPot
Semester: 1st
Semester theme: Network & programming
Project period: 16/10/17 – 20/12/17
ECTS: 10
Supervisor: Jens Myrup Pedersen
Project group: P1B130

[Christopher Damsgaard]

[Daniel Britze

[Jacob Vejlin Jensen]

[Kristian Noesgaard Nielsen]

[Magnus Stensli]

[Mikkel Steen Hansen]

Abstract:
This project takes its starting point in the rising problem of cybercrime. The results from the conducted analysis of stakeholders leads to smartphone users being set as the target group of the project. To cast light on the problem a user-survey is made to determine the behaviour of a generic smartphone user. Based on the problem analysis, it is established to research how selected malware interacts with sensitive data from an emulated Android device. In order to investigate this, it is determined to construct a Honeypot system consisting of a test environment running virtual Android machines, a containment system to control the tested malware and a module able to monitor and analyse the network traffic. The study lays the foundation for a basic Honeypot system with potential for further development.

Number printed: 1
Number of pages: 86
Appendix: 7
Enclosures: 0

By signing this document, each member of the group confirms participation on equal terms in the process of writing the project. Thus, each member of the group is responsible for the all contents in the project.

Synopsis

Denne rapport tager udgangspunkt I de stigende problematikker angående cyber-kriminalitet. Resultaterne fra den udførte interessantanalyse leder til, at brugere af smartphones vælges til værende fokuspunkt for projektet. For at belyse problemstillingen, udføres der en brugerundersøgelse. Spørgsmålene i undersøgelsen er rettet mod at fastlægge brugsmønstre for generelle smartphone brugere. Baseret på problemanalysen, besluttes det at undersøge, hvordan udvalgt malware interagerer med følsom data på en emuleret Android maskine. For at efterforske dette, besluttes det at konstruere et ”Honeypot” system bestående af: Et testmiljø hvori der køres virtuelle Android maskiner, Et ”containment” system til at kontrollere internettrafikken, samt et modul med evnen til at monitorere og analysere denne trafik. Projektet bygger fundamentet til et basalt ”Honeypot” system med potentiale for fremtidig udvikling.

Table of Contents

List of figures

1. Introduction

Cybersecurity is a complex field in growth, acting as the counterpart to the still rising cyber-crime.(1) In our global world, we become more comfortable with and dependant on, storing data on smartphones, computers and cloud-services.

As an example, everything from holiday pictures and private messages to banking information, credit card numbers and personal numbers (Social security numbers, Passport numbers, etc.) are stored by regular smartphone users.[1] The data is stored on local servers, cloud services, and on the local device's memory.

Furthermore, companies of all sizes, organizations and governmental institutions, depend on important data stored on their internal networks and local/user devices.(54) As an example, hospitals depend on data stored in their systems to diagnose patients, admit and discharge new patients and to operate medical machines and data analysis equipment.

Access to the Internet makes communication and outside interaction possible. As a result, large amounts of sensitive data are at risk of being hacked by criminals with the ability to bypass the security barriers. As we increase the amount of sensitive data that we store digitally, the criminal counterpart develops equally fast.

New attacks occur every second of every day, with the intent of stealing, damaging or encrypting data and converting the gained information into currency.(55)

The consequences of a cyberattack can be fatal if carried out with expertise and precision. An example of a recent cyberattack resulting in financial damages on a large scale is the "*NotPetya"* attack. It affected several governmental institutions as well as about 140 companies worldwide in July 2017.

The malware exploited a security hole in the Windows operating system, that allowed it to enter without root access and then spread through the companies' network of computers.

[1] Section 4.2.2

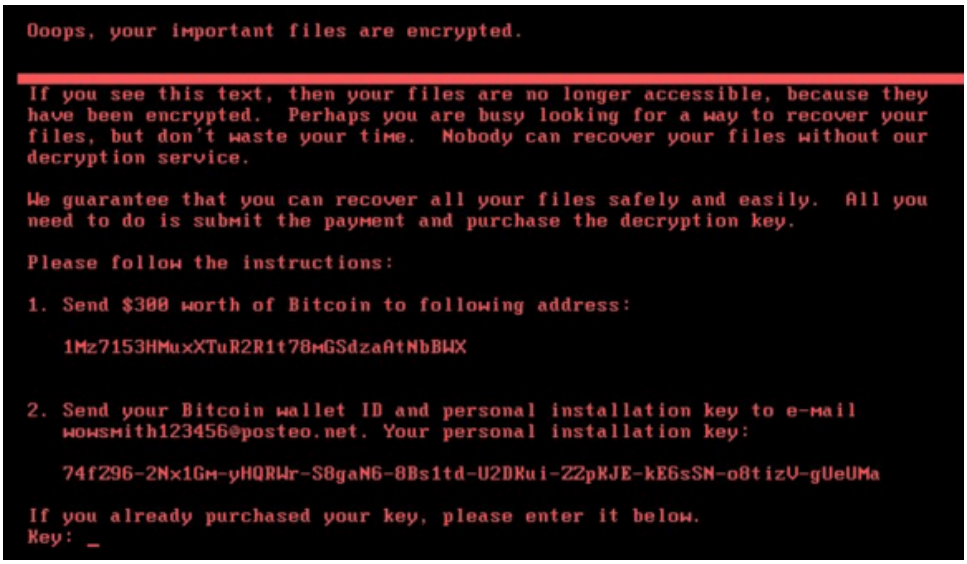


Figure 1: The ransomware "Petya" display-screen(2)

When the malicious code entered a system, it would not only encrypt all data but also damage it to a state beyond repair. The attack portrayed itself as the ransomware “Petya” that hit companies and individual windows OS users in 2016. “Petya” would encrypt all data on a system by making a boot override by using previously unseen tactics, and then demand a sum of 300$ in bitcoin to decrypt the files. (3) The blackmailing-screen appearing after infection is shown in *figure 1*.

A system infected by the “NotPetya” malware, would show the same blackmailing screen as the “Petya” ransomware, but the bitcoin transfer process was just a decoy. Behind the encryption, the malware was working full time on damaging all existing files stored on the system, acting as a destructive masterpiece. The estimated cost of damages done to the Danish logistics company Maersk resulted in a loss of an estimated 2 billion DKK.(4)

There have been discussions about the success of the attack, since some people believe that it did not extract any value directly from the attack. However, some people believe that the attack had been executed for a more political motivation instead of being directly about the income from affected users – meaning that the attackers probably had a financial or political benefit

from the affected companies losing billions in profits. Cyberattacks executed with political or ideological intentions are often defined as acts of cyberwarfare.(5)

The possibilities of cybercrime are virtually endless, and companies of all sizes, as well as individual users of the internet, can expect to be attacked.(6) The "Petya" and "NotPetya" ransomware attacks had extreme consequences for the companies and governmental institutions that were hit, but attacks aimed at individual internet users like "The invisible man", "Gooligan" and "Bank bot"[2] can be just as extensive.

Extracting personal information from smartphones, is a target of a large part of the cybercriminal environment. Information linked to personal identities and banking information, is stored in massive amounts globally and often proves easy to sell afterwards in criminal markets.
This chapter focuses on the cybercrime involving smartphones. Extracting personal information from smartphones, is a potential goal of the cybercriminal environment, but which data is of interest for hackers to acquire from average smartphone users? This will be investigated in the following chapter.

2. Sensitive data on smartphones

This chapter will investigate which data an average user may have on their smartphones, and why such information is important and interesting for hackers. To do this, it will be explored what type of information hackers have already stolen from average users, and put up for sale on the dark web and if such data is easy to find and purchase. What kind of protective measures are available for smartphones.

Currently, over 2.32 Billion people own a smartphone,(27) many of which use applications that store credit card information. These transactions are considered by big companies such as Alphabet and Apple to be secure, since they spend a lot of resources on securing their own services to avoid data leaks and scandals. However often the implementations are weaker than the actual methods researched into.

[2]Appendix 2

To use services such as the android and apple stores it is required to create a google/apple account. One of the risks involved with these user accounts, is that users tend to pick the same passwords and generally these personal passwords are rather weak.(7) As the article shows, most people reuse their passwords several times across many different applications, and this is considered as having an unsecure or weak password. As an example, for this is the most recent LinkedIn breach, here about 35% of accounts were actually using a password, from a previously known location. Meaning they have now been recognised to have patterns in their password management, which is never good since it makes passwords more predictable and easier to crack.

This means that if spyware can obtain one's google or apple account´s password, several other accounts and personal data are threatened.
A Google account also gives access to the user's email, which means that a hacker who has access can also respond to email verifications and therefore log in to sensitive personal data such as medical records and bank accounts as illustrated in *figure 2*.

Figure 2: The access to social media through Google mail

In some countries such as Denmark a two-step authentication system(8) issued by the government helps and forces people to have additional safety. This is done through a physical piece of paper with numbers and corresponding keys.

However, a problem with this physical layer is that it is possible to simply take a picture of it and store it on the user's phone. If a hacker has access to the user's phone it becomes a simple matter of downloading the picture to mitigate the entire process. In direct response to this, it is

against the user agreement to take photos of these papers.(9)

However, it has still proven to be ignored by many users for simple ease of use. To get an idea of how many people have sensitive data on their phone, including photos of personal data such as the "NemID" papers, a user survey is conducted later in the analysis.

The research has shown that people use patterns when choosing their own passwords, which makes it easier to gain access to their personal- and confidential-data located on their device. However, is this data easy to access and buy?

2.1 Availability of stolen personal information

This chapter will be mentioning the dark web and using the Tor browser. It is recommended that the average person does not use the Tor browser to browse the dark web. The group will use the Tor Browser and the hidden wiki to access the dark web to see the amount of effort is required to gain access to personal information.

A way of purchasing access to another person's personal information is through the "dark web". The dark web consists of a lot of sites in the .onion domain. These sites are not publicly listed and are therefore not available through a google search. To obtain the links to such sites the user must first download the Tor browser.(10)

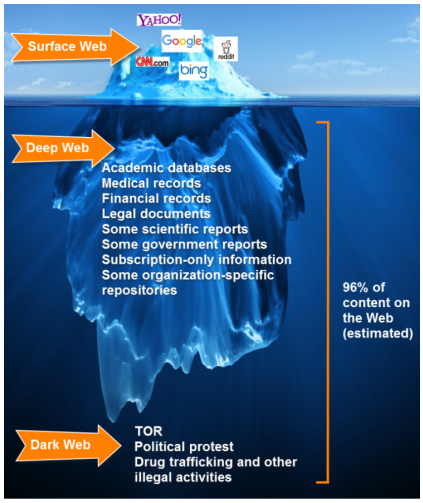


Figure 3: Shows a representing of the location of the dark web(11)

The Tor browser is an internet browser that attempts to make each user on the network anonymous. After getting the Tor browser, the user must then find a wiki with a list of .onion URL´s. One of the most used is “The hidden wiki”. As *"figure 3"* shows the web is only a small part of the internet, where most of the content on the internet is stored on the deep web and dark web. There are many versions of the hidden wiki with different levels of content. While some are moderated only showing legal sites some versions show .onion sites with illegal content. Common categories include:

- Buying illegal Drugs
- Bitcoin trading/money laundering
- Hiring Hitmen and Hackers
- Buying counterfeit Bills (Mainly USD and EUR)
- Buying stolen Credit cards, passports and other stolen personal data

While finding all of this may seem to be quite a challenge, the average user armed with the tor browser can find places to buy stolen credit cards, passports and more within minutes. Of course, the legitimacy of such sites is highly questionable. Most of them are outdated and rather basic, which makes it easier to fake a similar site.
But within just five minutes of the group searching on the dark web, six different sites claiming to sell stolen credit cards were found. If it is assumed just one of these sites is legitimate, then buying stolen personal information should be rather easy. This is shown in *"figure 4"*.

Figure 4: A visual representation of acquiring stolen data

This only goes for buying random people's personal data. If you have a feud with a neighbour and wish to ruin him/her by taking on tons of debt in their name, it is going to be a lot more difficult.

This is where the “hire a hacker” sites come in to play. Some of these sites, like rent-a-hacker

claim to be able to steal personal information for a large nominal fee. Sites like these aren´t nearly as common, and it must be noted that these sites might be fake as well.(12)
So, whilst finding sites that claim to be able to get stolen personal information are rather abundant on the TOR network, it is difficult to know if a given site is really trying to sell information, or if it is merely a scam. It therefore becomes hard to prove exactly how easy it is to purchase someone's personal data without illegally attempting to buy from multiple “vendors” on the dark web. While some people have openly claimed sites work,(13) testing on the subject cannot be done within the boundaries of the law, and therefore we lack a definitive answer.

Our research does provide us with information, that leads us to conclude that, an average user of the internet, can in a matter of minutes gain access to a hidden illegal site on the dark web, and possibly buy stolen identity and banking linked information. The next section will be about which security measures are relevant for smartphones.

2.2 Security measures relevant for smartphones

Due to the security risks involved with internet-connected systems, it´s all the more relevant to see what can be done to defend against and prevent future cyberattacks. Especially concerns about cloud technology on smartphones and the amount of data stored on them. A brainstorm on security measures, has revealed that it would be relevant to look at the following security measures:

- Antivirus
- User training
- Password manager
- Multiple factor authentication
- Honeypot

2.2.1 Antivirus

A common type of cybersecurity is antivirus. Antivirus is good middle ground for users to stay safe during browsing the web and downloading files. This is done with antivirus' URL's scanner and browser protection. Furthermore, file scanning can help detect and prevent infection of a user's device.

2.2.2 User training

Another kind of cyberthreat is called social engineering. In this kind of attack the initially exploited vulnerability is not located on the computer. Instead the user of the computer is tricked into allowing malicious software unto the computer.

Defence against mental tricks requires a vastly different approach, including raising awareness and teaching the end user how to spot social engineering attempts.

One might imagine a system to be deployed in training scenarios to map the extent of the vulnerability and to explore exactly which elements end up convincing the user of doing what the attacker wants. After having done this, the training scenarios might involve how to not get tricked into doing whichever tasks, the malware wants the user to do.

2.2.3 Password manager

A very different approach to increasing security is to consider the average user’s password strength and how many times each password has been reused by the user. While a password of sufficient length and complexity is relatively easy to deal with, even a moderate number of passwords of this type takes a vast amount of dedication to apply to one's digital footprint. This has not been done to a sufficient extent.(14)

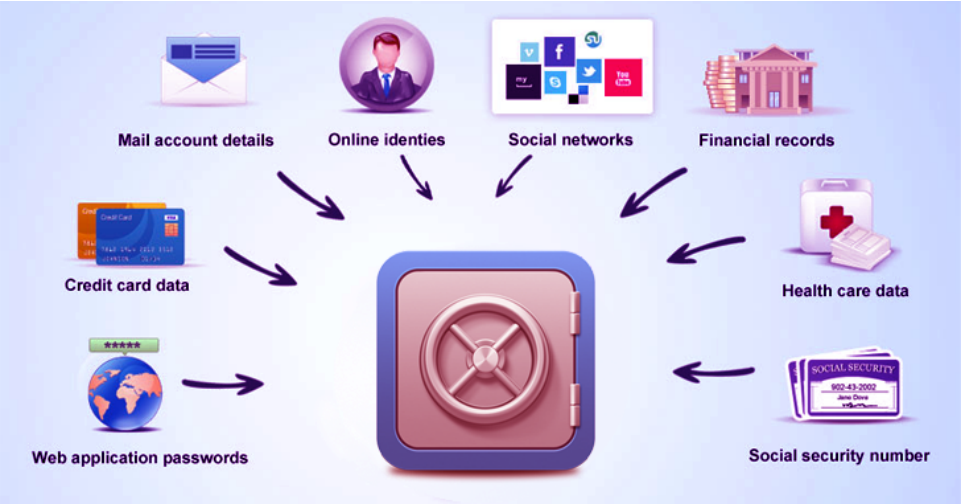


Figure 5: An illustration of a large number of passwords being contained together and encrypted(15)

To combat this problem, one could imagine creating a system to manage a great number of unique passwords. An illustration of such a system is shown in *"figure 5"*. This system would then be accessed by one very strong master-password and preferably several layers of authentication, with the password only being the first. This might become quite a hassle to deal with in everyday life, but thinking about it from a security point of view, then multiple layers are always better, since it's going to take longer to break down more layers than fewer layers. The efficiency on more layers will depend on the implementation of the layers.

2.2.4 Multiple factor authentication

Every password-based login could be coupled with a number of other layers of security such as small offline pieces of hardware generating additional security keys on the fly as represented by *"figure 6"*. Another two layers could be iris scanning and fingerprint scanning. Each of these layers can be cracked, but they all require different approaches and therefore compromising all of them at the same time would be more difficult than having to just crack a relatively simple password, which has been mentioned earlier.

Figure 6: A "NEM-ID" key-generator

On the other hand, piling on layers upon layers of security can easily result in the end user being frustrated in day to day life. While some might be willing to live with the hassle, others might opt to circumvent these security layers or outright disable them.

2.2.5 Honeypots

The biggest advantage of a honeypot is that it´s possible to learn more about the attacks themselves, and how they operate. Hereunder see specifically what the attack is doing, such as:

- Deleting files
- Granting themselves administration privileges
- Etc.

deleting files, giving themselves administration privileges etc. It´s even possible to see the exact commands used in the attack as well as monitor any file transfers the attack may initiate to and from the honeypot.

Often IT gets overwhelmed with data from audit trails, to assist with these audit trails there are some tools to help you monitor several things: devices, apps, logs and even entire systems. These tools are both good and bad, it's good that they help you monitor all of the necessities to make it easier for IT people to look through, the bad thing is that these tools just generate way too much data with no indications whether the alert is low- or high-priority.
With a Honeypot, you often get fewer alerts so it is easier to manage, analyse and then act upon the analysis, and lastly throw out the attackers before any serious damage is done.(16)

2.3 Summary

This chapter indicates that cybercrime is an issue, but preventative measures are being used.

There are many options on how to fight cybercrime and when considering which type of system to develop, it´s also important to compare the amount of already available solutions in the different categories. For example, there are already hundreds of antivirus solutions, whilst there are only a few notable honeypot systems.

Therefore, it has been deemed interesting to look into a Honeypot system. A honeypot system is easily expandable making future development a distinct possibility. On the other hand, a honeypot system does not immediately improve the security due to it having to find the malware and then analyse the behaviour of the malware, before being able to categorise and remove it.
Therefore, it has been decided that it would be interesting to look into building a Honeypot system basis, which will be the focus of this project, as it will allow for effective future development and give insight into the behaviour of malware.
Now that it has been decided that a Honeypot will be built, it would be valuable to know which stakeholders would be interested in a Honeypot and how much it could benefit them and the project. Therefore, the next section will be an analysis of the project's potential future stakeholders.

3. Analysis of stakeholders

Now that the product has been decided it is time to find out who it could benefit, and why it benefits them.

Improvements within the field of cybersecurity are in the interest of many different people, organizations, companies and individuals. Everywhere that network communication is implemented alongside storage of sensitive data, securing these data is essential.
Discussing which sector has the most need for IT-security improvements, is a long, complex and probably impossible road.
As concluded in the previous chapters, sensitive data is stored on personal devices connected to the internet as well as local networks for companies and organizations, that largely depends on this data being accessible only by people authorized with access.

This sensitive data is often linked to personal identity, economic relations or corporate secrets. Social media applications, mailing systems and programs handling transactions are just a few examples of applications usually represented on an Android device, that saves these types of data.[3]

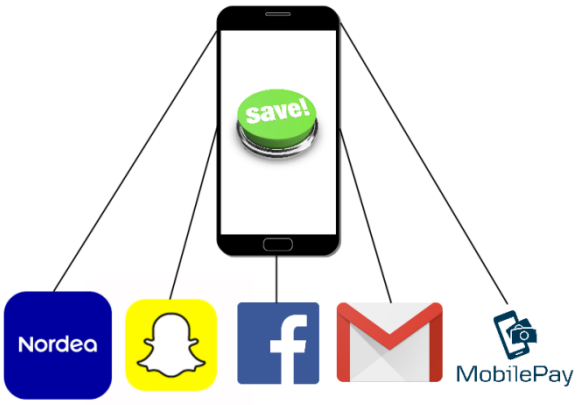


Figure 7: Illustration of applications storing sensitive data on smartphones.

From the analysis concerning the accessibility to illegally buying sensitive personal data, it has been concluded, that it is possible to buy an entire personal identity in a matter of minutes (Social security number, Banking information, Credit card information, Name and address) for a fairly small amount of money. And this can be done with marginal technical expertise, by an average user of the internet.(57)

From this, a demarcation to our project has been made. The end-goal is to improve IT-security, and the problem has now been specified to such an extent, that sensitive data linked to personal identities and financial information stored on smartphones is set as the focus of the project. Statistics of the smartphone sales to end users on a global scale, show that up to 86,1% of all smartphones are running the Android OS.(17)
This leads to further specifying the project, to focus on the Android operating system, as that would arguably have the greatest impact on improving IT-security.

[3] As described in section 4. Smartphone survey.

3.1 Description of method

To identify the relevant stakeholders for a given project a "stakeholder analysis"[4] can be carried out. This analysis is used to clarify what parties the project has potential to affect, and who should influence it to achieve the best possible results. The execution of the analysis has been divided into four steps as described below:

1. First, the stakeholders are identified by a brainstorming process. Stakeholders could be end-users, retailers, manufacturers, researchers, NGOs, or governments to mention a few. What they all have in common is that they are either affected by the project and/or have the potential to influence the project.

2. The identified stakeholders are categorized into four categories according to the two mentioned parameters – Possible influence on the project and how great of an outcome they are able to receive from the results.
    a. External (Low influence – little outcome)
    b. Grey Eminence (High influence – little outcome)
    c. Hostages (Low influence – great outcome)
    d. Resource person (High influence – great outcome)

3. Next is a discussion of the degree of each stakeholder's potential outcome as well as their potential influence, is conducted to better understand and evaluate each stakeholder.

4. From the attained understanding about parties with interest in the project, a further specification of the project-direction is defined.

Now that the necessary information required to use the stakeholder model has been acquired, the next step is to implement the stakeholder analysis.

[4] PV6 "Interessentanalyse og økonomi"

3.2 Implementation of analysis

To establish a general view of possible parties with an interest in the project, the following stakeholders and their possible influence/outcome of the project has been identified as shown in the table below.

Interested party	Possible influence	Possible outcome
Generic end-users	Share information about the different types of sensitive data they personally store on their devices → Ensure that our research is focused on the most relevant types of data	Improvements to security on Android devices → Users will feel more comfortable storing personal information on their devices → More personalized user experience and better usability
Cybersecurity researchers	Act as "expert sources" → Improve our understanding of the technical aspects → Better specification of focus	No obvious outcome.
Companies/Organizations	Share information about recent cyberattacks → Make sure that our research will be relevant and up to date	Improvements of security on devices used my employees/workers → Fewer successful cyberattacks on the companies' digital platforms and allow the users to be more comfortable storing and using already stored data
The Danish Centre of Cybersecurity	Share information about the threat of cybercrime in Denmark → Improve the specification of our research	No obvious outcome.
Google offices	Supply us with a number of fake Google accounts → Make the emulation of our virtual machines more realistic, as they can be connected to a social media account	Improved security on Android devices with linked up a Google account → More people using services provided by Google and comfortable storing data
Manufacturers of Android smartphones	No obvious influence.	Better security on Android devices → More people would be comfortable buying Android smartphones
Developers of Android apps	Share information about how they keep user data securely stored and what types of cyberattacks they are affected by most frequently → Improve the specification of our research and analysis of data from Honeypot	Better security of stored user data in their Apps → Improve user experience → More users
Banks and transaction apps	Share information about how financial and personal user information is stored → Improve the likelihood of our research resulting in an improvement of relevant IT-security	Improved security of stored sensitive data → Easier and more secure transactions and fewer successful cyberattacks with the intent of stealing sensitive data → More users and increased user experience

By evaluating each interested party based on their influence on the project and possible outcome of the result, they are further categorized into four categories as described in the "*Description of method"* section. The outcome is illustrated in "*figure 8":*

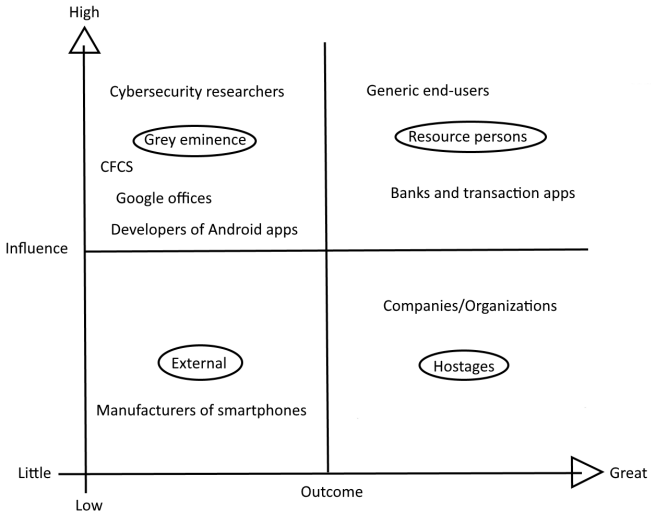


Figure 8: Categorization of interested parties

By looking at *"figure* 8" it becomes apparent that the main stakeholders are the Generic end-user and banks with a focus on their online transactions apps. Banks and third-party companies developing transaction apps makes for high priority targets of financial cybercrime and identity theft, since they depend on large amount of sensitive data stored in their systems. As an increased number of users handle their finances on their mobile devices, security improvements in the associated IT-systems are needed to secure the data from outside attacks.(18)
Banks and developers of Android apps handling transactions and storing financial information on users, are potentially interested in the results of the project.

Obtaining access to information revealing vulnerabilities in an IT-system is one of the most impactful drawbacks for cybersecurity research.
Most stakeholders are particularly careful, as these data has potential to support cyberattacks with massive consequences. It creates a dilemma, as it's a necessity for cybersecurity researchers to have access to these data, to contribute effectively to an increase in security systems.(19)
This leads to the formation of the following question:

- How to obtain access to information revealing security vulnerabilities on Android smartphones, and data about the frequency of typical malware attacks?

From the analysis, it is evaluated that banking institutions and developers of financial Android apps as well as generic users of the Android OS, has the potential to greatly influence and benefit from the project – but in two very different ways.
As discussed above it is a difficult task to convince companies to reveal vulnerabilities of their systems, as it would expose them to risks of cyberattacks.
In many cases this is not a risk they are willing to take – hereby deeming the process of dedicating data from financial institutes and banking app developers to be out of scope for this project.
An alternative approach is to research the data storage behaviour of generic Android users, by using an anonymous quantitative method like a questionnaire. Forming a survey focused on producing answers mapping this behaviour and the types of sensitive data stored by the majority of Android users, is expected to yield the necessary insight for this project to be further specified.

4. Smartphone survey

A survey is a tool used to get a broader perspective of general user behaviour. A survey in contrast to an interview gives less of an extensive view of an expert individual but gives insight into a larger amount of people. This in turn gives a better basis for using the data in a general sense rather than using an expert view as the general consensus surrounding the issue.
It is important to know how to conduct a survey to not mislead participants. The Survey will give insight into which types of explicit data are stored on smartphones. It will be especially helpful in focussing the project and potentially reinforcing the necessity of the project.

4.1 Description of method

To conduct a successful survey, a number of factors should be taken into account. A significant amount of attention to detail with the questions is needed.
To avoid complication, it is important to transform the research questions into survey questions instead. With the prerequired knowledge the group has, the survey questions can be answered

to a clearer and satisfactory amount which is beneficial for the participants and creators of the survey.

Participants will gain a greater feeling of understanding and the creators will give more people an incentive to answer. There should be a progressive form in the survey going from the simple questions and avoiding unnecessary questions that have no beneficial information to more specified questions. An example of an irrelevant question could be asking for the participant's age. As the survey focuses on the data on the smartphone and not on the individual user, the age of each user does not hold any meaningful information. Lastly it is important to test the survey before sending it out to a larger audience, to make sure that the survey can be understood and it can convey the questions properly.

Mnemonic rules when creating a survey[5]:

- The questions have to be unambiguous.
- The wording of each question must make a participant want to answer truthfully.
- The order of the questions must be evaluated closely. Questions must not lead specific answers in the next.
- The survey must be tested on a test-group of people from the relevant target group.

4.2 Implementation of user-survey

Currently Smartphones have seen like most technology, a rapid progression. Access to the internet means a chance to be exposed to cybercrime. To understand safety and user perception a survey has been made to provide insight into modern day smartphones users.

[5] PV lecture PV7 "Metodeseminar" slides 27-30

By using the information that is obtained from the survey, our group will be able to have a better picture of what the average smartphone user's habits for storing information are. By analysing the data from the survey, it will be possible to specify the focus of the project.

In this project, SurveyXact[6] has been used to generate the survey. It made it easy to make a well-made survey to forward to users and to check regularly about the information obtained. To broaden the group of potential people to answer the survey, it is aimed to be answered by smartphone users of other operating systems as well. As it is a quantitative questionnaire, we are dependent on getting a large number of results. By letting the pattern of storing sensitive data for users of all different smartphone operating systems being similar, we make sure that the survey is not too limited. Since this is a quantitative survey there is a dependency on getting a large enough amount of answers. The questions will be similar for any operating system to make sure the survey is not limited to a specific OS.

The manner of how the survey is done and which questions are asked can be seen in appendix 1 and 2.

The survey, when implemented in a meaningful way, gives a broad perspective into general smartphone user's behaviour. This raises the question of if the survey has discovered any worrying signs about user behaviour.

4.2.2 Correlation between the survey and general concerns for mobile users

Several participants' answers in the survey indicated a concern about smartphone security.

There is some concern about user safety. 63 participants took part in these questions. 32 % said yes to having credit card information stored on their local device. This is a one-third of all users that can become potential victims of cybercrime.
Maybe most concerning is that 17% said yes to having confidential papers on their phone which leave them open to identity theft should their data be compromised and almost 50 % do not have a concern for identity theft. Furthermore 21 % have had an attacker attempting a login on accounts on their device which could give them access to sensitive data. An example for this is

[6] A Survey creation program made by Ramboll. (https://www.surveyxact.dk)

an email sent out to the user in an attempt to login to Facebook from a source in Asia.
The results of the survey, gives reason to believe that personal data on phones is at risk.

4.2.3 Summary

In conclusion, a small percentage of users are a high-value target due to easy access to confidential information stored on their phone. Despite 21% having experienced an attack on their private data, 74% of users rate their comfort in storing personal data on their smartphones highly. This shows the trust mobile users have when they surf on their mobile devices. The survey is an indication of the easy accessibility of important data and lack of user awareness which could be a significant security problem.

4.2.4 Reliability of survey

To support the conclusion from the previous section other sources deemed reliable will be found. This conclusion is very important to the project and is therefore important to validate.

Sites tend to focus on Android due to a massive market for Android. Android is more open-source than Apple devices and slower at updating its OS. The consequences of this are more user freedom but also more security risks. Furthermore, Apple has strict guidelines for apps.(20) In 2011 smartphones even outsold PC and since then smartphones have been developing even further.(21) *Figure 9* illustrates this development in a graphical table:

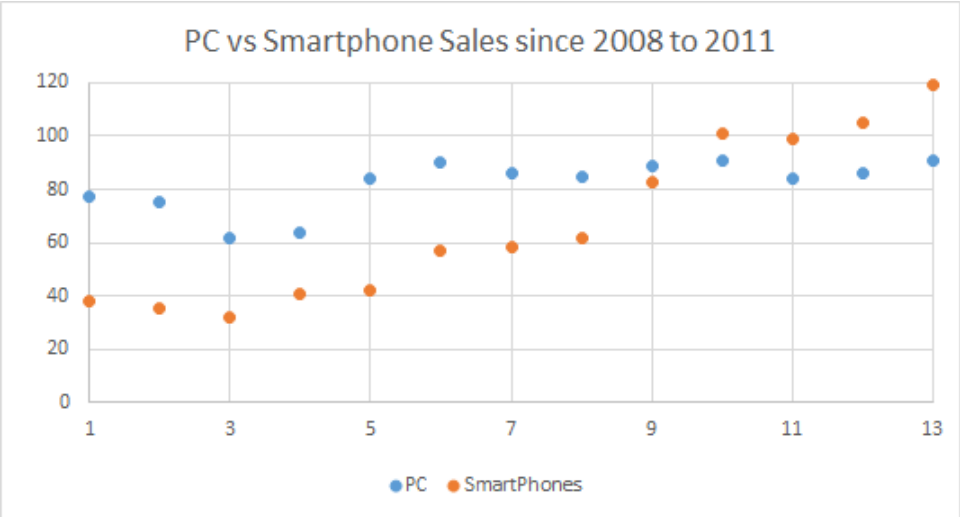


Figure 9: Graph of sales of PCs and Smartphones 2008-2011(21)

The y-axis is the number of sales in millions and the x-axis starts with the third quarter of the year 2008 and then progress one quarter each time. The blue series is sales of PCs and the red series is sales of Smartphones. The numbers are from the study conducted by the IDC reported by the Guardian.(22)

IDC smartphone share, 3Q11

Vendor	3Q 11 shipped units (million)	3Q11 market share	3Q10 shipments	3Q10 share	Change in sales
Samsung	23.6	20.00%	7.3	8.60%	223.30%
Apple	17.1	14.50%	14.1	17.00%	21.30%
Nokia	16.8	14.20%	26.5	32%	-36.60%
HTC	12.7	10.80%	5.9	7.10%	115.30%
Research In Motion	11.8	10.00%	12.4	15.00%	-4.80%
Others	36.1	30.60%	16.6	20.00%	117.50%
Total	118.1	100	82.8	100	42.60%
					source: IDC

Figure 10: A number of different vendors in the third quarter in 2011 sale statistics(22)

Statistics show, as in *figure 10,* a large increase in sales for Android in particular. Despite the small increase in sales compared to Android, Apple remains the second largest distributor of

smartphones. The table further indicates the large market of Android and a general increase in sales for different types of smartphone brands.
Smartphones have quickly become the number one accessible for most personal data for most people. However, the security on said devices is lacking behind the security on most PCs. This is based on a report made by the US Computer Emergency Readiness Team or US-CERT for short. They reported back in 2011 that mobile threats were on a rise back then and vulnerability in mobile phones rose 42% just from 2009 to 2010.

Like our survey the report by US-CERT reported that many people didn't think surfing the web on the phone or in general pose little to no danger, but the truth is mobile phones are getting more and more at risk. Old PC malware and social engineering tricks are being used effectively on mobile phones making it clear that mobile phones are clearly behind pc in security.(23)
Like the US-CERT, MalwareBytes share the concern of a loss of a mobile phone today means much more. It is relatively easy to extract data from the phone and with 17 % of our survey having pictures of "NEM-ID" or other personal information they would potentially be at a financial risk.

Besides that, Pew Research Center released a report in 2015 where they reported 57% of Americans do their banking through smartphones.(24) This is the same year when 1.4 billion smartphones were shipped out and sold worldwide. (60)

The Senior MalwareBytes analyst Nathan Collier in 2015 and the report from the US-CERT in 2011 all speak up against the nullified indifference there is to security in mobile phones and what measures you should take to take caution against attacks.
Banking and personal information is a part of most phones today, with 98% of our survey with some form of social media apps and 32% with credit card information stored on their phone and 17 % with even more information like driver license or "NEM-ID".
Besides all this most in the Survey had iPhones but the general picture most users are Android users and Chinese internet portal Tencent conducted a study that 27% of users rooted their de-

vice, wherein our survey we only had 8%.(26) In general, there are a select few who would be in a high risk of potential data loss and the overall problems and advice from 2011 and 2015.

Technology though is in a constant state of development and therefore old sources are quickly outdated. Therefore, sources from 2017 are needed about the present usage of smartphones and security here about.

Symantec released in April 2017 an Internet Security Threat Report or ISTR for short(25). Symantec gathers a significant amount of data through its technologies.
Furthermore, The Symantec Global Intelligence Network^{tm} tracks 700.000 global adversaries and uses 98 million attack sensors worldwide. The network monitors threats in 157 countries. The report includes a section about mobile phones.(24)

In the report Symantec makes a number of key findings. These are summarised[7]:

- *"The Android operating system remains the main focus for mobile threat actors. However, security improvements in Android's architecture have made it increasingly difficult to infect mobile phones or to capitalize on successful infections."*

- *"Attacks on the iOS operating system are still relatively rare. However, three zero-day vulnerabilities in iOS were exploited in targeted attacks to infect phones with the Pegasus malware in 2016."*

- *"The overall volume of the malicious Android apps increased significantly in 2016, growing by 105 percent. However, this rate of growth has slowed when compared with previous year, when the number of malicious apps increased by 152 percent."*

- *"Symantec blocked 18.4 million mobile malware infections in total in 2016. Data from Symantec-protected mobile devices shows that 1 in 20 devices will have experienced an attempted infection in 2016. Similar levels were observed in 2015."*

[7] Samme som 39

Symantec reported 3.6 million mobile malware detections in 2014. This increased 9.0 million in 2015 and increased even further in 2016 to 18.4 million cases of mobile malware. The number of malware family types has stagnated though. It increased from 277 in 2014 to 295 in 2015 and only a slight increase to 299 in 2016 indicating a stop in growth in mobile attack innovation.
The different malware types recorded was 2,227 in 2014 and increased by a large amount in 2015 to 3,944. This could give cause for concern but in 2016 this fell to 3,634 in 2016 further indicating a stagnation in mobile malware development.

A study by Symantec in the vulnerabilities in different mobile OS-system show that Apple had 178 and Android only 12 in 2014. This increased for both OS-systems in 2015 with Apple having 463 and Android 89. This changed significantly in 2016 with Apple providing added security and decreased the vulnerabilities to 290 and Android increased to surpass Apple with 316 vulnerabilities.

According to Symantec most mobile malware is still used for a financial purpose and most malware detected in Android which is on Symantec's top ten list, have a financial purpose. This includes Android.Opfake which brings in a large amount of revenue for attackers. The report backs up the concerns of the 2011 US-CERT report and 2015 Nathan Collier analysis.(25)

4.3 Conclusion

From the introductory analysis provided in the section above, cybercrime is determined to be problematic on a large scale for generic end-individual users as well as organizations and companies, and often they're tied together.
The estimated amount of individual smartphone users on a global scale is set to be over 2,3 billion by the end of 2017.(27) This is an increase of the 1,4 billion units sold in 2015. Supported by the conducted survey, it can be assumed that a certain amount of these users has personal or financial sensitive data stored on their devices, which makes them a high-value target for cybercriminals.

Following that the Android operating system represents a distinctive majority on the global market of smartphones, it is determined that focusing the research of this project on improving security for Android users, will lead the most impactful results.
The Android operating system represents a majority of the global smartphone market and attackers' favourite OS to attack. This project aims for the most amount of impact possible as of this reason the project will be directed to focus on Android users.
By conducting a user survey and comparing the results with other sources on the subject, there is a large number of attacks on Android that targets sensitive data with the goal of acquiring money form the victims. Multiple insecure data storage behaviour has been identified. Some of the noteworthy observations are:

- Approximately one fifth in the survey have confidential information on their device
- Approximately one fourth or one-fifth of the users in reports have rooted/Jailbraked their device.

As part of the solution to the global problem of rising cybercrime, honeypot systems are evaluated to have the potential of improving cybersecurity by acting as controlled environments, where malicious software a hackers' behaviour and tactics can be learned from.
For a Honeypot system focused on improving security for Android users to be effective, it is necessary to realistically emulate the true behaviour of these users. In many ways, the evolving technology of data sharing on mobile devices develop faster than the security can.
Smartphones repeatedly outsell PC's after the year 2011,(22) and focus on security improvements on mobile devices are therefore, more important than ever. There are several methods of which study of attacks could be done. The method has to be low-risk and still be able to gather some amount of data where malware can be studied and used purposely. It should be able to develop even further upon when the project is done.

As a contribution to the solution of the rising global issue of cybercrime, the Honeypot technique is assessed to be an effective tool of use. By infecting virtual Android machines emulated to the

use of a generic smartphone user with selected malware in a controlled environment, it is rendered possible to attain results indicating the realistic behaviour of the examined malware.
To shed light on this matter we will:

- Research if a predetermined malware type, can extract sensitive data from an Android device emulated to resemble a generic Android user.

The research will be built around the creation of a Honeypot system consisting of:

I. A test environment of virtual Android machines.
II. A containment system consisting of an IP-tables firewall and a bandwidth limiter.
III. A system with the ability to monitor and filter outbound traffic.

5. Requirement specification

Due to the analysis of the given problems with mobile malware and especially Android users, it was concluded that a Honeypot system with the ability to effectively emulate realistic Android users could be part of the solution.

This is also the same defence technique that is used by Symantec. They implement an IoT (Internet of Things) Honeypot to defend against malicious software.(25) For the creation of a Honeypot system, with the ability to effectively be used as a tool for researching if a predetermined malware type, can extract sensitive data from an Android device emulated to resemble a generic Android user, the following requirements are set:

- As a Honeypot is used to monitor and analyse the behaviour malicious software, it must be capable of securely containing the malware allowed into the system, and prevent it from spreading through the network.
- The conducted survey appointed some noteworthy user behaviour concerning storage of sensitive data on smartphones. The Honeypot must be able to emulate a generic Android user, refer to chapter 4.3.

6 Creating a Honeypot

6.1. Honeypot basics

Given that the project focuses on the creation of a Honeypot it is prudent to further discuss the basics of such a system. This next section will describe what a Honeypot is, what modules it is built from and what it can be used for.

A Honeypot is fundamentally a dummy computer, or a network of dummy computers meant to divert attention towards themselves by seeming vulnerable and valuable. This is done by running automated actions that mimic those of a real production system, without having any actual valuable data present. Adding software capable of data collection and analysing collected data, changes the characteristics of the honeypot from merely being a lightning rod to divert intruders, to being a defensive research station. As a Honeypot purposely diverts malicious traffic into the system for research purposes, a containment system is used to control traffic to effectively prevent and guard against, malware spreading through the network.

This structure can be divided into three main modules, each with a unique responsibility:

- **Test environment**

Module responsible for running virtual machines emulated to the desired use.

- **Monitoring and analysis**

Module responsible for monitoring the ongoing traffic in the test environment, collecting data from its behaviour and analysing gathered data.

- **Containment.**

Module responsible for keeping malicious traffic inside the Honeypot, and prevent any damages to the connected network.

Such a system-structure is illustrated in *figure 11.*

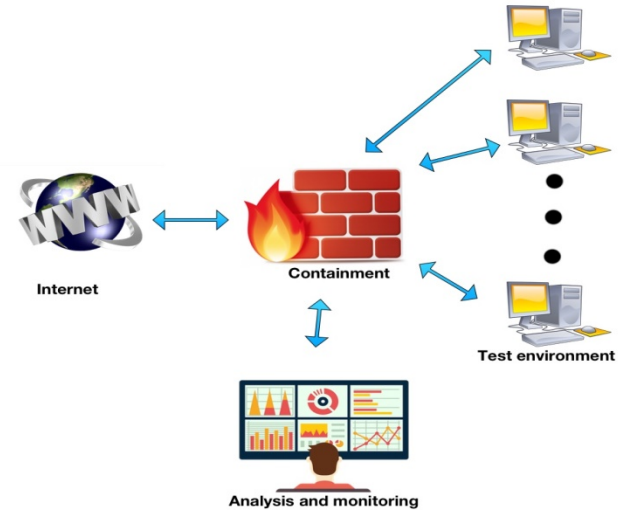


Figure 11: Illustration of the structure of a Honeypot system

If an intruder tries to run some exploit on the Honeypot, which would have yielded useful information on a production system, the network owner will then be aware of a security weakness without ever having lost any sensitive information given the nature of the honeypot.(37)

Running a Honeypot therefore serves two different purposes at the same time. Firstly, a Honeypot can trap intruders in a system where they pose no threat to the network owner hereby wasting the intruder's time. Secondly, the collection of data can reveal previously unknown flaws in the system allowing for pre-emptive security improvements.

When choosing to set up a Honeypot one must consider the interaction level of the Honeypot. If a Honeypot is set up to be very restrictive that would severely limit any given intruder's capacity for malicious activity. While this would be quite a safe setup, it might however be fairly easy for an intruder to realize that system in question is in fact, a Honeypot.

If this occurs to an intruder, the attack would cease, and further data collection would not be possible. This kind of setup is called a low-interaction Honeypot, because the possible interactions are relatively few.

To add some credibility to the Honeypot it would be possible to emulate certain services like DNS and HTTP to let the intruder believe that the Honeypot is a real system without even letting outbound traffic exit the local network.(37)

Conversely one could choose to set up a very insecure system with minimal or no restrictions on interaction. This high-interaction Honeypot might also allow outbound traffic to the internet through. This kind of setup would allow an intruder to progress much further into the system and would therefore allow the network owner to gain much greater amounts of data from an intrusion.

This does however pose an ethical question since allowing an intruder to take over and use a system for malicious purposes, makes the owner of the system carry some responsibility in the intruders continued activities.

One way to make a Honeypot seem more like a real system while minimizing the risk could be to set up a Honeynet. This would allow an intruder to send outbound traffic, like on a high-interaction Honeypot, but the network would then redirect the traffic to another Honeypot on a different network. This would in essence, allow the Honeypot to safely trick the intruder into believing, that there are no restrictions on outbound traffic.

Another kind of Honeypot configuration is called a shadow Honeypot. The idea is to put an Anomaly Detection System, or ADS, before the network itself, as illustrated in figure 12. Any regular data traffic is allowed through to the real production systems, while any anomalous traffic is redirected to a Honeypot, or a group of Honeypots.

While this screening process adds some amount of overhead to the entire network, it would be able to actively catch and isolate malware. This would make such a system even more useful as both an immediate malware defence as well as a source of information, allowing for future improvements to network security.(37)

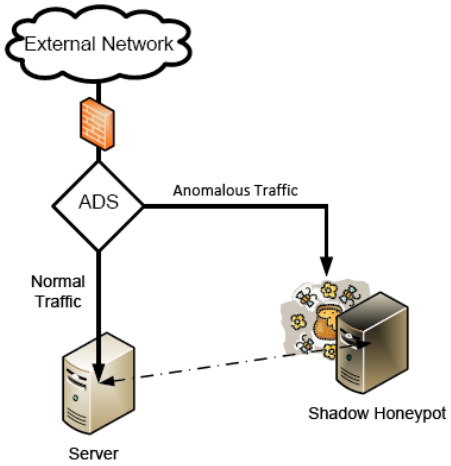


Figure 12: Illustration of a shadow Honeypot system(37)

In conclusion, it has been found that a Honeypot can be used as a defensive tool by diverting attacks and in the process, learn about the malware. When engaging in malware research, understanding the basics of malicious code is essential alongside a careful consideration of certain ethical considerations related to the subject.

6.2. Malware

In this chapter, the basics of malware will be explained followed by examples of specific types. Subsequently relevant ethics and risks involved with cybersecurity will be discussed, with the purpose of acquiring a deeper understanding of components involved with Honeypot systems. This knowledge will lead to a revised and updated requirement specification.

6.2.1. The basics of malware

This section will focus on describing the fundamentals of what malware is, and what malware does.

Malware is an abbreviation for “malicious software”, hinting towards any kind of software that has disruptive or damaging consequences for an infected computer. The Oxford dictionary of English, describes malware as follows: "*Software which is specifically designed to disrupt, dam-*

age, or gain authorized access to a computer system."(28) The types of malware will discuss different disruptive or damaging behaviours of malware. As this project is focused on researching the behaviour of malware coded to steal personal information from Android smartphones, the types with these capabilities will be of priority.

6.2.2. Types of malware

There are currently several different kinds of malicious software in use, which each has its own unique purpose, but some malware work in very similar ways. This following section describes different malware types:

- Phishing.

  This type of malware poses as something legitimate, but when you access the website or link it tries to fish for information. It tries to deceive you into thinking you have visited your website or link, which then requires your personal information, which can be either harvested or collected by hackers.(29) This can be more than just login details for the specific sites but also credit card information or other confidential information.(30)
- Spyware.

  Malware of this type monitor your movements on the internet. It sends information back to a central computer, which spams you with ads.(31) This is also known as adware. This can maybe seem harmless, but it can slow the computer to a point where a system reset is necessary because the spyware is extremely difficult to remove.
- Keyloggers.

  Keylogging malware is a form of spyware, which records everything you type on the computer with the intent of getting passwords and other sensitive information. This type of malware has use for corporations, and sometimes even parents, to log information of the user who is on the computer.(31)

Some of the malware that has been described, isn't necessarily evil or malicious, but is when combined with other types of malware. There are even some kinds of malware which do the same as something legal, this malware would be the adware type, which basically does the same thing as cookies when browsing.(31)

However, cookies are legal, since they ask for your permission to use your data, unlike the ad-ware which does it without asking. Therefore, it would be interesting to consider the ethics, risks and grey areas in the next chapter.

6.2.3. Ethics and risks in cyber security

When engaging in research within the field of cybersecurity, it is crucial to be cautious and aware of existing "grey areas" and ethical dilemmas. They constitute for important factors, considering that large parts of the methods used in cybersecurity often find themselves on the edge of the law.(32)

Individuals involved in cybersecurity are commonly divided into groups of coloured hats;(32) White hats, Grey hats, and Black hats. Each group represents a general viewpoint of committed research. White hats are hired by governmental institutions or companies supported by one of their cybersecurity engagements, to use hacking techniques to identify possible security breaches, and ultimately improve the security of the desired systems.
Black hats define the group of people, who develop and use hacking methods for criminal purposes. Grey hats define the middle ground, where they are not directly hired or supported by an authority, but instead of using hacking techniques to gather information about loopholes in security systems and selling them to criminals, they instead sell the information to governments, intelligence agencies or similar.(32) This division is illustrated in *figure 13*.

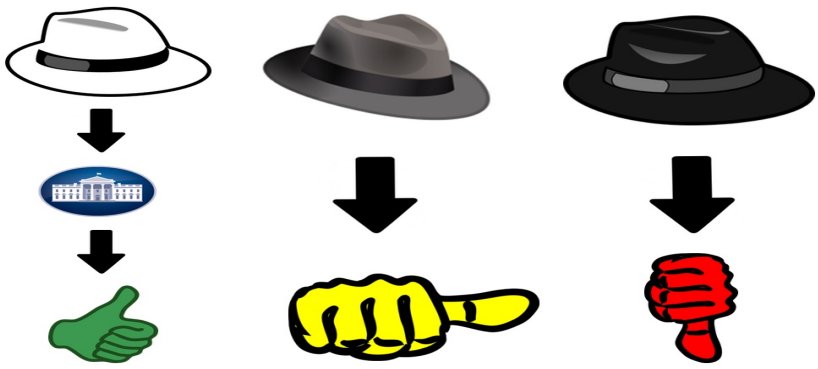


Figure 13: Illustration of coloured-hat hackers

There are no clear ethical standards or laws for right and wrong, leading to the white and grey hats, to sometimes be indistinguishable from black hats. In episodes where legal authorities have been involved, reputation and background checks alone of the discussed persons has been the only factors to determine the trustworthiness.(33)

Alex Stamos, former chief information security officer at Yahoo, is quoted for: "*You cannot promote the (true) idea that security research benefits humanity while defending research that endangered hundreds of innocents*".(33)
This quote is referring to an episode in 2015, where experienced cybersecurity researcher Chris Roberts, was arrested after publicly displaying serious security breach holes in the onboard network systems at the airline company, *United Airlines*. The incident was controversial of its type and made worldwide headlines.
One of the discussed arguments is that, even if Chris Roberts was to be trusted and had good intentions with his research, it could have ended in the hands of criminals, and therefore endangering the lives of hundreds of innocent civilians.(34)
The message of Alex Stamos' quote, is an expression for the difficulties involved with research endangering the lives of innocent people, while having intentions to benefit humanity.

In the latest cyber threat report from the Danish Centre of Cybersecurity (CCCS), the constantly evolving threat from cyber espionage and cyber criminality is rated as being "Very high".(6)
Taking the proportions of potential attacks into account; it is highly prioritized both in public state instances as well as in the private sector, to improve security against these attacks.

As stated earlier, today's' internet security experts use a variety of different techniques in the fight against cybercrime. Some of which involve breaking down systems, or at least trying to locate existing "loopholes" in computer networks, that could potentially expose confidential data from the system to criminals.
To effectively accomplish this, many IT professionals utilize the same destructive techniques as the black hat hackers. If not true to the reality of the global cyber threat, it won't be effective or matter at all.(35)

This defines one of the arguably most debatable grey areas: *Can there exist such a thing as hackers with good intentions? Is hacking a criminal act, even if used for the greater good?*

Essentially it is the philosophy behind the phrase "Violence begets violence"[3] that is the foundation of this ethical grey zone. In the context of the current cyber threat, this philosophy states that by using hacking as a tool to prevent the very same, you might end up promoting the criminal hackers' intentions and help them instead of fighting them. This sets up for a dangerous precedent.

Techniques, like Honeypots, are used by internet security professionals, as an attempt to set up traps for hackers and automated malicious code. In the process of luring a hacker or automated malware into closed controlled test-environments, these techniques often involve a process of purposely leaking data as a trail for the intruder to follow.
For these methods to produce results revealing methods of attack and communication, the intruder is allowed access as far into the system as possible.(36)

Setting up a Honeypot system, to analyse the behaviour of selected malware in virtual Android machines, could produce results on attack mechanisms in these systems. Analysing the produced data with the intent of locating patterns in the behaviour of the malware, could lead to an improved understanding of security parameters relevant for Android systems. Using a high interaction Honeypot system like so, exposes the system to the potential risk of the selected malware to successfully pervade some of the operating system's security measures, and potentially spreading through the connected network and cause damage.

As communication in a network becomes the analytical focus point, understanding the fundamentals of how communication is established in a system is essential.

6.3. Network communication

This section will discuss the basics of network communications and IP, TCP and UDP protocols associated with network traffic. This is important to understand the functionality and set up of a Honeypot, as well as the data produced.

To understand network communications requires knowledge of the OSI model and TCP/IP protocol stack. The Open Systems Interconnection, or OSI, model is the standard that all network communication systems optimally should follow. The OSI model consists of seven different layers and these layers operate independently of each other. This means that each layer only depends on the inputs and outputs of the surrounding layers. This structure is shown in *figure 14.*

	OSI Layer	TCP/IP	Datagrams are called
Software	**Layer 7** Application	HTTP, SMTP, IMAP, SNMP, POP3, FTP	Upper Layer Data
	Layer 6 Presentation	ASCII Characters, MPEG, SSL, TSL, Compression (Encryption & Decryption)	
	Layer 5 Session	NetBIOS, SAP, Handshaking connection	
	Layer 4 Transport	TCP, UDP	**Segment**
	Layer 3 Network	IPv4, IPv6, ICMP, IPSec, MPLS, ARP	**Packet**
Hardware	**Layer 2** Data Link	Ethernet, 802.1x, PPP, ATM, Fiber Channel, MPLS, FDDI, MAC Addresses	**Frame**
	Layer 1 Physical	Cables, Connectors, Hubs (DLS, RS232, 10BaseT, 100BaseTX, ISDN, T1)	**Bits**

Figure 14: This figure illustrates the different layers in the OSI reference model as well as some protocols relevant to each layer.

The idea is that each layer offers a service to the upper layer and relays the information on to the lower layer. The way that the internet is implemented does not quite follow this model. The implementation of the internet loosely follows this structure in what is called the TCP/IP stack.

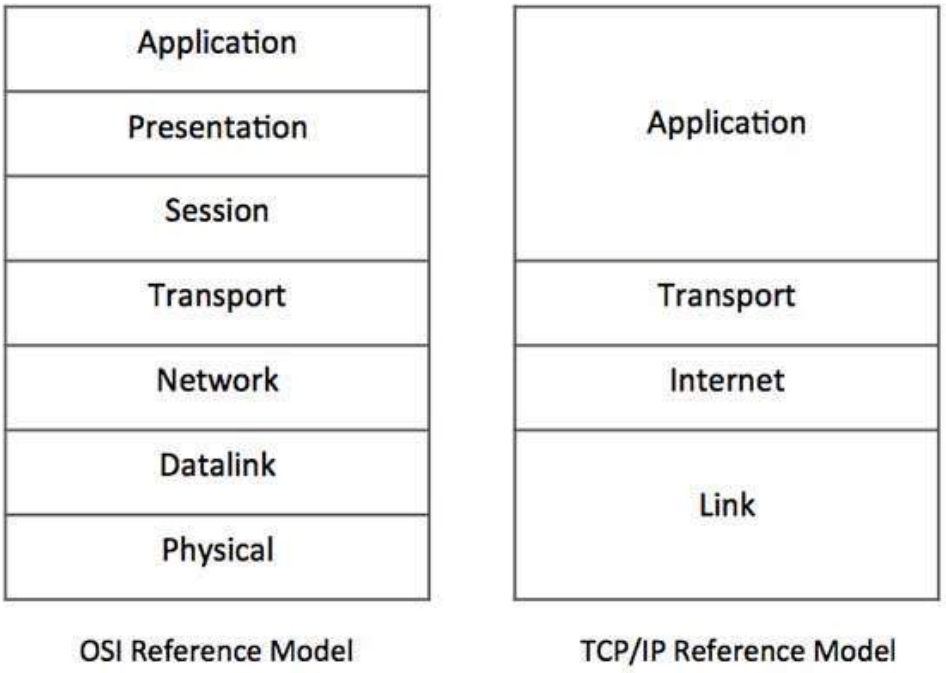


Figure 15: This figure compares the layers in the OSI model to the layers in the TCP/IP model

As seen in *figure 13* above the TCP/IP stack combines a few of the OSI model layers.
In each layer, several different protocols exist as seen in *figure 15* but these will not all be explained. Only TCP, UDP and IP will be explained, as they account for the primary transport protocols.

The link layer combines the physical transmission of bits from device to device with the datalink handling frames, comprised of bits, moving between adjacent network nodes on a local area network of a wide area network.

The internet layer handles routing through the network. This includes both choosing the shortest path but also choosing the least congested route. This layer is often associated with the Internet Protocol. This protocol is a connectionless protocol in that it does not require a connection to be setup and therefore has no way of knowing if the transmission is received correctly.

The transport layer offers end to end communication between hosts and it does so using the User Datagram Protocol or the Transmission Control Protocol. These two protocols have differing pros and cons.

If a message is to be delivered with little regard for latency then the TCP is very useful in that it allows for retransmission of corrupted data. Conversely if low latency is the primary concern then using UDP will result in lower overhead and therefore higher transmission speeds and lower latency.

Both transport protocols will often be used in conjunction with IP giving rise to the terms TCP/IP and UDP/IP. As mentioned earlier the IP is a connectionless protocol which is also true for UDP. Despite the connectionless nature of the IP it can be used in a connection-oriented way by pairing it with TCP.

The uppermost layer is the application layer and combines three layers of the OSI model. One could think of it as layers 1 through 4 handles the connection between two hosts and layers 5 through 7 can utilize this connection in differing ways. The first thing to note is that on these three upper layers the data to be sent or received is whole instead of being fragmented like in the lower layers. This allows for things like encryption and decryption among other things necessary for the given application, and also how the data is interpreted and presented to the user. These steps are parts of the upper three layers.

Basic understanding of TCP/IP is needed to understand how a Honeypot is useful. The reason being that a Honeypot acts like an end-user, meaning they have access to the upper three layers, which means access to decryption. This then allows monitoring of what the malware does, when arriving at an end-user.

Detailed requirement specification

Based on the acquired knowledge concerning the structure of a Honeypot system as well as an understanding of relevant network communication processes, a further specified set of requirements for the creation of a Honeypot is formed. The system will be structured in three modules; Test environment, containment and the combining of analysis and monitoring, where each module consists of the following:

Test environment

- A setup running virtual Android machines, realistically emulated to mimic the behaviour of a generic end user.

Containment

- A firewall able to filter packets and create rulesets to control inbound and outbound traffic.
- A bandwidth limiter with the ability to control the speed at which communication is allowed in the system.

Analysis and monitoring

- Analytical tools with the ability to monitor the traffic of packets and data structure.

For each of the modules -- their purpose, functionality, and structure, will be conducted in the following chapters.

7. Test environment

This chapter will discuss a few things. Firstly, it will be discussed what a virtual environment is. Secondly, it will be explained why such an environment is needed. Lastly, an overview of the process will be given. Virtual environments are easily replaced, which reduces the amount of damage malware can do on the real computer or server.

A virtual environment is an emulated system, on a device that is simultaneously running a base operating system.
This allows several things: Being able to use programs which normally would not be accessible, like using programs made specifically for a mac system while using a windows device. It also allows a user to test out programs on an emulated system before installing it on the base operating system.
This makes it possible for the user to catch some types of malware, before they are at risk of having their personal or credential information stolen. In extension, it is also possible to run tests on a virtual environment, to see what exactly the malware is doing when infiltrating what it perceives to be a device with a specific operating system.

To create a virtual environment, the essential tool is a program that enables the creation of virtual machines, such as "VMware" or "VirtualBox". Of course, it is also necessary to have the installation files of an operating system to run said OS. A practical part of using a virtual machine is that once a usable instance has been created, it is possible to clone that instance so making multiple similar instances is done quite easily. On top of that a virtual machine can take a *snapshot*. A snapshot is a form of safe state meaning that if something goes wrong or if there is virus in the system, the system can return to set safe state and try again

Finally, there would need to be enough excess computing power and hard drive space, to meet the minimum specifications of the operating system being emulated.

The setup process depends on both the Virtualisation tool used and the operating systems settings. But generally, the setup process after booting into the operating system, the normal setup process begins.
However, with android this is not the case. This is due to no official Android .ISO files being available. Therefore the "androidx86"(56) system is used. Androidx86 is essentially an open source x86 port for Android. This means that it can run with any virtualization program with support for x86 operating systems.

The fundamental procedure for setting up a test environment relevant to the project, is to use a software tool such as VirtualBox alongside an Android ISO x86 file. In the next section the available hardware will be discussed and how best to utilise it. It is necessary to know the number of systems that can be set up utilising the hardware available.

7.1 Hardware

This section will describe the fundamental hardware requirements of running a virtual Android machine. Moreover, it will also explore how many virtual machines can be run simultaneously by utilising the capacity of the available hardware.

7.1.1. Requirements for Android

This section will describe requirements to be able to run a single virtual android machine. It will also discuss why it is important to imitate an actual device. This is discussed to increase the likelihood of collecting accurate data.

The Android installation has a low recommended specs for its setup. When setting up the virtual machine it starts up an installation guide, where it asks for how much RAM and capacity it can use. It only recommends dedicating 1024 MB RAM and 8 GB capacity for each virtual machine.

If this recommendation was followed each virtual Android would look different from most current smartphones. As mentioned in the Honeypot section malware does in general take steps towards protecting itself from being observed and analysed. Therefore, presenting false information such as a mismatch between hardware and device model name, would allow malware to avoid effective analysis.

Therefore, creating virtual Androids with these low recommended specifications would not make for an effective test environment. Intruders could potentially become suspicious were they to discover an Android 6.0 device, with only 1GB RAM and 8 GB capacity dedicated to it. Instead there will be dedicated 4 GB RAM and 32 GB capacity (Samsung S8, Nokia 6, etc.), which is a common standard for phones nowadays.

In this section it was decided that the Android virtual machines should have 4 GB RAM and 32 GB capacity dedicated to them so that they resemble current smartphones, which will help make intruders less suspicious of the virtual machines.
The next section will discuss the server's capacity and how best to utilise it.

7.1.2. Our server's capacity

In this section the specifications of the server will be explored, and it will be estimated how many virtual androids it will be able to run simultaneously and as well as a further discussion of how to best resemble a real Android device. This is important for setting up the system, since it will clarify how the server could be set up.

The full server specifications can be seen in appendix 4, but what will be looked at is the server's CPU cores, RAM and HDD capacity, which would be 32 cores at 2.1 GHz, 512 GB RAM and 20 TB

HDD capacity, and since we have to give space to the Ubuntu operating system on the server, which takes at least 2GB RAM and 25 GB capacity according to Ubuntu's wiki (61).

The number of Androids according to the capacity would then at most be 19.975 divided by 32 and according to the RAM it would at most be 510 GB divided by 4. These numbers would be 624.22 and 127.5, rounding down would be 624 and 127.

Despite the vast amount of RAM and HDD capacities available running 127 instances of android with just 32 cores would not be feasible. Assuming that current generation smartphones feature nothing less than dual-core processors and commonly quad-core processors the number of virtual androids would have to be much lower. Setting aside four cores for the underlying Ubuntu OS and allocating just two cores to each virtual android would allow for 14 instances of Android simultaneously.
However, it would also be possible to not dedicate cores to any of the virtual androids. This could solve the problem of underperformance while creating a new problem of peak performance being too high. Optimally the setup would allow for realistic peak performance without having to dedicate cores and underutilize the available cores.

It was found that the server would be able to run 14 instances of android simultaneously while being aware that each instance is going to be performing worse than most current generation smartphones. A smarter solution to the problem of realistic performance is a possible improvement in the future. In the next section the data on each virtual Android is discussed along with a more practical discussion of how to run these virtual Androids.

7.2. Running virtual Androids

This section will discuss relevant considerations for running the virtual Androids. The first of these considerations being how to attain credible user data without compromising actual users. It is also considered how to make this data as credible as possible. Lastly, it is discussed how running the virtual Androids could be automated.

7.2.1. Emulating user data

One particularly important part of creating a credible virtual Android device is to put seemingly legitimate user data on the device. Otherwise, the virtual device will seem suspicious. Therefore, this section will discuss which data should be emulated.

Since the operating system is installed when creating the virtual machine, most of the system already looks like a real Android phone. An ISO-file of a 6.0 Android system will be used. One thing that is needed, is for it to have some extra data on it, so that it doesn't look like a completely new phone, and make intruders suspicious of it being a Honeypot.
To emulate a phone capable of both luring an attacker in while not making him suspicious would be to make sure that each of the emulated machines has either false "NEM-ID" or "false Google-account" or even both.
This works as the lure for the attacker. Since 98 % of our survey told us they have some form of social media on their phone, it would make sense to make false social media accounts as well. Since many Android phones are rooted which is a big potential for data loss it would make sense to make some Android systems rooted and the rest non-rooted.

The phone that will be the prime emulated device is the Samsung Galaxy S7. This is because the phone is released before the use the implementation of newer operative systems than 6.0 Android OS.
The phone came in its factory state with an Android 6.0 OS. This means we can emulate systems that have not had any added security through OS updates since 6.0 without seeming suspicious. Besides this, most newer smartphones have 4 GB ram or more which the Galaxy S7 has.
This allows for a near perfect match between our server and the emulated device yet there is a slight mismatch in the CPU. The servers run on 2,1 GHz where Galaxy S7 runs with a 2,3 GHz CPU. The hope is that the attacker doesn't know or doesn't notice the slight change in CPU.

The chapter focusses on which smartphone is realistically emulated with the hardware available. Furthermore, the chapter focusses on which data should be available on the phone. The next section will discuss how to set up usable fake user data.

7.2.2. Generating and implementing google accounts,

To ensure the realism of the Android virtual machines a google account must be added to it. When setting up an Android phone a google account is a necessity to unlock core functionalities such as the google play store. While tools such as “PVA Creator”(38) claim to have the ability to create multiple Google accounts, they will not be used in this project due to time constraints.

Instead the google accounts will be created manually. The “sensitive data” in the accounts will be generated through “fakenamegenerator.com”.(38)
The personal data generated is used in the google account setup process. As an account is needed for each virtual machine, this gives the project a lot of setup time depending on the number of virtual machines in use. For this project when each account is manually entered it´s put into a spreadsheet with username and password. Allowing the person setting up the Android machines to simply input username and password as given in the spreadsheet.
This also means the signing process on the virtual machines is completely manual, again requiring additional setup time and a substantial amount of human input.
For future development of this system an automation of this process would greatly reduce the amount of work per instance of the emulated Android system.

7.2.3. Automation of emulated machines

A large amount of data and processing of 14 virtual machines gives rise to the problem of incoherent and difficulty managing the virtual machines. The problem can be resolved with an automation of the virtual machines. This would create a stable platform to operate on. The chapter will focus on the basic requirements for an automated system.

To initialise and do a sequence of actions to each virtual machine individually is tedious to do manually. The margin of error is also a factor with manual control. To eliminate these factors a use of an automated system would be preferred. A large amount of data will have to be gathered and in an Ubuntu virtual environment the automated system will have to be adjusted to that.
A certain amount of script coding knowledge is needed to perform a VMware to automatically

start-up and shutdown. Furthermore, a script for opening up Wireshark is needed and then opening up different URL's to get malicious software downloaded. After a given amount of time the emulated machines should be shut down and then restarted from a given point that is deemed safe enough for usage.

The basic lines that are needed for shutting down and starting the VMware up. A guide on how to implement and use scripts in Linux can be found with: (58).

The basic function of an automated system will be the start-up and shutdown. To get scripts to function with Wireshark and URL's with malicious software more time would be needed to dedicate to the subject in order to create an understanding that could implement the scripts. The basic start-up and shut-down of the VMware is the most realistic prospect of the automated systems. This will lay the ground for future endeavours with automated system use.

Time is needed to understand and fully comprehend the use and implementation of scripts in general. When a better understanding of the subject has been obtained it is possible that a system with 14 units could be running autonomously.
This will allow for coherent use compared to manual use. Since the amount of data that would be gathered and the focus of the type of data there is a potential security risk and a need for a contained space that the research can be conducted in is needed.

In this chapter a number of considerations regarding virtualisation have been discussed and hereby given an overview of how to emulate a virtual testing environment. When emulating systems with malicious data a need for containment is necessary to avoid damage if the security is breached.

8. Containment

To effectively use a Honeypot, it is important to be in control of inbound and outbound traffic. Implementing a containment system as a check block sitting between the internet and the rest of the system is a method that will be used.

Depending on the research, this system has to be configured to the circumstances. Meaning, for

the malware to develop in the test environment, it's necessary to allow for some amount of outgoing traffic. A system with no outbound traffic allowed, won't seem luring or it might seem like an obvious Honeypot that gets attackers in and then they immediately leave upon entering.

For the Honeypot to produce usable results, the attacking source has to be let into the test environment, and reveal its methods to access data and communicate it out of the system. This communication should be closely monitored, filtered and limited by a containment system to avoid the malicious code to spread and cause damages. A relatively simple system, capable of the most important elementary tasks could be organized as follows:

- A packet-filtering firewall responsible for rulesets applying to all traffic communicated in the system.
- A bandwidth limiter responsible for limiting the speed of which communication is allowed, to effectively stop any attempts from malicious software to send out large amounts of harmful data.

8.1. IP-tables

IP-tables is a configurable packet-filtering software firewall provided by the Linux Kernel and implemented in all Ubuntu operating systems.(39) It allows for rudimentary control over inbound, outbound and forwarded traffic in a system.
Customization of the firewall is done by defining rulesets. As the firewall is implemented in the Linux Kernel software, it is operated by executing BASH commands in the default Ubuntu terminal.

IP-tables allows a user with root access to edit pre-existing or define new rules, telling the firewall how to treat a packet. These rules are saved in one of the five predefined *chains*:

- PREROUTING: Packets reach this chain before a decision of routing is made, meaning that it is the initial chain that all inbound traffic will reach.
- INPUT: If told so by the PREROUTING chain, the INPUT chain will handle packets send to be locally delivered.

- FORWARD: Packets that have been routed without a direct local address, traverse this chain to be redirected.
- OUTPUT: Packets being sent from the local system will be directed in this chain.
- POSTROUTING: After routing decision is made, the POSTROUTING chain makes for the last decision of executing the delivery.(40)

The chains are built after a sequential principle – as the name indicates. Each rule in a chain will traverse in the sequential order that they have been defined. When defining a new rule, it can be added to the end of the specified chain by using the command –A, or inserted as the first position by using –I.

Consider the following example of an INPUT chain containing rules allowing traffic for certain IP-addresses:

```
User@ubuntu:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  255.255.255.0        anywhere
ACCEPT     all  --  172.16.140.255       anywhere
ACCEPT     all  --  172.16.140.130       anywhere
ACCEPT     all  --  172.17.179.109       anywhere
ACCEPT     all  --  176.9.147.60         anywhere
ACCEPT     all  --  159.20.6.38          anywhere
```

If a packet is routed to the INPUT chain, it will systematically test if the sender matches the IP-addresses listed in the rules from top to bottom. If a new rule is added using the –A command, it will be listed below the already existing rules and effectively be examined as the last option. Only packets that didn't match any of the other IP-addresses will ever reach the last rule of the chain. Opposite, if a new rule is inserted by using the –I command, it will be listed as the top rule and be tested for as the first possible rule.

In conclusion, a packet will continue systematically traversing through the chain until it meets a matching rule. If a matching rule is met, the target command will be executed. Target commands involve either redirection of the packet, acceptance for the packet to move on or dropping packets to be discarded. If the last rule of the chain is reached, the default policy of the chain will be executed.(41)

Figure 16 illustrates how a packet traverses in a chain with default policy DROP, as the IP-tables firewall will be configured for this project. A packet, labelled number 1, approaches the first rule of the chain, it checks if it is a match. In this case, the rule looks for packet number 2, so the packet continues on to the next rule. If this rule checks for packets with the label 1, the traversing packet will be recognised and the policy of the rule will be carried out.

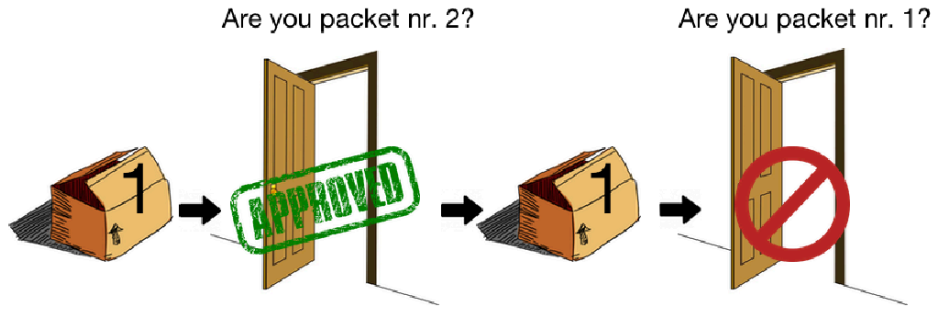


Figure 16: Illustration of a packet's journey through an OUTPUT chain with default policy DROP (Blacklisting)

8.1.1. Configuration of IP-tables firewall

The configuration of a saveable IP-tables script, with the ability to easily customize rulesets of each chain, would have the ability to act as a flexible containment feature. Before configuring chains of rules in IP-tables, it's important to be familiar with the most commonly used commands as listed below.

sudo = Root access of the Ubuntu system.
iptables = Access the iptables files.
-I = Insert one or more rules for the selected chain. (INPUT, OUTPUT or FORWARD)
-A = Add one or more rules to the selected chain. (INPUT, OUTPUT or FORWARD)

-s = Source (address, network name, hostname, etc.)

-L = List (Displaying the rules in a certain chain, OR all chains)

-j = Jump (Says what action is to be executed if the source is reached)

-D = Delete rule/rules

-p = Protocol (TCP, UDP, etc.)

Source: (42)

Requirements for IP-tables script

- Drop any inbound, outbound or forwarded traffic as standard.
- A customizable command string that allows for easy configuration of rules allowing traffic to/from certain IP-addresses.
- A customizable command string that allows for easy configuration of rules allowing traffic to/from certain ports.
- Ability to edit in the chains of rules. (Delete rules, move rules, jump rules)

Step 1 – Setting the default chain policies to DROP.

sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

sudo iptables -P OUTPUT DROP

Step 2 – Allow already established connections to communicate effectively

//Allows already established connections to communicate.

sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

//Allows loopback connections.

sudo iptables -A INPUT -i lo -m -j ACCEPT

//Allows ping connections.

sudo iptables -A INPUT -p icmp -m -j ACCEPT

Step 3 – Create command strings.

//Command string allowing access to a multiplum of certain ports.

```
sudo iptables -A INPUT -p tcp -m multiport --destination-ports
22,25,53,80,443,465,5222,5269,5280,8999:9003 -j ACCEPT
//Command string allowing for TCP traffic to go through a specific ethernet port.
sudo iptables -A INPUT -p tcp --destination-port 110 -i eth0 -j ACCEPT
//Command string allowing INPUT traffic for a certain IP adress.
iptables -I INPUT -s 198.51.100.0 -j DROP
```

Step 4 – Saving the configuration

IP-tables rules are by default set to reset if the system reboots. To save a script of rulesets the command "*iptables-persistent*" from the Ubuntu repositories, can be used.

```
sudo apt-get install iptables-persistent(43)
```

A script configured as described in the steps above can make for a customizable firewall in a basic Honeypot system. If implemented alongside a BASH-based script with the ability to limit upload and download speeds of the network. It can make for a basic construct of a containment system.

8.2. Bandwidth limiting

When building a Honeypot system, a vital question is how to secure the network communication, so the Honeypot system does not act as a gateway for the malware to spread through. This chapter explains the fundamentals of traffic control in Linux while discussing the pros and cons of different methods.

What risks are involved with poor traffic control?

If a Honeypot system is part of a larger network, the rest of the network is at risk of getting infected by malware from the Honeypot test environment. A solution to this problem is to acquire a dedicated network connection. By directing traffic through a dedicated network, the only other place on the network at risk is the ISP, however any potential damage dealt will be at a minimum.

Depending on the available connection to the Honeypot system, it could be a target for a botnet attack.[8] If such an attack is not detected and a DDoS attack is launched throughout the botnet with the Honeypot as an acting part. To avoid being a part of DDoS attacks, limiting the amount and speed of the outgoing bandwidth is a possible solution; This can be done in a Linux based system by implementing Qdisc classes.

What is a Qdisc?

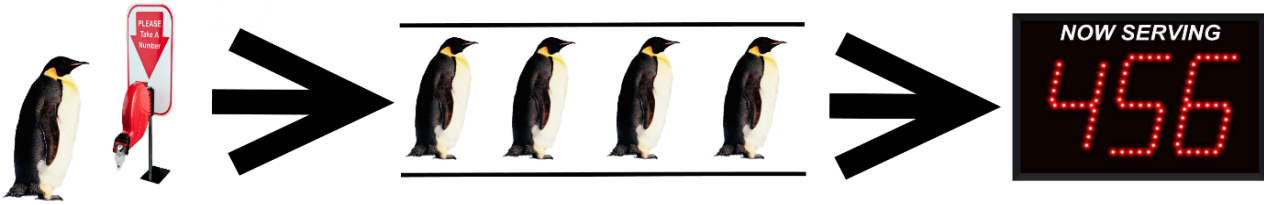


Figure 17: A FIFO (First in First out) system illustrated by penguins

Qdisc is a traffic scheduler built into Linux. By default, it uses the FIFO or first in, first out scheduling system. This means that the first packets to go into the scheduler will be the first to be sent out to their respective destination. Imagine a swarm of penguins (packets) going through a tunnel, but the said tunnel is only broad enough for one penguin. In this case the penguins will come out of the other end in the same order as when they entered, as illustrated in *figure 17.*

In this case a penguin (or packet) takes a number and gets in line. It then gets to leave when it´s number is called. The number calling always happen in the order that the penguins got their number. While there are other options available, they require more configuration of the Qdisc system than what time permits for this project.

To utilize the Qdisc system, classes must be created. A Qdisc class essentially defines the speed at which the packets are served. The simplest way of doing this is by using the HTB or Hierarchical token bucket system. (44)

[8] A botnet can be described as: "*The word Botnet is formed from the words 'robot' and 'network'. Cybercriminals use special Trojan viruses to breach the security of several users' computers, take control of each computer and organise all of the infected machines into a network of 'bots' that the criminal can remotely manage." (64)*

Hierarchical token bucket or HTB is a tool used to limit bandwidth to a Linux client. In most business application this is used to limit a single computer in a network, to stop it saturating the network. An HTB works inside a Qdisc class. By setting a Qdisc class with HTB commands in a client, one can limit the bandwidth used by the said client.

An HTB class is a setup with two parameters:

- Rate
- Ceiling

The rate is the total amount of guaranteed bandwidth a client has available, while a ceiling sets the maximum allowed bandwidth for a client.(44)

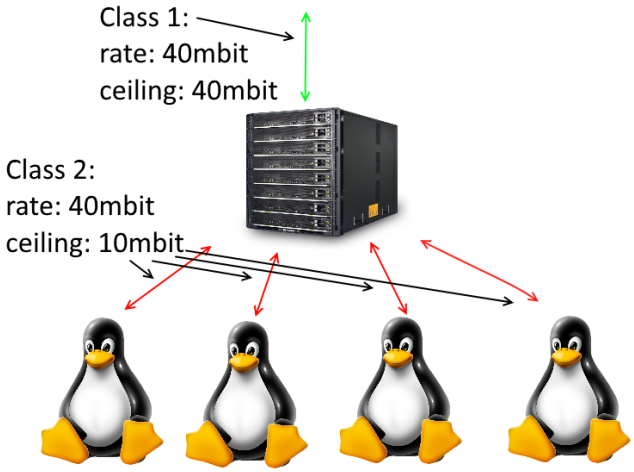


Figure 18: Illustration of a simple HTB system

Illustrated in *figure 18,* each Linux client is connected to a server that has a connection to the internet. In this scenario, each Linux client is limited to a 10mbit connection, ensuring that the total bandwidth cap is never met. In larger more complex systems, several classes with multiple rates and ceilings can be implemented.

A big limitation of this system is that it does not dynamically allocate ceilings. Which means that in larger systems a lot of the bandwidth is woefully underutilized. If only half the clients are active and using the ceiling rate whilst being able to transfer at higher rates. Dynamic allocation of such a system would be preferable.(45)

In this project, instancing the line-up is relatively simple. Because the object is to run several instances of Android and not Linux, the options available are rather limited. While other ways of doing it are achievable, the project uses a simple instance of Ubuntu running a relatively large amount of android emulations at once.

This leaves only one instance of Ubuntu to control, making it a simple task to limit the total amount of bandwidth to the server. Of course, this comes with the opportunity for future improvement of the system, if possible a way of allocating specific classes to each individual instance of android would allow for better overall bandwidth control. However, as it stands the only possibility is to set the total amount of traffic coming out of the Linux installation controlling the virtual machine.

A way of improving bandwidth control is to run multiple instances of Linux inside a server and take each of these as a cluster with a set amount of android system and giving them individual limits. Due to time constraints, this will not be pursued further in this report.

8.2.1. Implementing a Qdisc HTB bandwidth limiter.

When it comes to implementing a Qdisc system with an HTB class, two variables are to be considered. Namely the rate and ceiling. To set up the Qdisc system so it´s in the default Linux terminal the following command is used:

"sudo `tc qdisc add dev (Client name) handle 1: root htb default 11`"

The first command line essentially tells the client that a Qdisc system with HTB controls is going to be made and gives it the handle 1: It also tells that the default HTB class is "11". Therefore, the default Qdisc class is "1:11". Next the actual bandwidth limitations are added:

"`sudo tc class add dev (Client name) parent 1:1 classid 1:11 htb rate (ceil)Mbit burst (Burst)kb`"

Here, the “class” command sets the line to add a function to the “classid 1:11” default class. Then the actual speeds are input to the class. And with those two relatively simple lines of code, a basic outbound bandwidth limiter is made inside the Linux ecosystem.

To make the system more realistically look like a modern smartphone, setting a slightly longer artificial delay of 30-100 ms is considered. As this project has a dedicated fibre optic line, a low ping is expected by default. One can think of the added ping as a gate in the FIFO example that puts a minimum time inside the queue before the penguins can leave. To add the delay the following code is used:

```
”sudo tc qdisc add dev (Client name)  parent 1:11 handle 10: netem delay (delay in ms)ms”
```

Here essentially the same usage works, however the ping is added to the 1:11 class directly and becoming its own 1:10 subclass. This means that it works with the 1:11 class, but can also be used without the parent 1:11 class. Allowing the user to set the delay by changing the class in use by the system to 1:10 instead of 1:11.(45)(44)

8.2.2. VBoxManage safety management

VBoxManage has an incorporated way of limiting the bandwidth of specific groups, which can be created and identified in the program. This way each virtual machine can either have their group, and therefore their own bandwidth limit, or they can be arranged in bigger groups, and then have an overall bandwidth limit.(51)

Examples of common commands used for VBoxManage traffic control:

1) "*VBoxManage bandwidthctl add <name> -- type disk|network –limit <megabytes per second[k|m|g|K|M|G]"*

2) "*VBoxManage bandwidthctl set <name> -- limit <megabytes per second[k|m|g|K|M|G]"*

The first command adds a new bandwidth group, giving it a name and setting the bandwidth limit, either in the default megabytes per second. It's also possible to set another unit, which are

the k, m, g, K, M and G commands, each of these puts the unit to something specific. In the named order it would be kilobits, megabits, gigabits, kilobytes, megabytes, and gigabytes.(51) This could potentially be useful if the need to set up specific bandwidth limits on each individual virtual machine becomes apparent.

8.3. Automation of bandwidth limiting

Bash-commands in Ubuntu can run completed scripts that accept inputs, meaning that variables, such as IP addresses, can be decided by user inputs. For example, take the code used for bandwidth limiting:

```
#!/bin/bash
TotalSpeed=$1"mbit"
BurstSpeed=$2"mbit"
sudo tc qdisc add dev wlp2s0 root handle 1: htb default 1
sudo tc class add dev wlp2s0 parent 1: classid 1:1 htb rate $TotalSpeed ceil $BurstSpeed
```

Firstly, the script start for a bash command is used:

```
#!/bin/bash
```

This gives the console the understanding that a bash script is about to be used. Secondly the two variables needed are inserted. In this case “TotalSpeed” is our Ceiling and “BurstSpeed” is our burst speed:

```
TotalSpeed=$1"mbit"
BurstSpeed=$2"mbit"
```

. To input the variables into the terminal simply type in the first variable followed by the second variable. This is done in the same command line as for where one activates the script. For example:

“. /Scriptname Variable1 Variable 2”

Of course, the more Variables the more inputs are needed from the user. In this case the variables need to be in the HTB or “Hierarchical token bucket” programming language. Therefore, a text defining what type the variable is. In this case it´s mbit. So, in this script the speed is defined in mbit.

Once the variables are defined by user input the script can then be run whilst inserting the variables:

```
sudo tc qdisc add dev wlp2s0 root handle 1: htb default 1
sudo tc class add dev wlp2s0 parent 1: classid 1:1 htb rate $TotalSpeed ceil $BurstSpeed
```

Here the “TotalSpeed” variable is inserted into the htb rate and the “BurstSpeed” variable is inserted into the ceiling.

Different kinds of bandwidth limiting can be used in the system to limit the attackers’ capabilities. A method to control traffic with IP-tables has been found and later could be implemented given more time. Analysing the data is necessary to learn the behavioural patterns of the malware. Therefore, an analysis software is needed and is discussed in the following chapter.

9. Monitoring and analysis

This section will discuss the choice of analysis software used, how to install it and how to use it. This is integral in the creation of a Honeypot since the data collection and analysis is the strength of a Honeypot.

9.1. Choice of software

In this section it will be discussed which analysis tool is favourable, and why that is, and then used in the project.

Cuckoo is very modular due to it being an open source program, meaning that it can be custom tailored to the exact needs of the user. However, it is not the only monitoring and analysis tool available, as an example there is also the program Wireshark. If it would happen that there is not enough time to setup Cuckoo properly, then Wireshark could be used as an alternative since it is relatively simple to get its basic setup going. When multiple android virtual machines are created they will be configured in slightly different ways so that different kinds of data can be collected at the same time.

They will all be created with the same base functionalities so that they have the same starting point. Then they will be exposed to slightly different conditions, as in not all the virtual machines will necessarily be infected with the same virus.

By utilizing cuckoo's ability to provide detailed reports regarding how malware behaves in an android system it will be possible to see if any malware infiltrates any of the virtual android devices and see what happens to them.

It is decided that Cuckoo is the first choice due to its extensive capabilities in analysis. Wireshark is kept in reserve due to its simpler setup process in comparison to Cuckoo. The next section will explore Cuckoo.

9.2. Cuckoo

This chapter will be about Cuckoo, what it is, what it does and creating an installation guide.

Before diving too much into what cuckoo is, one needs to know what it is built upon. Cuckoo is but a name that the inventors of this specific sandbox have given it. A sandbox is a form of virtualization software which can be used as something similar to a testing ground.
It allows processes and programs to run in an isolated virtually created environment, where it is able to do a multitude of things with those applications without posing any threat to others.(46)

There are a multitude of reasons as to why one should use a virtual test environment instead of just using a normal pc or phone. Mainly it is to protect the information already on the pc/phone from being stolen or damaged. The way it works is by creating a virtual machine within the computer, while still being separate from the actual computer. It will appear to be a separate machine when looked at by others over the internet due to it having its own IP address and OS.(47)
Creating a virtual machine requires the machine to have a specific minimal amount of resources allocated from the host machine in order to run. The specific amount of allocated resources is decided by the operating system that is to run on the virtual machine.

Were the virtual machine to be attacked by a hacker or otherwise infected with malware, it would only damage the virtual machine. Moreover, the hacker would not be able to extract any of the information on your actual computer through the virtual machine as it is its own entity.
As such the sandbox is able to protect the data on the host machine. Cuckoo is a sandbox setup within some predetermined parameters and within them there is a high amount of customizability. Cuckoo is as they say on their own website, "the leading open source automated malware analysis system.(48)

It is a program used to analyse malicious code in a safe environment, hence the sandbox part of its name. Cuckoo claims that with the default settings it is able to do the following:(48)

-"Analyse many different malicious files and websites."

- This includes a range of files such as word documents, emails and PDF files. This means that Cuckoo is able to analyse nearly all kinds of files that an average person will have on their computer, as these are the file types that are default on a computer or a smartphone.

-"Trace API calls and general behaviour of files"

- This is where the general behaviour of a file is interpreted and made understandable to anyone.

-"Dump and analyse network traffic"

- Dumping means saving all the traffic data without sorting through it. Network traffic analysis is a process where network traffic is recorded, reviewed and analysed, with the intent of improving the understanding of the traffic and hereby securing better performance and security.

-"Perform advanced memory analysis"

- This is where it analyses the entire virtual machine to check if everything is as it is supposed to be or if there has been tampering with files. This is done by YARA which is used to classify and identify malware.(49)

Cuckoo is being installed within a virtual machine running Ubuntu, because the server that will be used for the actual honeypot runs on Ubuntu as well, so it would be beneficial to know it works exactly as when tested on our own setup.

To install Cuckoo a list of programs and libraries are required. A program that is needed is VirtualBox(51) which is the program the actual emulation is going to happen in. Another program that needs to be present for Cuckoo to work is Python 2.7.(51)

An installation guide has been created and is available in the **appendix** section [6].

Due to inexperience with Python coding, Cuckoo ended up taking too much time to configure properly for the first setup, however it is still expected to be ready for when it should be installed on the server. Since there still was a need for a monitoring tool in the first setup then Wireshark was deemed as an alternative tool for the project.

9.3. Wireshark

To know what Wireshark can do then it would be logical to look for the original, and perhaps, the best place for information about Wireshark, which would of course be the developer webpage itself. (59) To quote the site what Wireshark is:

"Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.

Wireshark is perhaps one of the best open source packet analyzers available today."

Some of the features listed on the developer website are:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.

- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics

An installation guide has been created and available is in the **appendix** section [7].

Wireshark works already from the beginning, however only some data are relevant for the project, which will require an appropriate setup made to monitor for the data we want.(50)

9.3.1. Configurations

Now when Wireshark is installed it needs to be configured to define the functionality it needs to have. Some of Wireshark's functionalities will be explained in this section, and in further detail on the following website.(52)

Wireshark comes with a more comfortable user interface. This interface can be installed on Linux Ubuntu using the command: *sudo apt-get install wireshark-gtk.* This is only after the installations guide has been followed. By pressing Edit and then Preferences you come into a different options setup menu.
Here changes to the interface can be made. Under the same menu, there is a possibility to change toolbars or add toolbars.
The ultimate feature is the capture options. The capture options purpose is to monitor and record traffic one specified network. Currently it is the most basic form of capturing we utilize Wireshark for. All packages will be listed in the "Packet Listing" panel. To gain additional information on a package, a simple click on the package will give the user the possibility to analyse the packet.

Now that the basic usage and configurations of Wireshark have been clarified. Then it would be possible to use it as the monitoring tool for the Honeypot. Now that all of the components for

the Honeypot have been explored, it would be great to have an overview of what has been achieved.

10. End product

To summarise what has been attained in the project this section will compare the goals stated in the requirement specification to the progress made. The goal was to create a Honeypot consisting of the following three elements: A test environment of virtual Android machines, A containment system consisting of an IP-tables firewall and bandwidth limiting and A system with the ability to monitor and filter outbound traffic. As well as collect data on malware executed in the Honeypot. The structure of this system is shown in *figure 19.*

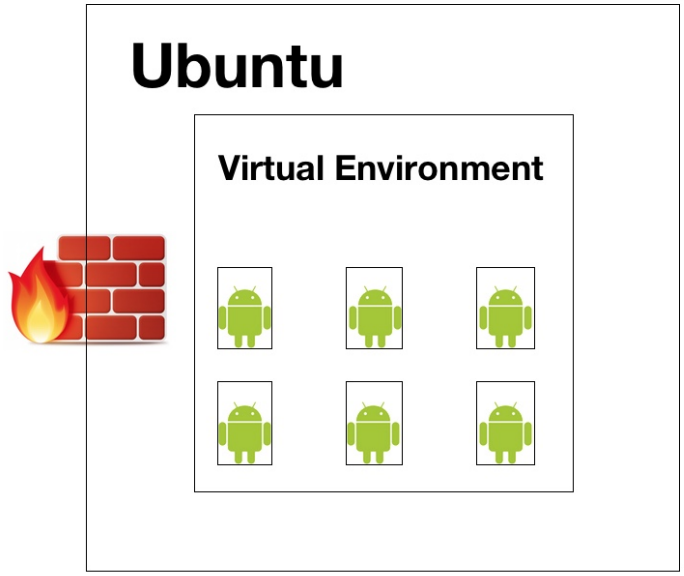


Figure 19: Illustration of end-product

"A test environment of virtual Android machines"

It has been shown that it is fairly easy to create an Android virtual machine from an .iso-file. Moreover, it has also been found that it is possible to copy these virtual machines and restore them to an earlier point. These features, combined with the command line interface offered by VBoxManage, should allow for automation of the handling of virtual machines through bash-scripting.

"A containment system consisting of an IP-tables firewall and bandwidth limiting"

To safely build a Honeypot a containment system is needed. This has been attained by two steps. The first being an IP tables firewall which has proven to be able to approve or deny in- and outbound traffic. As a further safety measure a bandwidth limiter has also been implemented and tested. It has proven to be able to slow any in- and outgoing traffic to the desired transfer rates. These transfer rates can be controlled through VBoxManage limiting individual virtual machines or through the overall bandwidth limiter.

"A system with the ability to monitor and filter outbound traffic"

The project kicked off with the intent of having Cuckoo running in a virtual Ubuntu environment using VirtualBox. After finding guides that show how to install Cuckoo it seemed possible to get it up and running before the end of the project. The installation of the program is complete and the only thing left to do is to configure the program. During the installation, a guide was created which was to be used as part of an automatic installation script. Configuring Cuckoo turned out to be harder than originally anticipated and as so an alternative was needed.

As a replacement for Cuckoo, Wireshark has been found and will be used to analyse files and network traffic instead of Cuckoo. Wireshark is an easy installation in comparison to Cuckoo as it comes in a complete package upon download. The basic functionality is available immediately after installation, however it is capable of more advanced packaged filtering if configured to do so.

"Research if a predetermined malware type, can extract sensitive data from an Android device emulated to resemble a generic Android user. "

It has been found in the theory section about malware that several different malware types can steal personal data. It has also been found that the type of data compromised depends on the exact type of spyware in question.

Combining the three Honeypot elements in a single functional implementation has not been done yet. Therefore, no data has been captured at the current stage. However, it is believed that the necessary elements are in place.

11. Analysis and reflection

This section will analyse and reflect on the choices made in terms of software and reflect upon possible future improvement. To analyse the software used one must first identify what a Honeypot system ideally should be.

Ideally a Honeypot system should be versatile. Meaning that it allows for a multitude of applications, including but not limited to other research projects. It must therefore have basic functionality such as traffic monitoring and a configurable firewall whilst being customizable.
To achieve the most amount of progress in the project, the virtualisation and monitoring are done through third party software (VirtualBox, Cuckoo and Wireshark). Whilst the firewall and bandwidth limiter are made through coding scripts in the programming language “bash” so it works in the Ubuntu terminal.

To assess the possibilities for improvement, the first logical standpoint is to see where the Honeypot is the most limited in its current implementation.
Firstly, all the system is done through a mix of different inputs, from googles account generator to the Linux terminal, resulting in a poor user experience.
This can be solved through the implementation of a GUI that incorporates the functions used in the Honeypot system.
The honeypot systems implementation of firewall and bandwidth limiter are simple, so further improving on these aspects by for example bandwidth limiting each android installation would be a good improvement.
Another improvement is being able to emulate the user input in the Android systems, to give a more realistic “user”.
In short there are many possibilities for improving the system. Of course, it´s only relevant to improve the system in accordance with requirements for future research projects.
So, while a base has been established, there are many possibilities for expanding and improving the Honeypot in the future.

According to the graph shown in *figure 10*, the sale of smartphones has exceeded the sale of computers, and hasn't shown to be slowing down so far. When looking at the sales of computers, and considering home computers entered the market in 1977(53) and has stabilised in the 2010'. Then it could maybe be expected of smartphone sales to settle and stabilise in 5 or 10 years. Symantec's report showed an increasing amount of attacks on smartphones but the usage of different types of malware has been beginning to stagnate. This could indicate that hackers are comfortable with current malware types or defence against malware is beginning to catch up. In worst-case-scenario it could be that the current defence against malware is failing to detect new malware types. When looking back, problems stated with security was reported in 2011 with the US-CERT and in 2015 with senior analyst Nathan Collier and persist in 2017.[9] When looking at PCs, which have been around for approximately 40 years, malware has been an issue almost since the beginning and is still a persistent problem. It is possible as long as the technology exists, hackers and malware will accompany it. If malware is considered a disease, it is a disease where the symptoms are treatable. However, the disease will continue to exist and develop and the combat of it will continue.

12. Conclusion

This study aims to enlighten issues regarding cybercrime. The conducted research gives rise to concerns about smartphone security and presents a versatile Honeypot system to be an adaptive countermeasure.

However, by acting as a potential research station for studies about malware targeting Android systems, it can be a meaningful part of acquiring greater understanding of the attacking sources and improvements of cybersecurity measures.

The global market for digital devices indicates smartphones running the Android OS to make up for a majority of sales (60) and if seen in context with the accessibility to large amounts of sensi-

[9] As described in section 4.2.4.

tive data stored on these devices, it makes for a high valuable target of cybercrime. To further support this statement, a stakeholder analysis is made to specify the focus of the project.

To collect information on types of data stored on a typical smartphone, a survey is made to generate knowledge about user patterns. Results from which, presents potential threats for Android devices as well as gives insight in the types of sensitive data stored on smartphones. The problem analysis therefore concludes, a need of research concerning malware targeting Android smartphones containing relevant types of sensitive data.

Based on this, it is ascertained to create a Honeypot system. For such a system to produce valid results the following measures are necessary:
- Realistic emulation of generic Android users.
- Emulation of hardware specifications similar to a real Android phone.
- The ability to contain malicious traffic and prevent damages to the connected network.
- Effective analytical tools to monitor and analyse network communication.

The system is designed in a modular manner, which allows for further development and customizability. The study lays the foundation for a basic Honeypot system, that fulfil the set requirements for a versatile and expandable research station. The basis stands to be implemented in future development projects, as a structured foundation able to be customized and expanded upon quickly, relative to current cybercrime threats.

13. References

1. *Evolution in the World of Cyber Crime [Internet]. InfoSec Resources. 2017 [cited 17 December 2017]. Available from: http://resources.infosecinstitute.com/evolution-in-the-world-of-cyber-crime/*
2. *'Petya' Ransomware Outbreak Goes Global — Krebs on Security [Internet]. Krebsonsecurity.com. 2017 [cited 17 December 2017]. Available from: https://krebsonsecurity.com/2017/06/petya-ransomware-outbreak-goes-global/*
3. *Fruhlinger J. Petya ransomware and NotPetya malware: What you need to know now [Internet]. CSO Online. 2017 [cited 17 December 2017]. Available from: https://www.csoonline.com/article/3233210/ransomware/petya-ransomware-and-notpetya-malware-what-you-need-to-know-now.html*
4. *Novet J. Shipping company Maersk says June cyberattack could cost it up to $300 million [Internet]. CNBC. 2017 [cited 17 December 2017]. Available from: https://www.cnbc.com/2017/08/16/maersk-says-notpetya-cyberattack-could-cost-300-million.html*
5. *Cyberwarfare [Internet]. En.wikipedia.org. 2017 [cited 17 December 2017]. Available from: https://en.wikipedia.org/wiki/Cyberwarfare*
6. *Cite a Website - Cite This For Me [Internet]. Fe-ddis.dk. 2017 [cited 17 December 2017]. Available from: https://fe-ddis.dk/cfcs/CFCSDocuments/Cybertruslen%20mod%20Danmark.pdf*
7. *Cohen E. 35% of Users Have Weak Passwords; the Other 65% can be Cracked [Internet]. Blog.preempt.com. 2017 [cited 17 December 2017]. Available from: https://blog.preempt.com/weak-passwords*
8. *NemID [Internet]. Nemid.nu. 2017 [cited 17 December 2017]. Available from: https://www.nemid.nu/dk-da/*
9. *Loiborg C. Populær app hjælper med at bryde NemID-regler: 'Folk gør det alligevel' [Internet]. Version2. 2017 [cited 17 December 2017]. Available from: https://www.version2.dk/artikel/populaer-app-hjaelper-med-bryde-nemid-betingelser-folk-goer-det-alligevel-52323*

10. The Tor Project I. Download Tor [Internet]. Torproject.org. 2017 [cited 17 December 2017]. Available from: https://www.torproject.org/download/download.html.en

11. How To Access Dark Web Anonymously 10 Step Guide (with Pictures) [Internet]. Dark Web News. 2017 [cited 17 December 2017]. Available from: https://darkwebnews.com/help-advice/access-dark-web/

12. Deep Web Scam Sites [Internet]. Dark Web News. 2017 [cited 17 December 2017]. Available from: https://darkwebnews.com/deep-web/deep-web-scam-sites/

13. Dark web - ordering cloned credit cards [Internet]. YouTube. 2017 [cited 17 December 2017]. Available from: https://www.youtube.com/watch?v=opz20ywjE44

14. VanDam L. [INFOGRAPHIC] Introducing The Psychology of Passwords - The LastPass Blog [Internet]. The LastPass Blog. 2017 [cited 17 December 2017]. Available from: https://blog.lastpass.com/2016/09/infographic-introducing-the-psychology-of-passwords.html/

15. Cite a Website - Cite This For Me [Internet]. 1.bp.blogspot.com. 2017 [cited 17 December 2017]. Available from: https://1.bp.blogspot.com/-LY7pI45tMq0/V5r_XV083RI/AAAAAAAAo6M/MivILg0_4Vs7UgLKZJqM5vhvYujQCCcpgCLcB/s1600/best-password-manager-software.png

16. Ng C, Ng C. Why A Honeypot Is Not A Comprehensive Security Solution [Internet]. Varonis Blog. 2017 [cited 17 December 2017]. Available from: https://blog.varonis.com/why-a-honeypot-is-not-a-comprehensive-security-solution/

17. 2017 G. Mobile OS market share 2017 | Statista [Internet]. Statista. 2017 [cited 17 December 2017]. Available from: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

18. Popularity of mobile banking apps makes them cybercrime targets: McAfee | Mobile Marketer [Internet]. Mobilemarketer.com. 2017 [cited 17 December 2017]. Available from: https://www.mobilemarketer.com/ex/mobilemarketer/cms/news/research/11812.html

19. Is Your Security Policy Holding Your Company Back? [Internet]. Business Value Exchange (BVEx). 2017 [cited 17 December 2017]. Available from: https://businessvalueexchange.com/blog/2013/09/16/security-policy-holding-company-back/

20. Mearian L. Android vs iOS security: Which is better? [Internet]. Computerworld. 2017 [cited 17 December 2017]. Available from: https://www.computerworld.com/article/3213388/mobile-wireless/android-vs-ios-security-which-is-better.html

21. Taylor C. Smartphone Sales Overtake PCs for the First Time [STUDY] [Internet]. Mashable. 2017 [cited 17 December 2017]. Available from: http://mashable.com/2012/02/03/smartphone-sales-overtake-pcs/#BhTZJcZcY8qN

22. Arthur C. Samsung confirmed as smartphone chief as growth continues, says IDC [Internet]. the Guardian. 2017 [cited 17 December 2017]. Available from: https://www.theguardian.com/technology/2011/nov/03/q3-2011-smartphone-growth-continues

23. Cite a Website - Cite This For Me [Internet]. Us-cert.gov. 2017 [cited 17 December 2017]. Available from: https://www.us-cert.gov/sites/default/files/publications/cyber_threats-to_mobile_phones.pdf

24. Zamora W, Zamora W. Top 10 ways to secure your mobile phone - Malwarebytes Labs [Internet]. Malwarebytes Labs. 2017 [cited 17 December 2017]. Available from: https://blog.malwarebytes.com/101/2016/09/top-10-ways-to-secure-your-mobile-phone/

25. Cite a Website - Cite This For Me [Internet]. Symantec.com. 2017 [cited 17 December 2017]. Available from: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

26. Over 27.44% Users Root Their Phone(s) In Order To Remove Built-In Apps, Are You One Of Them? | Androidheadlines.com [Internet]. AndroidHeadlines.com |. 2017 [cited 17 December 2017]. Available from: https://www.androidheadlines.com/2014/11/50-users-root-phones-order-remove-built-apps-one.html

27. billions) N. Number of smartphone users worldwide 2014-2020 | Statista [Internet]. Statista. 2017 [cited 17 December 2017]. Available from: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

28. malware | Definition of malware in English by Oxford Dictionaries [Internet]. Oxford Dictionaries | English. 2017 [cited 17 December 2017]. Available from: https://en.oxforddictionaries.com/definition/malware

29. 5 ways to Prevent Mobile Phishing – Safe and Savvy Blog by F-Secure [Internet]. Safeandsavvy.f-secure.com. 2017 [cited 17 December 2017]. Available from: https://safeandsavvy.f-secure.com/2011/04/19/prevent-mobile-phishing/

30. Phishing - Hvad er phishing - hvad gør jeg ved phishing? - Data-teknik [Internet]. Data-teknik. 2017 [cited 17 December 2017]. Available from: http://data-teknik.eu/phishing/

31. 12+ Types of Malware Explanied: Scariest and Dangerous! [Internet]. MalwareFox. 2017 [cited 17 December 2017]. Available from: https://www.malwarefox.com/malware-types/

32. Zetter K, Zetter K, Barrett B, Newman L, Greenberg A, Graff G et al. Hacker Lexicon: What Are White Hat, Gray Hat, and Black Hat Hackers? [Internet]. WIRED. 2017 [cited 17 December 2017]. Available from: https://www.wired.com/2016/04/hacker-lexicon-white-hat-gray-hat-black-hat-hackers/

33. Tough Challenges in Cybersecurity Ethics [Internet]. Security Intelligence. 2017 [cited 17 December 2017]. Available from: https://securityintelligence.com/tough-challenges-cybersecurity-ethics/

34. Zetter K. Hacker's Tweet Reignites Ugly Battle Over Security Holes [Internet]. WIRED. 2017 [cited 17 December 2017]. Available from: https://www.wired.com/2015/04/twitter-plane-chris-roberts-security-reasearch-cold-war/

35. Cite a Website - Cite This For Me [Internet]. Pdf.textfiles.com. 2017 [cited 17 December 2017]. Available from: http://pdf.textfiles.com/security/palmer.pdf

36. John Harrison S. Honeypots: The sweet spot in network security [Internet]. Computerworld. 2017 [cited 17 December 2017]. Available from:

https://www.computerworld.com/article/2573345/security0/honeypots--the-sweet-spot-in-network-security.html

37. A Practical Guide to Honeypots [Internet]. Cse.wustl.edu. 2017 [cited 17 December 2017]. Available from: https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html#sec1

38. PVA Creator [Internet]. Pvacreator.com. 2017 [cited 17 December 2017]. Available from: http://www.pvacreator.com

39. Get a whole new identity at the Fake Name Generator [Internet]. Fakenamegenerator.com. 2017 [cited 17 December 2017]. Available from: http://www.fakenamegenerator.com

40. IptablesHowTo - Community Help Wiki [Internet]. Help.ubuntu.com. 2017 [cited 17 December 2017]. Available from: https://help.ubuntu.com/community/IptablesHowTo

41. Iptables [Internet]. En.wikipedia.org. 2017 [cited 17 December 2017]. Available from: https://en.wikipedia.org/wiki/Iptables

42. [Internet]. 2017 [cited 18 December 2017]. Available from: http://www.thegeekstuff.com/2011/06/iptables-rules-examples/

43. How To Set Up a Firewall Using Iptables on Ubuntu 14.04 | DigitalOcean [Internet]. Digital-ocean.com. 2017 [cited 18 December 2017]. Available from: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-iptables-on-ubuntu-14-04

44. Classless Queuing Disciplines (qdiscs) [Internet]. Tldp.org. 2017 [cited 18 December 2017]. Available from: http://tldp.org/HOWTO/Traffic-Control-HOWTO/classless-qdiscs.html

45. HTB manual - user guide [Internet]. Luxik.cdi.cz. 2017 [cited 18 December 2017]. Available from: http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm

46. Geier E. How to Keep Your PC Safe With Sandboxing [Internet]. PCWorld. 2017 [cited 18 December 2017]. Available from: https://www.pcworld.com/article/247416/how_to_keep_your_pc_safe_with_sandboxing.html

47. [Internet]. 2017 [cited 18 December 2017]. Available from: https://www.quora.com/Do-virtual-machines-have-different-IP-addresses

48. Cuckoo Sandbox - Automated Malware Analysis [Internet]. Cuckoosandbox.org. 2017 [cited 18 December 2017]. Available from: https://www.cuckoosandbox.org/

49. Welcome to YARA's documentation! — yara 3.5.0 documentation [Internet]. Yara.readthedocs.io. 2017 [cited 18 December 2017]. Available from: http://yara.readthedocs.io/en/v3.5.0/index.html

50. How to Use Wireshark to Capture, Filter and Inspect Packets [Internet]. Howtogeek.com. 2017 [cited 18 December 2017]. Available from: https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/

51. Oracle VM VirtualBox [Internet]. Virtualbox.org. 2017 [cited 18 December 2017]. Available from: https://www.virtualbox.org/

52. Congleton N. Basic of network protocol analyzer Wireshark On Linux - LinuxConfig.org [Internet]. Linuxconfig.org. 2017 [cited 18 December 2017]. Available from: https://linuxconfig.org/basic-of-network-protocol-analyzer-wireshark-on-linux

53. Home computer [Internet]. En.wikipedia.org. 2017 [cited 18 December 2017]. Available from: https://en.wikipedia.org/wiki/Home_computer

54. Marr B. The 7 Most Data-Rich Companies In The World? [Internet]. Linkedin. 2017 [cited 19 December 2017]. Available from: https://www.linkedin.com/pulse/7-most-data-rich-companies-world-bernard-marr

55. Data Breach Statistics by Year, Industry, More - Breach Level Index [Internet]. Breach Level Index. 2017 [cited 19 December 2017]. Available from: http://breachlevelindex.com

56. Android-x86 - Porting Android to x86 [Internet]. Android-x86.org. 2017 [cited 19 December 2017]. Available from: http://www.android-x86.org/

57. The Copenhagen Post - Danish News in English [Internet]. Cphpost.dk. 2017 [cited 19 December 2017]. Available from: http://cphpost.dk/news/danes-personal-information-is-cheap-and-easy-to-buy-online.html

58. Cánepa G, Posts V. Using Shell Scripting to Automate Linux System Maintenance Tasks - Part 4 [Internet]. Tecmint.com. 2017 [cited 19 December 2017]. Available from:

https://www.tecmint.com/using-shell-script-to-automate-linux-system-maintenance-tasks/

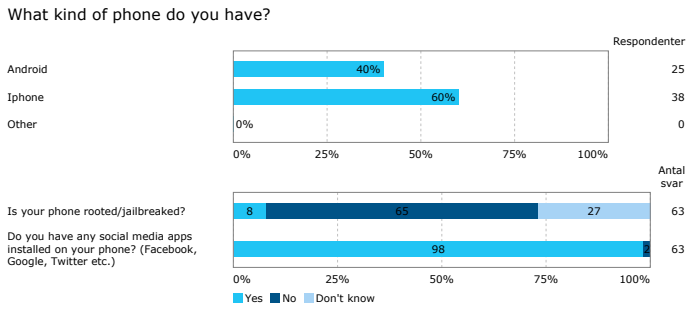
59. Chapter1.Introduction [Internet]. Wireshark.org. 2017 [cited 19 December 2017]. Available from: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html

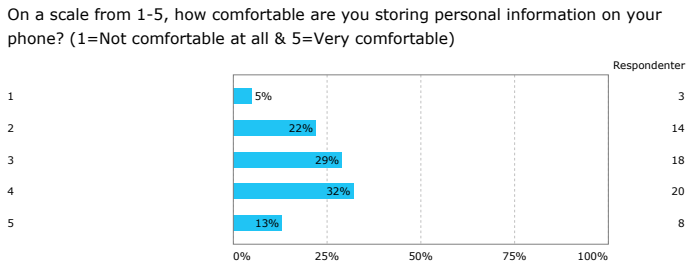
60. Booton J. The rise and fall of the PC in one chart [Internet]. MarketWatch. 2017 [cited 20. December 2017]. Available from: https://www.marketwatch.com/story/one-chart-shows-how-mobile-has-crushed-pcs-2016-04-20

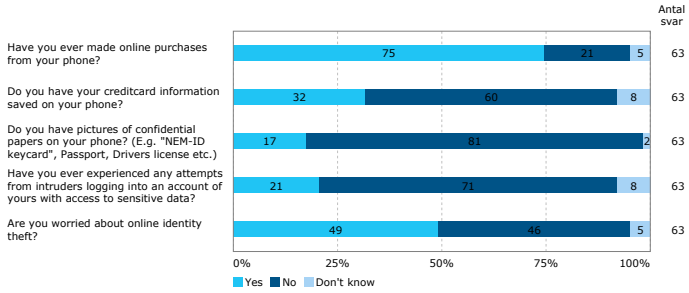
61. Installation/SystemRequirements – Community Help Wiki [Internet]. Help.ubuntu.com. 2017 [cited 20. December 2017]. Available from: https://help.ubuntu.com/community/Installation/SystemRequirements

Appendix

Appendix 1: Results from user survey







Appendix 2: The User survey

Question 1

What kind of phone do you have?

Iphone 60% (38 users)

Android 40% (25 users)

Question 2.1

Is your phone rooted/jailbreaked?

Yes 8% No 65% Don't know 27% (63 users)

Question 2.2

Do you have any social media apps installed on your phone (Facebook, Google, Twitter etc.)

Yes 98% no 2% (63 users)

Question 3

On a scale from 1-5, how comfortable are you storing personal information on your phone? (1=Not comfortable at all and 5=Very comfortable).

1. 5% (3 users)
2. 22% (14 users)
3. 29% (18 users)
4. 32% (20 users)
5. 13% (8 users)

Question 4.1

Have you ever made online purchases from your phone?

Yes 75% No 21% Don't know 5% (63 users)

Question 4.2

Do you have your creditcard information saved on your phone?

Yes 32% No 60% Don't know 8% (63 users)

Question 4.3

Do you have pictures of confidential papers on your phone? (E.g. "NEM-ID keycard", Passport, Driver license etc.)

Yes 17 % No 81% 2% Don't know (63 users)

Question 4.4 Have you ever experienced any attempts from intruders logging into an account of yours with access to sensitive data?

Yes 21 % No 71% Don’t know 8% (63 users)

Question 4.5

Are you worried about online identity theft?

Yes 49% No 46% Don't know 5% (53 users)

Appendix 3: Malware types

The invisible man

The Invisible man malware targets Android's accessibility services. It places an overlay on your legitimate banking apps. This in turn makes the malware act as a keylogger which gets the victim's login details, when they try to access their banking account.

The malware can work on even fully updated Android devices and disguised itself as a fake flash player download.

https://www.theregister.co.uk/2017/08/02/banking_android_malware_in_uk/

Gooligan

The businessinsider provides information about a cybersecurity company Check Point which discovered a type of malware called Gooligan. Gooligan malware targets Android devices and steal email addresses. The malware targets apps such as Gmail, Drive and Photos. Check Point reported that back in July 2016 approximately 85 million Android devices were infected with another type of malware. The malware generated 300.000 dollars in ad revenue every month. And Check Point reported in August that Google AdSense was targeted by hackers to steal Android Users banking data.

The fight against malware is ever ongoing and people are advised to inform themselves and steps are being taken to furthermore secure users from malicious software. http://www.businessinsider.com/how-to-check-if-your-google-account-is-infected-with-malware-2016-11?r=US&IR=T&IR=T

BankBot

A recent discovery by Czech cybersecurity firms Avast and ESET and SfyLab based in Amsterdam found that a malicious software known as BankBot has infected thousands of Android users. The apps infected with this malware were simple flashlight apps. The malware then waited two hours before posing itself as

a system required update which then allowed the malware to gain administrative privileges. Then BankBot waited for the users to enter their details in their bankapps which were immediately shared with the attacker.

Appendix 4: Server hardware specifications

Komponenter

1	PowerEdge R740/R740XD Motherboard
1	Intel Xeon Gold 6130 2.1G, 16C/32T, 10.4GT/s 2UPI, 22M Cache, Turbo, HT (125W) DDR4-2666
1	Group Manager, Enabled
1	Unique Random Password
1	Chassis with up to 8 x 3.5" SAS/SATA Hard Drives for 2CPU Configuration
1	PowerEdge 2U Standard Bezel
1	Riser Config 2, 3 x8, 1 x16 slots
1	PowerEdge R740 Shipping Material
1	No Quick Sync
1	Dell EMC Luggage Tag
1	Performance Optimized
1	2667MT/s RDIMMs
16	32GB RDIMM, 2667MT/s, Dual Rank
1	Intel Xeon Gold 6130 2.1G, 16C/32T, 10.4GT/s 2UPI, 22M Cache, Turbo, HT (125W) DDR4-2666
1	iDRAC9,Enterprise
2	200GB SSD SATA Mix Use 6Gbps 512n 2.5in Hot-plug Drive,3.5in HYB CARR, Hawk-M4E,3 DWPD
1	10TB 7.2K RPM SATA 6Gbps 512e 3.5in Hot-plug Hard Drive
1	Dell 1.6TB, NVMe, Mixed Use Express Flash, HHHL AIC, PM1725a, DIB
1	PERC H330+ RAID Controller, Adapter, Low Profile
2	Standard 1U Heatsink
1	No Internal Optical Drive
1	Dual, Hot-plug, Redundant Power Supply (1+1), 750W
2	C13 to C14, PDU Style, 10 AMP, 6.5 Feet (2m), Power Cord
1	No Trusted Platform Module
1	Order Configuration Shipbox Label (PO Number, Ship Date, Model, Processor Speed, HDD Size, RAM)
1	Intel X710 Quad Port 10Gb DA/SFP+ Ethernet, Network Daughter Card
1	Power Saving Dell Active Power Controller
1	ReadyRails Sliding Rails With Cable Management Arm
1	Unconfigured RAID
1	OpenManage Essentials, Server Configuration Management
1	Thank you for choosing Dell ProSupport Plus. For Tech Support, visit http://www.dell.com/contactdell

Software

1	6 Standard Fans for R740/740XD
1	No Operating System
1	No Systems Documentation, No OpenManage DVD Kit

Service

1	Basic Deployment Dell Server R Series 1U/2U
1	Base Warranty
1	3Yr Basic Warranty - Next Business Day - Minimum Warranty
1	5Yr ProSupport Plus and Next Business Day Onsite Service

Appendix 5: Extended explanation of stakeholders

1. **Generic end-users of Android devices**

The project has potential to increase cybersecurity on Android devices and effectively improve the system's ability to keep sensitive information safe from outside attacks. Such improvements are expected to make users more comfortable storing sensitive information on their devices. Increasing the amount of personal sensitive information, could lead to a more customized and personal user experience result in improvements of the usability of the products. The actual users of Android devices, could supply us with information about what types of sensitive data they have stored, and where. From this, we could specify our analysis to look for only relevant malware, and emulate android devices to be near an actual true end user.

2. **Cybersecurity researchers**

Professional researchers within the field of cybersecurity, could be involved in the project as "expert sources". As an example, an expert interview with Johnny[10] could be carried out, to help determine the priorities of the technical parts when building a HoneyPot system, as he has years of experience doing so. The result of the project would not directly be beneficial to him.

3. **Companies/organizations of any size and type using the Android OS to store sensitive data**

Any company or organization that depend on data being stored securely on Android devices, could be positively influenced by an increase in the IT-security of these devices. It would lead to fewer successful attacks against the companies' digital services, and for the users of the IT-systems to behave more freely and for the company to take better advantage of digitally stored data.

4. **The Danish Centre of Cybersecurity (CFCS)**

The Danish Centre of Cybersecurity is a governmental instance, responsible for handling national concerns as IT-security authority. Much like the professional researchers in the field of cybercrime, CFCS could contribute to the project with their experience in the field, and supply information that could help us, in the process of determining what security threats to focus our research around.

[10] Who is Johnny?

5. **The Danish police's department of cybercrime**

The department of cybersecurity in the Danish police force would, much as CFCS, be able to supply us with important information. Assuming that they are in the possession of information about the most recent and most frequently happening attacks against private persons as well as companies and organizations, we would be able to correct the focus of our project more specifically if they share some of this information with us.

6. **Google offices**

Google offices, could be an important part in the process of emulating actual Android users. Almost all Android devices are connected to a Google-account, and therefore we need our virtual machines to be as well. If not, a potential hacker intruding the system, would see through the HoneyPot instantly. Google could supply us with a number of fake google accounts to use for our project. On the other hand, the end product of the project could be beneficial for Google as well. As mentioned, google is a very well represented part of most Android devices, and more effective security of these, could bring along increased numbers of users and have a positive influence on their business.

7. **Manufacturer of smartphones running the Android OS**

Assuming that a part of the world's population is worrying about online theft of their personal data, and therefore chooses not to buy a smartphone, it can be argued that IT-security improvements would have positive influence on the market of smartphones.

8. **Developers of Android Apps**

Developers of Android Apps, would be able to take advantage of a more secure system, as their users would be more comfortable storing more sensitive data. The user experience could be customized more specifically to the individual user, and as a result, the overall user satisfaction of the apps could be increased. The developers could be beneficial to our project, by supplying us with information on how they keep their user data stored and secure, and what attacks they have been impacted by most frequently.

9. **Banks & transaction apps (Mobile pay, Swipp, Transferwise etc.)**

Banks offering banking apps for their customers and companies behind transaction apps, could benefit greatly from their customers information being more secure. Greater possibilities for easy transactions and fewer successful hacking attacks of sensitive customer data, could be a result from the project.

Appendix 6

// setup for python libraries install

Sudo apt-get install python python-pip python-dev libffi-dev libssl-dev
Sudo apt-get install python-virtualenv python-setuptools
Sudo apt-get install libjpeg-dev zlib1g-dev swig

// these are the software packages from the apt repositories which are required to get cuckoo to install and run properly

// now to installing the Django-based Web Interface

sudo apt-get install mongodb

// now install PostgreSQL as database

sudo apt-get install postgresql libpq-dev

//KVM as machinery module

sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils python-libvirt

// now a server pack which is properly not needed

sudo pip install XenAPI

// Now to downloading Virtualization Software

wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -

// (Here i needed to fix some broken packages, and i used)
// sudo apt-get update

// to install VirtuaBox

sudo apt-get install virtualbox-5.1

// now to installing the tcpdump. where networks activity is being dumped during malware execution.

sudo apt-get install tcpdump apparmor-utils
sudo aa-disable /usr/sbin/tcpdump

// not sure if it is needed to disable aa because it is only relevant if there is used CWD

// to make sure that cuckoo does not have admin/sudo privileges the following command is used

sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump

// Installing the extra software "Volatility"

git clone https://github.com/volatilityfoundation/volatility.git

// "git" is not install so that will have to be done

sudo apt install git

// now the "git" command works, so now the previous command can be run.

git clone https://github.com/volatilityfoundation/volatility.git

---

// Not sure if more libraries are needed for "Volatility" but will skip for now

---

// Now M2Crypto is needed. It is not known what it does because there is no description.
// It requires "swig" to install. It was installed, but here it is anyway.

sudo apt-get install swig
sudo pip install m2crypto==0.24.07

//Now for adding a user on the virtual machine for cuckoo use.

Sudo usermod -a -G vboxuers cuckoo

// It might already be there but doing it anyway it is just an insurance.

//Now for installing the actual cuckoo program

virtualenv venv
. venv/bin/activate
sudo pip install -U pip setuptools
sudo pip install -U cuckoo

// This is what is needed to get Cuckoo installed
// Run Cuckoo
cuckoo

//It can take a long time to start up, so have patience

cuckoo community

Appendix 7

Installation guide

Since we use Linux on our virtual system it is optimal to follow guides that are specifically focused on Linux. In order to install Wireshark, the following commands are used: (https://askubuntu.com/questions/700712/how-to-install-wireshark The answer is from Thusitha Sumanadasa)

- Step 1: Adding the official PPA then we go to terminal by pressing ctrl+alt+t and typing the command: *sudo add-apt-repository ppa:wireshark-dev/stable*
- Step 2: updating the repository is necessary. This is done with the command:
  sudo *apt-get update*
- Step 3: install newest version of Wireshark with the command:
  sudo apt-get install wireshark
- Step 4: start up Wireshark with the command:
  sudo *wireshark*
- Step 5 (if error couldn't run /usr/bin/dumpcap in child process: Permission denied) Go to the terminal again and issue the following command:
  sudo *dpkg-reconfigure wireshark-common*

When the question about continue pops up, say YES by pressing y and hit enter. Then a Wireshark-group should be made and users can now be added. This is done with:
sudo *adduser $USER wireshark*

After restarting the virtual machine and opening Wireshark it should work, and now it just needs to be configured to look for whichever kind of data is relevant.