Problem description

Many software development organizations have launched public bug bounty programs, where users can submit bugs they discover, and be rewarded commensurate with the severity of the bug. Bug bounties have been identified as a software security activity in the BSIMM, and there are several commercial operators that can provide a bug bounty program as a service. The candidate must survey the current state of the art in bug bounty programs, and identify organizations that have implemented such programs, and that are willing to be interviewed about their experiences. The assignment will also interview acknowledged security researches that have participated in bug bounty programs. The candidate should then design a new external or internal bug bounty program suitable for a Norwegian context that can be implemented with a partner organization that currently has no such program. The work will be carried out from Oslo, and weekly supervision meetings will be held over Skype.

Abstract

In a world where hackers take companies' computer systems as hostage in exchange for ransom, and authorities threaten with fines if personal information ever get into the wrong hands, more and more resources are spent on securing computer systems. Companies have begun to reserve more room in their budgets for information security. However, not all companies have the required resources to secure their system. Over 99% of all companies in Norway are small to medium-sized businesses (SMB), and these companies do not always have the required budget to pay for the standard security solutions that are available today.

A potential alternative to standard security solutions that has had major growth in recent years is securing a company's information systems through bug bounty programs (BBP). This is a process where companies invite hackers to research vulnerabilities on their system, with monetary rewards being given in exchange for these vulnerabilities. This master's thesis aims to explore this alternative, as a cheaper but still effective way for Norwegian SMBs to increase the security of their information systems and services, and still remain a business that can make a profit.

Limited research has been done on bug bounty programs as a viable method for security, and even less has been done with the Norwegian market in mind. In this project, a review was performed on research literature concerning bug bounty programs. There was not much research available, and the good attributes were found in existing bug bounty programs.

Four semi-structured interviews were then held with with people from the bug bounty program community in Norway. Two Norwegian companies that host their own bug bounty programs, gave insight in how a successful bug bounty programs functions in Norway. Furthermore, two bug bounty hunters gave insight in expectations, experiences and impressions the people sitting on the other side has.

The findings in this thesis indicate that if an SMB can afford traditional penetration testing, spending the money on that is most likely the best option, where as companies of a larger size can purchase Bug Bounty as a Service (BBaaS) reduce the expenses made for each vulnerability uncovered. However, the thesis also propose a structure for a bug bounty program that Norwegian SMBs can try to implement for their systems, if penetration testing is not an option. The thesis also uncovered some interesting details of the relationships between the hackers and the company, and the balance of power among them.

Sammendrag

I en verden hvor hackere tar datasystemer som gisler i bytte mot løsepenger, og myndighetene truer med bøter hvis personinformasjon kommer på avveie, blir det stadig viktigere å sikre datasystemene sine mot trusler. Flere firmaer har begynt å sette av mer plass i budsjettene deres for tiltak som fremmer informasjonssikkerhet. Likevel, er det ikke alle firmaer som har midlene det kreves til å sikre systemene sine ordentlig, hvor det ofte er de små firmaene som sliter mest. Over 99% av alle firmaer i Norge faller under kategorien små til mellomstor bedrift (SMB), og det vil da være mange firmaer i Norge som vil slite med kjøpe sikkerhetsløsninger etter dagens standard.

Et potensielt billigere alternativ til vanlige sikkerhetsløsninger, som har hat markant fremgang de siste årene, er å sikre bedriftens løsninger gjennom bug bounty programmer (BBP). Dette er en prosess hvor firmaer inviterer hackere til å lete etter sikkerhetssvakheter i systemene sine, hvor penger blir belønnet i bytte mot rapporter om sårbarheter. Denne masteroppgaven sikter på å utforske dette alternativet som en billigere, men fortsatt effektivt fremgangsmåte for norske SMB-er. En fremgangsmåte som vil styrke informasjonssikkerheten til bedriften, samtidig som bedriften kan fortsette å være lønnsom.

Det er gjort lite forskning på om bug bounty programmer er en bærekraftig metode for å bedre sikkerheten, og enda mindre med det norske markedet i tankene. Det ble i løpet av prosjektets gang utført en litteraturgjennomgang av disse forskningsartiklene. Mengden artikler var få, og informasjonen i dem var utdatert, eller tatt i bruk i eksisterende programmer.

Fire delvis strukturerte intervjuer ble holdt med personer fra det norske bug bounty program-miljøet. To norske selskaper som har implementert deres egne bug bounty programmer gav innsikt i hvordan et vellykket program fungerer i Norge. Videre gav two bug bounty-jegere innsyn i forventinger, erfaringer, og inntrykk personene som sitter på den motsatte siden har.

Funnene i oppgaven indikerer at om en SMB har råd til tradisjonell penetrasjonstesting, så burde pengene bli brukt her. Større bedrifter på den andre siden kjøper ofte Bug Bounty som en tjeneste, og reduserer slik kostnadene betraktelig for hver sårbarhet som blir funnet. Oppgaven presenterer likevel et oppsett for et bug bounty program som norske SMBer kan ta i bruk hvis penetrasjonstesting ikke er en mulighet. Oppgaven avdekket også interessante detaljer om forholdet mellom bedrifter og hackerne, samt maktbalansen mellom dem.

Table of Contents

List of Tables

Abbreviations

SMB	=	Small and Midsize Business
BBP	=	Bug Bounty Program
BBaaS	=	Bug Bounty as a Service
OWASP	=	Open Web Application Security Project
PTES	=	Penetration Testing Execution Standard
CVSS	=	Common Vulnerability Scroing System
CIDR	=	Classless Inter-Domain Routing
VRT	=	Vulnerability Rating Taxonomy
IoT	=	Internet of Things
CDN	=	Content Delivery Network
CWE	=	Common Weakness Enumeration
RCE	=	Remote Code Execution
XSS	=	Cross-Site Scripting
CSRF	=	Cross-Site Request Forgery
API	=	Application Programming Interface
SOC	=	Security Operations Center

Chapter 1

Introduction

In recent years, securing systems through Bug Bounty programs has gained remarkable popularity, where Hackerone alone has paid out 42 million dollars since their start, where 18 million of these were paid in 2018 (Hackerone (2019a)). The concept of crowd-sourced security provides a new way of thinking and a potential positive economic aspect. Whereas traditional penetration testing requires a fee to be agreed upon up front, without the assurance of security vulnerabilities being uncovered, bug bounty programs allow the possibility to pay only when security flaws are found, and then repaired by their team of developers. Brugcrowd alone has seen a 40 percent increase in all bug bounty programs launched the last year (Bugcrowd (2018)). A bug bounty program is the act where a company takes precautions and creates contingencies that allow researchers to research vulnerabilities in their systems with the possibility of gaining monetary rewards in exchange for disclosing these vulnerabilities. The concept is fairly new and unproven in the Norwegian market, but may have the potential to increase security at a reduced cost if performed right.

1.1 Target Group

In Norway a small to medium-sized business (SMB) is defined as a company consisting of less than 100 employees, and that more than 99% of all businesses in Norway fall under this category. Close to 47 percent of the people working in the private sector in Norway is employed by an SMB (NHO (2020)). These types of companies are small, and often function on the concept of living hand to mouth, margins are low, processes move quickly, resources are restricted, but the company has to deliver results to survive. Under these circumstances, there is little room for something as costly information security.

There is no doubt that these circumstances are problematic in world where securing your information systems grows more and more essential. Therefore, in this thesis I wanted to look at an alternative way for these businesses to secure their systems, that would still prove effective and cost efficient.

1.2 Methodology

The problem that this thesis has attempted to resolve was to create an optimal bug bounty program for the Norwegian market. In order to address this issue, it was important to take into account several views on the subject. A literature review gave insight into how the most outstanding bug bounty programs are made, as well as input from other research performed on the subject. A semi structured interview was held with a Norwegian company, Visma, which had implemented their own bug bounty program. The company provided experience of how running a bug bounty program is in the Norwegian market. During the thesis an opportunity to interview a second Norwegian company, Finn.no, also appeared, which also had implemented a bug bounty program. The thesis will further introduce two more interviews, with two experienced Norwegian bug bounty hunters, providing a great insight in to the minds of the people that an ideal bug bounty program want to attract. One of them had also been on the security team for a company that hosted their own bug bounty program, and provided a view of how it is to be on both sides of the table. Based on the knowledge gathered through these scientific methods, the means to implement a bug bounty program that may be applied for a company in the Norwegian market.

1.2.1 Literature review

A literature review was done on research papers concerning bug bounty programs. However, very few papers were acute for the thesis. Due to the topic being new, there were not many research papers to base the review on, and the research papers that existed were mostly outdated or proposed something that was already implemented in existing bug bounty programs.

This lead to the idea of reviewing existing bug bounty programs that perform well in the market. The thesis therefore present four different bug bounty programs, as well as two actors that provide bug bounty programs as a service (BBaS).

The goal was then to find common traits in these programs, that can be utilized for the target group which is a Norwegian SMB.

1.2.2 Semi-Structured Interview

The semi-structured interviews was first arranged with Visma, and the two bug bounty hunters. After a period of researching bug bounty programs to come up with questions that could be asked in the interviews, the interviews was held.

The interviews had the open structure of a semi-structured interview, where some questions were prepared on beforehand. However, during the interviews something more similar to a natural dialogue regarding bug bounty programs and their experiences with them occurred.

During the interview with one of the bug bounty hunters, another company that was Norwegian and had a bug bounty program came up. This lead to me contacting them, to hold an additional interview, which had a great result for collecting experience for the bug bounty program this thesis proposes.

An experience with this kind of research method I experienced, was that without an exact date for the interviews being agreed upon, I kept postponing the matter of the inter-

views. Feeling that I was not ready yet, the questions were not good enough, and needed more knowledge on the subject. In retrospection, setting the date early will force the interview, and become a much tidier procedure.

1.3 Limitations

The thesis has some limitations that need to be pointed out. First of all, the existing bug bounty programs that are presented are only four. There are several thousand bug bounty programs worldwide that can be looked upon.

Second, is that only two bug bounty hunters have been interviewed. Considering that hackerone has over 300,000 hackers registered (Hackerone (2019a)), a more quantitative analysis of bug bounty hunters may be attempted in the future Hackerone (2019a).

Third, is that only two Norwegian companies were interviewed, and that neither of them fit the target group which is Norwergian SMBs.

1.4 Assumptions

An assumption that this thesis is taking, is that many of the SMBs in Norway have limited resources to spend on measures to improve information security. The thesis also assumes that when these companies work, shortcuts might be taken to increase profits, such as hiring cheap developers from other countries, or refusing to perform security testing of software and infrastructure.

Chapter 2

Basic Theory and Definitions

In this chapter the information required to understand the remainder of this thesis will be presented.

2.1 Traditional Penetration Testing

Traditional penetration testing is the activity where security specialists are hired by a company to test the security of an information system the business is using. Often the price and scope is determined in advance, and there is normally no guarantee that the security audit will uncover any vulnerabilities.

Penetration Tester

A penetration tester is an individual working with network security and attempts to break into or discover vulnerabilities in different computer systems or software. This is performed by doing tests, where the end product is a report that summarizes the findings.

2.1.1 Penetration Testing Execution Standard

The Penetration Testing Execution Standard (PTES) is a framework with seven sections that cover everything related to a penetration test. It covers the starting phase from the initial communication and reasoning behind a penetration test, onwards through the intelligence gathering and threat modelling, where the goal is to gain a better understanding of the tested organization, and then the framework further covers the vulnerability research, followed by exploitation and post exploitation, and finally how the reporting of vulnerabilities should be handled (Relik (2016)).

2.1.2 Security Vulnerability

A security vulnerability is any flaw in an information system that increases the risk of the system being misused by a malicious user (of Standards and Technology (2020)). This therm covers anything from physical security vulnerabilities, for example a hole in a fence, to flaws in software and hardware, for example the username and password of an administrator stored in the source code of a web page. When dealing with bug bounty program, the vulnerabilities that are found are mainly from the latter type.

2.1.3 OWASP Top Ten Project

The Open Web Application Security Project (OWASP) Top 10 is a document that represents the most broad consensus of the most severe security risks to web applications (Project (2020b)). The main goal being to educate about the consequences of the most common and most important web application vulnerabilities (Project (2017)).

2.1.4 OWASP Risk Rating Methodology

When identifying vulnerabilities it is important to assess how high risk the vulnerability exposes the rest of the system or company for. The OWASP Risk Rating Methodology is an approach for estimating the severity of these risks. By utilizing a system for rating risks, the business can save time and eliminate the need for arguing about priorities. OWASP explains this in a simple manner of risk being equal to the likelihood times the impact of the flaw being exploited, and propose a six step process for reaching the final number (Project (2020a)).

2.1.5 Hacker

An individual skilled at computers, who is able to gain access to computer systems without permission.

2.2 Bug Bounty Programs

The dictionary describes a bounty as a ”money paid as a reward” (Unviversity (2020)). A bug bounty program is a framework that facilitates the procedure where a business can crowdsource their security procedures where the company pays for valid vulnerabilities that are reported to the company through the right channels.

2.2.1 Public Bug Bounty Program

A bug bounty program that is publicly available to security researchers around the world, that helps scale testing, and gains access to an extensive, diverse set of skills. A program suited for systems that are publicly faced (Bugcrowd (2020a)).

2.2.2 Private Bug Bounty Program

A private bug bounty program provides a controlled testing environment with a smaller set of highly experienced and proven security researches. Suited for systems that are not publicly available, such as staged environments, applications requiring authorized access, or devices (Bugcrowd (2020a)).

2.2.3 Managed Bug Bounty Program

A program that is managed by another company that is a offer bug bounty programs as a service (BBaaS). The company will design, manage, and support the program for the customer from beginning to end (Hackerone (2020b)).

2.2.4 Bug Bounty Hunter

A security researcher targeting systems that are part of a bug bounty program. The researcher is a hacker that is following the program's rules of engagement, and does not need to have this as a full time job, and can combine it with a regular job.

2.2.5 Rules Of Engagement

There exists a risk by opening the business' systems to a world of white-hat hackers. The rules of engagement are a set of restrictions the company can publish to stay in control of the situation. The rules contain everything from the attack surface and what kind of vulnerabilities that are accepted, to the process of reporting the vulnerabilities and how the payment process will be (Mingyi Zhao and Grossklags (2017)).

2.3 Additional Definitions

This thesis uses several technical terms that are widely used withing information security. This section contains a list of definitions for the terms that are used.

2.3.1 Security.txt

In the information security world there exists a proposed standard called "security.txt", which is a file companies can host on their web pages with instructions to how vulnerability researchers can inform the organization about vulnerabilities in their systems (EdOverflow and Shafranovich).

2.3.2 Crowdsourcing

Crowdsourcing is the act of tasking a large group of people with something that, when solved, will have a positive impact for the party handing out the task. An example being, asking the Internet for help, instead of letting an employee handle the task.

2.3.3 Sandbox

A sandbox is an isolated computer environment where untested code can be run. In information security terms, the sandbox environment is thus an environment where vulnerabilities have less impact (Rouse (2018)).

2.3.4 Common Vulnerability Scoring System Version 3.0

The Common Vulnerability Scroing System (CVSS) is a framework used to communicate the characteristics and severity of software vulnerabilities. The CVSS measures vulnerabilities in three groups :

- Base, which represents the essential qualities of the vulnerability that are constant and across computer environments.
- Temporal, which represents the characteristics of the vulnerability that change over time.
- Environmental, which represents the characteristics of the vulnerability that are unique to the user's environment.

Based on the three categories, the vulnerability will end up with a score from 0-10 where 10 is a vulnerability which is easy, and likely to be exploited, and has a devastating impact, and 0 is hard or unlikely to be exploited, with a low impact(Inc (2020)).

2.3.5 Payload

A payload is the component of an attack that causes harm to a computer system (Cloudflare (2020)). For vulnerabilities, it is the code, or value of an input parameter that makes an attacker able to compromise the confidentiality, integrity or availability of the system.

2.3.6 Zero-day Vulnerability

A zero-day vulnerability is a software security flaw in a computer system, software, or hardware that is known to the vendor but has not yet been fixed with a patch, and is open to being exploited by a hacker (Norton (2020)).

2.3.7 Classless Inter-Domain Routing (CIDR) Notation

The Classless Inter-Domain Routing (CIDR) notation is a method of defining blocks of IP-addresses into a routed network, where you also define the size of the block by the number of bits that are pre-determined. An IPv4-address consits of 32 bits, and with a CIDR-notation of 172.28.1.0/24 then has 24 determined bits, leaving 8 bits describe the remaining IP-addresses in that block. The mentioned block will then cover all the IP-addresses from 172.28.1.0 to 172.28.1.255 (Address (2020)).

Chapter 3

Existing Bug Bounty Programs

In this section I will cover the literature of some of the state of the art bug bounty programs that exists in the world today. In all the cases I will try to present the most significant parts that have made them become the programs of the stature that they have. These points will later be analysed, used as the foundation of a bug bounty program fit for SMBs in the Norwegian market.

3.1 Bugcrowd

Bugcrowd is the number one crowdsourced security platform in the world, and has experienced large growth in the recent years, where they facilitate bug bounty programs for organizations. Every year they release an article about the state of their bug bounty program, releasing statistics of how the market has moved. By understanding what Bugcrowd and other similar platforms provide, a basis for what could probably be the easiest solution for companies wanting to attempt bug bounty programs to tighten security will be achieved.

3.1.1 Programs

As of 26.11.2019, there are 110 available programs presented on Bugcrowd's website, where among the most well known, are Mastercard, Trip Advisor, and Tesla. Among those 110 programs, 80 programs provide monetary bounties in exchange for vulnerabilities (Bugcrowd (2020c)).

3.1.2 Vulnerability Rating Taxonomy (VRT)

At the start of 2016 Bugcrowd released a framework that provides a baseline vulnerability priority scale that can be utilizied by hackers and companies, and has since been evolved, and became open source in 2017.

The VRT is used as a tool to help decide the technical severity rating a vulnerability carries, also taking into account the potential differences among cases that can vary in severity. The framework divides vulnerabilities into top level categories, that further gain sub-categories, which then can be nested with a third level called variants (Bugcrowd (2020d)).

When the vulnerability has been mapped to its catergory, sub-categories, and variants, a priority level of 1-5 can then be set. This priority level ultimately decides the severity of the vulnerability, and thus which reward range a vulnerability should be placed in. A complete list of the vulnerabilities that have been categorized by Bugcrowd so far can be found at their homepage (Bugcrowd (2020e)).

3.2 Hackerone

Hackerone is alongside Bugcrowd considered to be one of the market leaders for Bug Bounty Program as a Services (BBaaS) The clients are varied, and range from large national institutions such as the U.S. department of Defense, and U.S. Air Force, to the well known websites such as Github and Airbnb (Hackerone (2019b)). The company consists of a community of over 300,000 individuals they call their hacker community, where the community grows by 600 hackers every day.

3.2.1 Programs

As of 30.11.2019, Hackerone presents 349 different bug bounty programs. However, several of them are seemingly outdated, or completed, and the real number of active programs are 269. Among these, 110 are managed bug bounty programs. From the total number of programs, 172 of the programs provide bounties for submitting bugs (Hackerone (2020c)).

Private Programs

An option for companies that want a subtle bug bounty program is Hackerone's private program, which are not published to the public. Hackers will then not be able to see the program until they receive an invitation, and vulnerability reports to the program will remain confidential unless the company chooses to disclose them. Hackerone also states that all programs are private as default, and may remain that way for as long as they want.

Hackerone further recommend that the companies starting bug bounty programs start out private to prevent them from being bombarded with report submissions, and that private bug bounty programs have the option to limit the number of hackers that are invited to the program. A public program can later be chosen when the company are comfortable with running the program.

Public Programs

Public bug bounty programs are open to the public, and thus open to report submissions from all the hackers registered on Hackerone. They state that doing this prematurely can

be overwhelming, due to a large influx of new reports being submitted, where submissions can increase by up to five to ten times the number of submissions in private programs.

Hackerone also states a list of criteria a company should fulfill before taking the step to a public program:

- The company has invited more than 100 hackers to the program.
- The company has received 10 vulnerability reports.
- The program meets HackerOne's response standards.

The response criteria Hackerone requires the company to meet to go public are that the company's first response to a hacker takes no more than one day, and that the time to triage takes no more than two days.

Hackerone also performs a controlled launch when going public. A steady flow of attackers will be invited on a continuous basis, where the invitations will pause after the company receives five valid reports within 30 days.

Categories

Hackerone divides their programs into several categories:

- CIDR
- Domain
- iOS: apps from App Store
- iOS: apps on Testflight
- iOS: .ipa
- Android: apps from Playstore
- Android: .apk
- Windows: apps from Microsoft Store
- Source Code
- Executable
- Hardware/Internet of Things (IoT)

3.2.2 90 Day Leaderboard

Hackerone presents a leaderboard for their 100 top hackers for the past 90 days. The hackers are rated on three categories:

- Positive reputation - When hackers submit valid reports, reputation is gained.

- Non-negative signal - The hacker does not submit invalid reports. An invalid report may result in negative signal.
- Zero code of conduct violations - If program policies are broken, code of conduct violations may be the result.

Furthermore, to move up in ranking the hackers either has to increase their total Hackerone score that is calculated by Reputation x Signal Percentile x Impact Percentile where:

- Reputation is based on valid reports that are submitted
- Signal and Impact Percentile is calculated against all the program's hackers, where impact is the impact the submitted reports carry.

The reason why a hacker should increase their ranking on Hackerone is specified to be eligible for invitations more private bug bounty programs (Hackerone (2020a)).

3.3 Google Vulnerability Reward Program

In November 2010, Google launched their own bug bounty program called the Google Vulnerability Reward Program, and has since become one of the largest public bug bounty programs on the market.

3.3.1 Scope

In principal, any Google-owned web service that handles sensitive userdata is included in the scope of the Google Vulnerability Reward Program. Which ultimately makes the scope to be:

- *.google.com
- *.youtube.com
- *.blogger.com

Where "*" is a wildcard, implying the domain in question and all attached subdomains.

The program also covers bugs found in the Google Cloud Platform, Google developed apps and extensions, as well as some of their hardware devices. However, these are covered by two other Bug Bounty Programs; Android Security Rewards Program, and Chrome Vulnerability Rewards Program.

Google also lists two important exclusions from the program:

- Third-party websites, which are services carrying the Google brand, but are operated vendors or partners. The exclusion is mainly based on legal reasons, where Google are not authorized to allow vulnerability research on services they do not own.
- Recent acquisitions. To provide time for an internal review of services recently acquired, Google sets a 6 month grace period on the services. Where bugs reported before this grace period ending, will not likely provide a reward.

3.3.2 Qualifying Vulnerabilities

To avoid confusion and unnecessary vulnerability reports, Google provides a definition on what they consider a valid. A qualifying vulnerability is ”Any design or implementation issue that substantially affects the confidentiality or integrity of user data”. Some examples of this are the traditional vulnerabilities:

- Cross-site scripting
- Cross-site request forgery
- Mixed-content scripts
- Authentication or authorization flaws
- Server-side code execution bugs

However, Google also points out that other methods for abusing their systems are accepted. These methods are weaknesses that are not covered by the traditional categories of vulnerabilities, but can hinder the intended functionality of the system. An example of this is if an individual would be able to submit a vast amount of reviews of a business on google maps. This would lead to the rating of the business to become inaccurate, and will ultimately end up to damage Google’s reputation.

3.3.3 Non-qualifying Vulnerabilities

In addition to a list of valid Vulnerabilities, Google has also published an overview of low risk weaknesses that do not provide rewards:

- Vulnerabilities in the domains *.bc.googleusercontent.com and *.appspot.com. These are domains where Google’s cloud customers host applications. Applications that the Vulnerability Rewards program does not authorize testing for.
- Cross-site scripting vulnerabilities in ”sandbox” domains. Google has several domains utilized to safely isolate untrusted content. Unless the vulnerability can provide a proof of concept where sensitive user data is compromised, JavaScript execution will not provide a reward.
- Due to it being regarded as a feature and not a security bug, executing JavaScript in a user’s own personal blog on *.blogspot.com doe not provide rewards.
- Google require URL redirection to make their systems cooperate, and thus accepts the risk connected to redirects.
- Vulnerabilities related to Google-services labeling third party content.
- Bugs requiring an unlikely amount of user interaction to perform the attack.An example can be a reflected Cross-Site Scritping attack, where the user needs to type in the JavaScript that will be executed in the victim’s browser.
- Cross Site Request Forgery vulnerabilities that logs a user out.

- Security bugs that exploits vulnerabilities in a user's outdated network browser and plugin.
- Exposed version information does not directly make the system vulnerable to attacks.
- Spoofing on Gmail and Google Groups.
- User enumeration. Google does not accept accept vulnerabilities regarding user enumeration, unless the vulnerability also bypasses Googe's rate limiters.
- Bypassing the limit of the number of accounts that can be verified by a single SMS number.

3.3.4 Rewards

A bug bounty program, is not a real bug bounty program without its bounties. Google presents the rewards in a structured matter. The rewards are divided into two main categories; rewards for security vulnerabilities, and rewards for abuse-related methodologies. To help determine the amount of the reward, a table ranging the bugs in a table accounting for the probability of the attack being successful combined with the impact of the attack is used.

Rewards for Security Vulnerabilities.

In addition to providing rewards of up to 100 000 dollars for vulnerabilities in the Google Cloud Platform (Google (2019)) the rewards from the Google Vulnerability Rewards Program range from 100 dollars to 31,337 dollars, depending on a set of different impact targets:

- Target group 1 - Applications permitting the compromising of a Google account
- Target group 2 - Other highly sensitive applications
- Target group 3 - Normal Google applications
- Target group 4 - Applications with lower privileges, such as acquisitions that have not been integrated and sandbox domains

Category	**Group 1**	**Group 2**	**Group 3**	**Group 4**
Remote code execution	\$31,337	\$31,337	\$31,337	$1,337-5,000$
Access to file systems or databases	\$13,337	\$13,337	\$13,337	$1,337-5,000$
Logic security bugs	\$13,337	\$7,500	\$5,000	\$500
Execute code on user's machine	\$7,500	\$5,000	\$3,133.7	\$100
Other valid security vulnerabilities	\$500-\$7,500	\$500-\$5,000	\$500-\$3,133.7	\$100

Table 3.1: The table shows what secondary vulnerabilities are regularly rewarded with according to the target group.

Rewards for abuse-related methodologies

Google also presents a matrix for deciding the amount of rewards for bugs that are abuse-related, where the amount is decided on a combination of potential probability of the flaw being exploited and the impact of the submitted technique.

		Impact		
		High	**Medium**	**Low**
Probability	**High**	Up to $5,000	$1,337-$3,133.7	$500
	Medium	$1,337-$3,133.7	$500	$100
	Low	$500	$100	$Hall of Fame Credit

Table 3.2: The table shows what severe vulnerabilities are regularly rewarded with according to the target group.

The impact is assessed based on the attack's potential for causing privacy violations, financial loss, and harm to users. Whereas the probability is based on the technical skill set needed to perform the attack, the potential threat actors, and the likelihood of an attacker discovering the vulnerability.

Disclaimer

Following the reward tables is also a disclaimer stating that the amount is always decided by the reward panel. The panel is authorized and may decide to pay higher rewards for unusually clever or severe vulnerabilities, or lower rewards for vulnerabilities that require unlikely or unusual user interaction. A report can also be marked as containing multiple security bugs, or that multiple reports are so closely related to another that they only warrant a single reward.

Google also point out that they are aware some hackers are not interested in monetary rewards. For these individuals they propose to donate the reward to charity, where Google then also will double the reward.

3.3.5 Investigating and reporting bugs

Google asks that when researchers investigate vulnerabilities, that they only target their own accounts, and to never attempt to access anyone else's data. They also ask the hackers to not perform any activity that may be disruptive or damaging to other users or Google.

For participants that have uncovered vulnerabilities that are covered by the vulnerability reward program, they forward the user to a website that contains a form specially designed for submitting vulnerabilities. Further underlining that this form is for technical vulnerabilities only, and that enquiries concerning problems with Google services are to be submitted at the Google Help Center.

3.3.6 Legal points

The company finish the program site with some legal statements. Google explains that they are not able to reward individuals that are on sanctions lists, or are in a country that

is on a sanctions, such as Cuba, Iran, North Korea, Sudan and Syria. They also state that there might be further restrictions that the researcher's country have, that Google have to comply with. The hackers are also forbidden to violate any law that they are under the jurisdiction of.

Further, Google states that the program is not a competition, but rather an experimental and discretionary rewards program. They have the right to cancel the program at any time, and also reserve the right to decide whether or not to pay a reward.

3.3.7 Hall of Fame

In addition to monetary rewards, Google also have a hall of fame list of the top contributors to the vulnerability reward program, and presents the hackers that have qualified for rewards since 2015. The list ranks the hackers in the following categories:

- Volume - How many valid bug reports the hacker have sent in.
- Severity - How severe are the bugs that have been submitted.
- Recency - The most recently submitted reports earn full points total, and the report's points count continuously start to diminish after six months, for up to 50% point loss.
- Charity - If the hacker donate their reward, they will earn extra points.

Prior to 2015 the Hall of Fame is comprised of static lists of unranked contributors (Google (2020a,b)).

3.4 Facebook

Facebook launched a bug bounty program in 2011, and is on of the oldest programs on the market. Since the beginning, it has paid out over 7.5 million dollars in bounties, where the biggest single paid out bounty reached 50 000 dollars, which is a considerable amount considering their average prize lies on around 1500 dollars (Newman (2018)). The bug bounty program offers guidelines for independent parties that have found a security, or are looking for, vulnerabilities on either Facebook or one of their other organizations. The company has set up guidelines for how to report bugs, and divides it into a responsible disclosure policy and bug bounty program terms.

3.4.1 Responsible Disclosure

The responsible disclosure policy describes the terms for a hacker that wants to report a bug to Facebook, where a hacker needs to comply to not risk legal actions being taken against him in form of a lawsuit or law enforcement investigation.

- The hacker needs to give Facebook reasonable time to investigate and mitigate an issue that he reports before releasing any information about the vulnerability to the public.

- The hacker can not interact with an account (including modifying or accessing data from the account) without the owner's consent.
- The hacker needs, by best effort, to avoid privacy violations and cause disruption to others. This includes unauthorised access to or destruction of data, and disruption or degradation of Facebook's services.
- The hacker does not attempt to exploit the vulnerability he discovers, including demonstrating additional risk.
- The hacker should not intentionally break any other applicable law or regulation.
- The hacker is not authorised to access user or company data, including information and data related to identifying an individual.

3.4.2 Bug Bounty Program Terms

Facebook acknowledge and reward security researches that want to aid them in keeping the security up to standards by reporting vulnerabilities in their services. They have every right to award the vulnerability reports they receive based on risk, impact, and some other factors that they do not disclose.

- The researcher needs to follow Facebook's responsible disclosure guidelines.
- Reporting a security bug at Facebook means identifying a vulnerability in their services or infrastructure which generates a security or privacy risk. They ask the hacker to note that Facebook are the ones who ultimately determines the risk, and that software bugs are not always issues regarding security.
- The report needs to describe a problem with one of the programs or services listened in Facebook's bug bounty program's scope.
- Facebook has a list of potential security issues that they want to exclude from the program, these are listed in a section for out of scope items.
- When submitting a report, the hacker is asked to use a specially crafted form especially made for submitting security vulnerabilities.
- If the hacker violates or disrupts the privacy of the solution when investigating a bug, Facebook asks them to disclose this in the report.
- When the hacker researches security bugs, he is asked to utilize designated test accounts. However, if the issue cannot be reproduced with the test accounts, it is allowed to use real accounts, as long as they have consent from the account owner.

Furthermore they state that if the researchers are able to follow the company's guidelines, they promise to honor the following terms:

- They will investigate and respond to all incoming reports. However, they precociously inform that the due to the volume of reports they receive, a response might take some time. The reports are prioritized after what risk they pose.

- The bounties are determined by (but not limited to) impact, ease of exploitation, and the quality of the report. The minimum amount for a bounty is 500 dollars, meaning that extremely low risk issues might not be qualified for a payout.
- Facebook aims to pay similar bounties for similar cases. However, the amounts for issues may change over time, hence rewards in the past do not necessarily guarantee similar bounties in the future.
- If duplicates occur, Facebook will only pay bounties to the first person to submit an issue, and a given bounty is only paid to one individual.
- The hacker has the option to donate the bounty to a chosen charity, if this is the case, Facebook will double the amount of the bounty.
- Facebook reserves the right to publish the reports, and accompanying updates for the vulnerabilities.
- Facebook has a published list of researchers who have submitted valid security reports to their program. To be included in this list the hacker must have received a bounty from Facebook, but the hacker also has the right to not participate. The organization does retain the right to modify the information that comes with the names on the list.
- Facebook verifies that the rewards are allowed by applicable laws. This includes US trade sanctions and economic restrictions.

Including the terms listed above, Facebook may also offer promotions in regards to their Bug Bounty program. These promotions might have additional rules, that are described on a separate website reserved for promotional bug bounty programs.

Facebook also enlightens the hackers that when using Facebook services, or services of connected organizations, the hacker are subject to their Terms and Policies, and also reserve their right to cancel or modify the program at any time they deem fit.

3.4.3 The Scope

The Facebook bug bounty program is restricted to a scope of services. Where a bounty is rewarded for security bugs on Facebook or one of the following products also managed by Facebook:

- Instagram
- Internet.org/Free Basics
- Oculus
- Onavo
- Open source projects by Facebook such as for example osquery
- WhatsApp

They also specify that services that they might use, but are not owned by Facebook are not included in the program. Although vulnerabilities in these services are of interest, Facebook does not guarantee that their policies apply to services from other companies.

Furthermore, they state that vulnerabilities in third-party apps or websites that utilize features by Facebook are generally also not in scope of their bug bounty program. However, one exception exists, which is security bugs leading to Facebook user's session tokens being exposed to unauthorized parties. This exception is restricted to passively observing data being transmitted from the hackers devices, and the researcher is not allowed to manipulate any requests that are being sent from these apps that use Facebook features. The company also prohibit the researchers from accessing data by using any session tokens than their own. Only third-party apps that have at least 50,000 active users are covered by the scope.

The scope is then further specified in a table for what domains, subdomains, and apps that are eligible and ineligible for bounty hunting:

Target	Eligible	Ineligeble
Facebook	Websites: facebook.com, fb.com, fb.me, messenger.com, thefacebook.com, accountkit.com	Websites: events.fb.com, fbsbx.com, investor.fb.com, media.fb.com, newsroom.fb.com, research.fb.com, search.fb.com, work.fb.com, research.fb.com, madebykorea.fb.com
	Apps: Ads Manager, Facebook, Facebook Lite, Workplace by Facebook, Groups, Hello, Mentions, Messenger, Moments, Pages Manager, Paper (by Facebook), Work Chat	Apps: Facebook for Blackberry, Facebook for Windows
Instagram	Websites: instagram.com	Websites: blog.instagram.com
	Apps: Boomerang, Hyperlapse, Instagram, Layout	
Internet.org	Websites: freebasics.com, internet.org	
	Apps: Free Basics	
Oculus	Websites: oculus.com	Websites: answers.oculus.com, forums.oculus.com, support.oculus.com
	Hardware: All first party hardware	
	Software: First party PC and mobile apps	
Onavo	Websites: onavo.com	Websites: Websites: blog.onavo.com
	Apps: Onavo Count, Onavo Extend, Onavo Protect	
Open Source	Code repos: https://github.com/facebook/	Code repos: https://github.com/facebookarchive/
WhatsApp	Websites: blog.whatsapp.com, translate.whatsapp.com, web.whatsapp.com, whatsapp.net, www.whatsapp.com	Websites: alpha.whatsapp.com, media.whatsapp.com
	Apps: WhatsApp	
Other partnerships /acquisitions		Websites: daytum.com, drop.io, face.com, friendfeed.com, monoidics.com, opencompute.org and spaceport.io

Table 3.3: The table shows the eligible and ineligeble items of Facebook's scope.

3.4.4 Out of scope

Facebook further lists up a few items that are outside of scope, and will not grant a reward:

- Attacks that target people and not computer systems, such as social engineering and spam campaigns.

- Services that target the availability of the service such as denial of service attacks.
- Content injection, which is the act of publishing malicious content on Facebook's services. Publishing content is a core feature, and a significant risk needs to be demonstrated for it to grant a bounty.
- Vulnerabilities in third-party applications or websites that integrate with Facebook, which includes most pages on ""apps.facebook.com". The special circumstances described in the program policies are still valid and do grant a reward.
- Executing scripts on sandboxed domains such as fbrell.com or fbsbx.com.

3.4.5 False positives

The organization also lists a set with false positives that will not be rewarded with a bounty:

- Open redirects. Facebook has a speicified function for open redirects, called "linkshim" and vulnerabilities attached to the function is a false positive.
- Reports pointing out that profile pictures are publicly available, and reports covering other information Facebook regards ass public such as; username, ID, name, current cover photo, gender, and information people has made public.
- Sending messages to anyone on Facebook. This is regarded as a feature.
- Being able to access photos via raw image URLs from Facebook's CDN (Content Delivery Network).
- Non-case-sensitive passwords. Facebook regards it as a feature that they accept the first letter of a password capitalised, or the full password in capital letters.

`https://www.facebook.com/whitehat`

3.4.6 List of researchers

The bug bounty policies mentions a list of vulnerability researchers, where everyone that has received bugs has the possibility of being a part of. The list is a simple list of names (ranked in 2018-2019) that go back to 2011. The list presents all researchers that have contributed to the program of the corresponding year. Where some of the names also link to the blog or linkedin page of the attacker. `https://www.facebook.com/whitehat/thanks/`

3.4.7 Report Vulnerability Form

Furthermore, the policies mention a place to submit vulnerabilities. The researcher first has to check a box on the website for "I found a technical security bug in a Facebook product." and a form pops up asking for more information. The information Facebook is requesting is:

- Which product the vulnerability is connected to.
- Which type of vulnerability is it.
- The title for the report.
- A description of the vulnerability, where they state that this section should be the longest, and request that the hacker be as thorough and descriptive as possible. This section should also contain the impact that the vulnerability has for the service.

The form also requires a detailed instruction of how to reproduce the vulnerability, and asks the researcher to fill in the following information:

- The users that are needed to reproduce the vulnerability.
- Which environment on the service, for example a group on Facebook.
- Which web browser is used.
- The operating systems where the vulnerability exist.
- Any other details or descriptions that are too complex to ask for in advance.
- A numbered list of steps that are required to perform the attack. Each number only having one or two sentences for each step.

Finally it is possible to upload file attachments to further demonstrate the process and impact of the vulnerability. The submission process also underlines the fact that by reporting a vulnerability, the researcher is agreeing to the terms of the bug bounty program stated above (Facebook (2020a,b)).

3.5 Verizon Media

Verizon media started their bug bounty program in February 2014, and are running a managed program on Hackerone. As of February 2020 5763 vulnerability reports spread over 44 assets has been resolved on their program with rewards averaging between 394-500 dollars, and is thus one of the largest bug bounty programs on Hackerone (Hackerone (2020d).

Verizon Media handles brand names such as Yahoo, HuffPost and TechCrunch, and invite security researchers to help protect all their resources and users with a bug bounty program, and encourage the hackers to review their policies to be compliant with their rules for hunting bugs.

The program presents a table of content regarding the terms and regulations of the program in three parts:

- Rules of Engagement, consisting of three more parts:
    1. Program Rules
    2. Legal Terms

3. Safe Harbor

- Responsible Disclosure of Vulnerabilities, also divided:
  1. Testing
  2. Crafting a Report
  3. Scope
- Rewards:
  1. Reward table
  2. Important Vulnerabilities
  3. Borderline Out-of-Scope items, that reward no bounty
  4. Items not to report

3.5.1 Rules of Engagement

The rules of engagement that Verizon Media lists are rules they expect their participants follow if they were to submit a vulnerability report, and warn that violation of the rules can result in ineligibility for a bounty and/or removal from the program. After three violations the hacker earns a temporary ban from the program, and four strikes will make the removal permanent.

Program Rules

The rules are as follows:

1. Vulnerabilities are only to be tested against accounts that the hacker owns or accounts that he/she has permission from the account owner to test against.
2. It is prohibited to use a finding to compromise or extract data, or to use the vulnerability to gain access to other data systems. A proof of concept should only be used to demonstrate the issue.
3. If sensitive information is obtained as part of a vulnerability, the hacker must avoid saving, or distribute it after discovery. If copies of information occur, it is to be returned to Verizon Media.
4. Participants are not authorized to perform actions that may be disruptive, damaging, or harmful to Verizon Media, their brands, or users, which includes social engineering, phishing, breaches of physical security and denial of service.
5. The hackers are to follow the program scope. Only reports submitted to the program and contain vulnerabilities concerning the scope qualifies for monetary rewards.
6. Researchers are not allowed to publicly disclose vulnerabilities they uncover without Verizon Media's written permission to do so.

Legal Terms

The legal first state that the researcher needs to comply with Verizon Media's Terms of Service, Privacy Policy, and all applicable laws and regulations governing privacy and processing of data.

Also in the terms, the company reserve the right to change and modify the terms of the bug bounty program at any time, and note point out that any researcher may not participate if he or she is a resident or located in a country appearing on any U.S sanctions lists.

Furthermore, the legal terms state clearly that the company does not grant authorization to individuals to extract personal information or content of Verizon Media users. The researcher is also not allowed to publicize the information without consent, or modify or corrupt programs or information that belongs to Verizon Media.

Verizon Media also restricts the participation to the program from individuals that are connected to the organizations in some matter, meaning that if a person matches one of the following criteria, is ineligible to receive bounties from the program:

- Employees (including a 12 month lockout after ended employment)
- Contingent workers
- Contractors and their personnel
- Consultants
- Immediate family members and individuals living in the same household of anyone mentioned above.

Safe Harbor

The safe harbor is a section where Verizon Media states that they will not initiate a law suit or law enforcement investigations against researchers that report vulnerabilities to the program. This is contingent to the attacker fully complying with the program's policies.

Verizon Media also states that if the hacker's research involves the networks, systems, information, applications, products, or services of any third party that is not them, they have no control over the third party pursuing legal action. This is due to them not being able to authorize security research in the name of other entities. However, if the researcher has followed the program's policies, and is the target of legal actions, Verizon Media will make a statement that the researchers actions were conducted in compliance with the policies.

3.5.2 Responsible Disclosure of Vulnerabilities

This section describes how the researcher is going to act when submitting a security vulnerability, and explains what he can expect. The researcher is instructed in how to perform testing, and how to construct the report for a potential vulnerability. Verizon also explains that they will be performing after best effort, and that they will respond to submissions as soon as possible, and make every effort to fix the bugs they receive within 90 days of the bug being triaged.

Testing

Due to the massive amount of data traffic that is transferred to and from Verizon Media resources every day, the organization proposes a method for them to easier identify and separate the researcher's testing traffic, from normal data and malicious actors out in the world. Horizon encourages the researchers to:

- Where possible, register accounts on the researcher's Hackerone email address.
- Include his IP address in the bug report, it will be kept private.
- Include custom headers in HTTP requests on all outbound requests. For example:
  - A header with the researcher's username.
  - A header that includes a unique flag.

Some other instructions the hackers are encouraged to follow are:

- To only use authorized accounts to not compromise the privacy of other users.
- To utilise the following commands to demonstrate privileged user access on their servers:
  - Read permissions: `c`at /proc/1/maps
  - Write permissions: `t`ouch /root/¡researcher's hackerone username¿
  - Permission to execute commands: `i`d, hostname, pwd
- To minimize mayhem by following the program rules at all times. Automated scanners/tools may trigger state changes or damage proudction systems, and are not allowed.
- To stop testing, and report the finding before risking to cause any damage. Permission to test further can then be requested.

Crafting a Report

Verizon Media point out the importance of being able to reproduce and verify an issue, else a bounty cannot be awarded. To help streamline their process, they present a list of what submitted report should contain, and warn the user that a reduced bounty may be the result if the user fails to produce:

- A description of the vulnerability.
- The steps required to reproduce the vulnerability.
- Proof of the vulnerability being exploited, normally in the form of screenshots or video.
- A description of the impact the vulnerability carries for other users, or the organization.

- A Proposed CVSSv3 Vector and Score
- A list of the involved URLs and the affected parameters.
- Additional URLs that are vulnerable, other payloads, and code for the Proof-of-Concept code
- System information, such as browsers used, OS, and/or application versions used for testing.

The researcher is also told to not store the evidence on any external services. It should only be stored with the report when submitted.

Furthermore the program states some other guidelines for reports that cover special scenarios that might occur. In the event that a research uncovers a vulnerability that is applicable for multiple unique hosts, but still the same bug, Verizon Media offers a 10 percent bonus bounty for each vulnerable host. If the researcher discovers additional hosts while the vulnerability is being triaged, the hacker is encouraged to add this host to existing report. However, if a separate report is filed for the other host while the company is resolving the first issue, the report will be treated as a duplicate.

Another scenario that the organization describes is in the event that a payload used to demonstrate a vulnerability is applicable for several input parameters. If this happens, the researcher is encouraged to consolidate reports, to receive one total bounty that is the sum of the additional parameters.

Program Scope

Verizon Media has several programs for specific brands and web properties, and encourage the researchers to report vulnerabilities to the program that the bug belongs to. They have a detailed scope list, which also explains the out of scope assets.

The list of items in scope is a vast list of services that the company owns, and can be divided into a variety of challenges:

- Different domains that are in scope
- Links to repositories containing source code, items and domains in the source code that are out of scope needs to be listed.
- A variety of Yahoo applications and other apps for for android, iOS, tvOS, FireOS and web.

Listed next to each service in scope, is also an indicator whether the service is eligible for bounty or not, and if the service is important for the company.

The policies also mention several items that are not covered by the scope. This either due to the service being sold, is outdated and has been phased out, or strictly that even though the domain name indicates that it is owned by Verizon Media, it is not.

3.5.3 Rewards

The rewards is a sum of money paid to the first person to disclose an unknown security issue to Verizon Media. Vulnerabilities that qualify for a reward will be judged on severity, which is solely determined by Verizon Media, and have the amount decided thereafter. In the program, providing more complete research, proof-of-concept code and detailed write-ups may increase the bounty. Lack of providing substantial information can also influence in a negative manner, where policy violations can even lead to rewards being denied. Reports that require specific browser configurations may also cause a reduced reward.

Payout Table

Verizon Media divides the vulnerabilities into categories based on their severity, which is done after identifying the final impact the vulnerability carries for the company. Each category has its decided payout range. A vulnerability in a category will be rewarded with an amount withing that range:

Severity	Payout Range
Critical	$10,000-15,000$
High	$3,000-10,000$
Medium	$500-3,000$
Low	$0-500$
None	$0

Table 3.4: The table shows how the vulnerability's impact corresponds with it's rewarding range.

Valued Vulnerabilities

Along with a list over the amounts each vulnerability can receive, a table with example vulnerabilities that are considered as valuable are also presented. The vulnerabilities are classified by a scoring framework called the Common Weakness Enumeration (CWE) classification. The table lists some common attack names that they classify within each CWE, and is only meant as a guide. Verizon Media has the final saying in a vulnerability's severity. A part of the table is shown below:

Low severity	High Severity	CWE-ID	CWE	Bug Examples
Critical	Critical	CWE-78	OS Command Injection	RCE, Code Injection, LDAP Injection
Critical	Critical	CEW-120	Classic Buffer Overflow	Buffer Overflow
High	Critical	CEW-89	SQL-Injection	SQL Injection

Table 3.5: The table shows examples of vulnerabilities and their corresponding CWE.

The table listed with the program is not final, and non-listed vulnerabilities may also be eligible. Some vulnerability types may also fall under several CWE's at once. The

researchers are also told that 0-day vulnerabilities may be reported 30 days after the vulnerability has been made publicly available. Verizon Media has a team tracking such issues, and if a service has already been identified by internal resources, a reward will not be granted.

Borderline Out-of-Scope

A list of issues that are eligible for submission, but not eligible for bounty or any reward is also presented. The reports will be triaged, and either be deemed informative if valid, or spam if not valid. The researchers are asked to keep two things in mind when submitting a bug, the first being attack scenario or exploitability, and the second being the security impact of the bug.

- Any non-Verizon Media Applications - ”Self” XSS
- Missing Security Best Practices - HTTP Host Header XSS
- Confidential Information Leakage - Clickjacking/UI Redressing
- Use of known-vulnerable library (without proof of exploitability) - Intentional Open Redirects
- Missing cookie flags - Reflected file download
- SSL/TLS Best Practices - Incomplete/Missing SPF/DKIM
- Physical attacks - Social Engineering attacks
- Results of automated scanners - Login/Logout/Unauthenticated CSRF
- Autocomplete attribute on web forms - Using unreported vulnerabilities
- ”Self” exploitation - Issues related to networking protocols
- XSS in flash files not developed by Verizon Media (e.g. Camtasia, JW Player, Flowplayer swf files) - Software Version Disclosure
- Verbose error pages (without proof of exploitability) - Denial of Service attacks
- Verizon Media software that is End of Life or no longer supported - Account/email Enumeration
- Missing Security HTTP Headers (without proof of exploitability) - Internal pivoting, scanning, exploiting, or exfiltrating data

Do Not Report

At the very end, Verizon Media also provides a list with issues that should not even be reported. Issues that will not be considered are:

- Issues that resolve to third party services.
- Issues that does not affect the newest versions of web browsers.
- Issues that they are already aware of, or have been reported previously.
- Issues that require unlikely interaction from the user.
- Issues regarding disclosure of information that does not present significant risk.
- Issues involving Cross-site Request Forgery with minimal security impact.
- Issues regarding CSV Injection.
- Issues involving incomplete or lack of the email configurations SPF/DKIM.
- Issues involving security best practice concerns.

Statistics

The Verizon Media also presents some of the program's statistics on the program's homepage. Regarding Response efficiency, 98% of the reports they receive meed Hackerone's response standards, where the average time to first response is 7 hours, and the average time to triage is two days. The time it takes for bounties to be paid out is 18 days.

Whereas for the rewards, the minimum bounty being paid out is 50 dollars. The total amount of bounties that have been paid out thus far is $8,000,000, where $600,000 of them have been awarded during the last 90 days, a time frame where they have received 676 reports. 5763 reports have been resolved in total, with an average bounty range of $394-$500. A total number of 1291 hackers have been thanked during the lifetime of the program (Hackerone (2020e)).

3.6 Lime's Bug Bounty Program

The urban transportation company Lime, which offers mobility services such as electric scooters and bikes in large cities across the world, are running a managed bug bounty program on of the other large platforms Bugcrowd. The program has in total rewarded 11 vulnerabilities, and point out that a validation of a report will made within four days of submission with an accuracy of 75%. This means that three out of four reports that are submitted will be accepted or rejected within four days.

Target Name	Type	Known issues
https://apps.apple.com/us/app/lime/id1199780189	iOS	2 unique issues
https://play.google.com/store/apps/details?id=com.limebike	Android	18 unique issues
https://api.lime.bike	API	3 unique issues
https://webviews.lime.bike	API	2 unique issues
https://juicer.lime.bike	API	1 unique issue
https://admintool.lime.bike	Website	4 unique issues
proxy-production.lime.bike	API	No known issues
https://ops.lime.bike	API	No known issues
*.lime.bike	API	1 unique issue

Table 3.7: The table shows the scope and what category each service is in.

3.6.1 Rewards

When reports are submitted to the program a process for prioritization and rating the findings is started. Lime's bug bounty program utilizes the Bugcrowd Vulnerability Rating Taxonomy for this process, however, they also explain that a priority may be modified if it's likelihood or impact justifies it.

The categories P1 to P4 each have their bounty range:

Technical Severity	Reward Range
P1 - Critical	$2,100-5$,000
P2 - Severe	$1,000-1$,250
P3 - Moderate	$450-600$
P4 - Low	$100-200$

Table 3.6: The table shows the priority groups and their corresponding reward range.

The final category P5 which is the lowest impact category does not receive any rewards for submissions.

3.6.2 Scope

The company's bug bounty program includes all their major targets in the scope, including the rider apps, backend APIs, and the web applications that support their daily operations. A minor description of the system's functionality is included, so the hacker can get to know the target a bit more. A table with in scope items is also presented:

And a table of out of scope items:

Target Name	Type
https://li.me (hubspot)	Website
https://help.li.me (zendesk)	Website
https://*.li.me	Website

Table 3.8: The table shows out of scope items.

The program only authorize testing on the targets in scope, but does also specify that if the hacker believes he has identified a serious vulnerability on a system that is outside the scope, he should report it or verify it with Bugcrowd's support team.

Policies

The policies are two sentence that forbids the researchers to perform behaviour that is disruptive, accesses users' private information, endangers users/the public, or is in any way harmful. If the researcher uncovers a vulnerability he believes will cause any of those issues, he is to stop testing and report them immediately.

Credentials

The security researcher needs to use their own credentials when testing. These can be acquired by self-registering for the Lime rider/juicer applications found on App Store and Google Play.

Safe Harbour

Lime bug bounty program operates with a safe harbour philosophy. This means that when researching for vulnerabilities in accordance to the policy of the program, they will assume the activity to be:

- Authorized in the accordance to United State laws and regulations, and the company will not initiate or support legal action against the researcher for accidental, good faith violations of the program's policy.
- Exempt from the restrictions of Lime's Terms and Conditions that would otherwise interfere with the vulnerability research activity
- Lawful, and helpful in improving the overall security of the company.

They also expect the researchers to always comply with all applicable laws for the individual (Bugcrowd (2020b)).

Chapter 4

Semi Structured Interview

4.1 Visma

The interview with Visma was performed over Google Hangouts with two representatives from the company. The representatives were at first presented with some general questions about their bug bounty program. Initially most of the available information about Visma's crowd-sourced security showed that the company had a Responsible Disclosure program, and a part of the company located in Finland, Visma Enterprise Oy, had previously hosted a public bug bounty program through Hackerone (Hackerone (2016)). Due to the low detail in information the program presented, my expectations were not high. After a short round of introductions, it was established that the Responsible Disclosure program was not the intended bug bounty program, and that the program was only hidden from public through Hackerone's private bug bounty settings and was far more comprehensive than I first assumed.

Visma's bug bounty program started with the Finnish department hosting a bug bounty program through Hackerone. After seeing the results from that program, a decision was made to start a program for the entire world wide enterprise. The program is a private bug bounty program where an invitation is required to participate. When asking the reason for going for a private bug bounty program, several reasons were explained:

- A private bug bug bounty program allows Visma to select the participants for the program. This was done in collaboration with Hackerone, where Hackerone provided some participants they recommended, as well as Visma brought in theirs, where invitations were given to hackers that they were familiar with, either from bug bounty programs or penetration testing, or hackers reached through Visma employee's social network.

- Most of the services available in the scope of Visma's bug bounty program require user accounts to be provided to the security researchers. In a private program, it is possible to restrict this process by restricting the number of participants of the bug bounty program, which will keep the hackers from losing patience.

To host Visma's private bug bounty program, they had chosen to use a BBaaS-provider, where only two companies were realistic providers; Bugcrowd and Hackerone. After discussions with both companies, Hackerone ended up as the winner for hosting Visma's bug bounty program. The main reasons were price, Hackerone's large pool of penetration testers, and the previous experience from the Finnish bug bounty program.

Hackerone has two options for private programs; unmanaged, and managed, where Visma had gone for the unmanaged program. Although a managed bug bounty program takes care of almost all the logistics involved in bug bounty programs, Visma chose to do most of the process themselves. This allows them to do all the communication with the penetration testers. This involves receiving reports, evaluate them, if something is unclear ask for clarification, and finally triaging the vulnerability. After triaging, the hacker can be paid. One of the main reasons for choosing an unmanaged program was that it is cheaper than a managed one. However, the people working at the SOC-department, the section of Visma handling the program, also wanted to do it like that because they consider it to be a fun, and giving to work closely with some of the world's best bug bounty hunters.

Hosting the program requires resources, where there are always two members of their Security Operations Center (SOC) on duty for receiving bug reports. The SOC-members work shifts, and divide the responsibility among six team members, that also have other tasks within the organization. Among the most important tasks is to respond to the reports that are received. By being quick to respond to submissions, the hackers feel valued, and gain a positive impression of the program. Visma had experienced in only a few cases where submissions lacked information for triage. In these episodes a dialog with the hacker is held until sufficient information is obtained, and the triage can commence. In the process where submissions have been received, there were two measurements that were valuable; ”time to triage”, and ”time to bounty”. To keep up the interest of bug bounty researchers it is important to keep these as low as possible, and a goal at Visma is that the hacker has received the bounty within two days after the vulnerability has been validated by Visma's security team that handles the bug bounty program.

Regarding the structure of the company, this validation process has a few steps. Visma is divided into several teams, where each team is responsible for several assets. In the one year duration of the bug bounty program, a total of 40 assets divided amongst 17 teams have been integrated in the bug bounty program, where an asset is a domain name for an application. The on-boarding process for getting a team ready for the program is handled by one person at Visma, which works with preparing teams as a full time job. This involves finding teams that are interested in being a part of the program, making them understand what it means and requires, and ultimately has control of whether the requirements are fulfilled or not. The most important requirements the teams need to satisfy are:

- The asset that is being launched needs to undergo a penetration test, to ensure that the solution is somewhat matured regarding information security.
- All vulnerabilities uncovered during the security audit need to be mitigated before the asset can be launched in the program, avoiding the outcome of having to pay for vulnerabilities that are already known by the Visma developers but have not been mitigated yet.
- The team needs to be motivated to mitigate all vulnerabilities reported from the

program within 90 days of discovery.

Whenever a new asset is placed on the program, a surge of reports are normally received in the first two weeks. There is a competition amongst the researchers to find the bugs before the others, and on average 10 reports are submitted for every new asset. Within three weeks, most of the vulnerabilities are found, and the activity on the asset dies down. Due to these surges, Visma only introduce one new team/asset per week to maintain control, and underlines the importance of control.

I then asked them if they had ever considered to host the bug bounty program themselves without anyone's help. This was not something they were thinking of, where some of the reason was that they had tried with the responsible disclosure policy. The policy had resulted in some reports, but the quality was lower than what they now were receiving from the program. The researchers were most often students, or individuals living in far off countries. The students delivered worse reports, and the far off researchers were difficult to reach when they wanted to send out swag, such as t-shirts. Hackerone solved these problems, the hackers there were skilled and professional, and Hackerone handled the payment process where this could be done by the click of a button, Visma pointed out.

Regarding the reward decision process, one of the items they had done was to have a table present at Hackerone's portal that showed four different impact levels with a corresponding max reward for each level. Although deciding the reward of a vulnerability most often was simple and done quickly, it was not always this transparent. Some vulnerabilities might normally be connected with high rewards, such as code execution. It had happened that they had received a code execution vulnerability with a low impact, and thus could not be rewarded with the highest bounty. Visma then continued that sometimes they even had to perform a vote internally to pin point the reward of a vulnerability.

When asked if they were satisfied with the program, they were very satisfied. The people managing the bug bounty program were not the only ones being happy about the program, Visma regularly received positive feedback from the hackers, and from the teams of developers that had been included into the scope. Some of the points they were satisfied with were the time to triage and time to bounty, which they said were the most important numbers for a successful bug bounty program.

4.2 Bug Bounty Hunter One

The interview with bug bounty hunter number one, who wanted to remain anonymous, was held during a meeting in Oslo in January 2020. The bug bounty hunter has worked as a penetration tester for several years, and has been active as a researcher on Hackerone since 2015, where he has earned most of his bounties. While hunting bounties, he has recently also been a part of managing a bug bounty program that during the interview became apparent also were hosting their program through Hackerone. While also being employed in a day job, bug bounty research has been a good and lucrative hobby, where through Hackerone he has had over 20 reports approved in private bug bounty programs, and additionally 5 bugs trough public bug bounty programs. Regarding motivational factors, one of the most important ones is that he considers it very entertaining to hunt bugs. Information security is a field that he takes great interest in, and bug bounty hunting is a

great way to help secure the world, as well as try out new attacking techniques, and the monetary reward also helps. He has also disclosed vulnerabilities to companies that do not have a bug bounty program, some not even a responsible disclosure program, all just to be a Good Samaritan with a passion for information security. When starting up a bug bounty program, a factor that can create a great deal of distress are invalidated reports, that can possibly overload the people receiving bug reports. Among these, the bug bounty hunter has only had one. He reported a bug that he had deemed a security vulnerability, but the company answered that they accepted this risk and saw it as a feature. Although they disagreed, the bug was accepted but no payout was made. When asked what he thought about the interaction with the company, he meant that the company handled the situation well. By accepting the bug instead of rejecting it, the hunter's reputation on Hackerone was intact, and the company didn't have to pay for anything that they did not want to fix. So far the biggest bounty he has received was a payout of 1000 dollars. When asked if he considered the bounty to be reasonable, he said that considering the amount of work he had to do to find the bug, it was generous. However, due to the massive impact that the bug carried, it was a fair amount. In fact one of the reasons he chose to look for bounties for that specific programs were the high bounties, and lists up several main criteria for why he engages in a bug bounty program:

- The program has a big scope. A large attack surface increases the probability of finding bugs.
- The max bounties are high, 3000 dollars is a nice max bounty.
- The program is known for its short response time.
- Time to triage needs to be short
- The bug needs to be resolved within a reasonable amount of time. On Hackerone, only parts of the reputation is handed out after triaging, the rest after the case has been closed.

Regarding the last point, the hacker has mixed experiences. Some bug reports have remained opened for 10 months, which is a frustratingly long time for a vulnerability to be resolved, and some bugs are mitigated and closed within 90 days, which is the usual period and something that is desired.

The bug bounty hunter has also been on the other side of the program, being a security resource on the team of an international corporation, where he dealt with incoming bug reports, deciding the bounties, and communicating bugs to the software developers. Later on he also managed the budget as a stand-in manager. When setting up the bug bounty program they were in dialogue with two different bug bounty as a service providers:

- Hackerone
- Bugcrowd

After considering all three, the company ended up with a private bug bounty program hosted at Hackerone. The decision was made due to it being a familiar platform, that someone on the team had had previous experience with. When asked why they went for

a private bug bounty program, the answer was foremost to stay in control of the program, as well as to avoid too much attention. Starting up a bug bounty program presents a whole array of uncertainties. How many bug reports would they receive? Would they be able to respond in a timely fashion? Would the team of developers be able to mitigate the bugs in time? A private bug bounty program allowed the company to stay in much more control of these problems than a public bug bounty program would. They were also in control of inviting the hackers they wanted to participate in the program. Ultimately a private program allowed a controlled launch.

In regards to deciding the scope of the program, several factors were in place. First and foremost applications that they felt were ready for the program were inducted into the scope. A program that was ready, was mainly a program that had been tested where no internal security bugs were open on the application. However, due to lack of resources for internal penetration testing, applications that they wanted to test the security on were also attempted to include in the scope. At times these applications were generating a lot of bugs, and the company then chose to take the application out of scope to let things cool off for a while, until the bugs could be fixed, and they were up to speed after a short while, and the application could be reinstated in the program. One might think that changing scope to reduce incoming reports might result in a bad reputation amongst the hackers participating in the program, but the Norwegian bug bounty hunter did not experience any negative feedback in doing so, as long as they were running a decent program. In fact, several things were specifically done to keep the hackers happy:

- It was important to respond quickly to reports. A common consensus is that programs that have a quick response time are preferred by bug bounty hunters.
- A non-monetary reward, swag, was sent out to those who they wanted to reward a little extra. Such as finding an interesting security bug.
- Hackers who argued about the impact of their bug were locked out from the program if this was a repeated offense. They create extra work, and provide a negativity that you just do not want in your program, so best way is to just get rid of them. He underlined that healthy arguments about bugs is good for the program, but troublemakers that repeatedly argue without substance is not.
- It is important to be willing to cooperate. Sometimes bounty hunters ask for something that is not normally provided or allowed through the program, such as asking to talk about a bug they found on a talk at a security conference. A process had to be performed with the company's legal team, but was sorted out in the end.
- Sometimes, bugs that did not fit into the scope of the program, or were according to policies were received. A nice gesture was to grant a small bounty, often 50 dollars, if the work was still good or somewhat helpful.
- Maintain a positive language when communicating with hackers. Say thank you, and be appreciative of their work.

When asked if they made any changes to the program while it was running other than taking applications in and out of scope, was alter the policies of the program. It is not

possible to know for sure what to expect, and what you receive from a bug bounty program, so it was important to asses the program while hosting it. Sometimes bug reports would be submitted with something that the company wouldn't really view as a risk, or valuable information, so these types of bugs would be added to a policy for vulnerabilities that would not grant a bounty.

In total, they were very happy with the results that the bug bounty program gave. Although it was only considered a supplement for the already in place penetration testing, the bounty hunters did provide bug reports on vulnerabilities that the company didn't find themselves.

When speaking about bug bounty programs in the Norwegian market, the bounty hunter mentioned some areas where improvement could be done to increase the popularity and impact of them. Openness is required for it to gain traction. If companies that host bug bounty programs have positive experience with them, they should be more open about the program. This will show that it actually works for companies, and others will then follow. However, the bounty hunter is rather sceptical to if it can replace regular penetration testing, and sees it more as a supplement. Nonetheless, he does believe that companies that can afford it, should implement a bug bounty program to increase the baseline of their security further. This does require a passionate soul in the company, that has a general understanding for information security. It is important for a company that has a bug bounty program to actually understand the reports that they receive, and as well be able to calculate what impact the vulnerability carries.

4.3 Bug Bounty Hunter Two

The interview with Bounty Hunter Number Two was held in Oslo in the beginning of February 2020. He had been participating in bug bounty programs for over three years, where several bugs findings at Google had earned him place among the top 20 hackers at the Google Hall of Fame, and was paid Vulnerability Research Grant on multiple occasions throughout the year. The bug bounty hunter was also working as a full time penetration tester at the time.

When asked about what programs the hunter had participated in, he had spent his time focusing on two programs; Microsoft and Google. Due to his full time job, time was a scarce resource, so he wanted to focus on the programs that he enjoyed the most. Among the two he had worked on, he had the most positive experiences with Google, where the reason was:

- They are very open.
- They are effective, this includes time to triage, and time to decide rewards.
- They are easy to report to. A single point of contact for submitting bugs.
- The Vulnerability Research Grant.
- The hacker community. The top hackers are invited for a social gathering, to network and to learn more about security.

He further explained that Microsoft are somewhat similar, however they take longer, are less efficient, and when comparing bugs, the rewards are slightly lower.

When asking if he only does bug bounty programs, or if he also participates in responsible disclosure programs, he said he mostly did only bug bounty programs. However, he had submitted bugs to companies where he knew they were not handing out rewards. These bugs were mostly results of hobby projects a while ago, or vulnerabilities he found at other companies, that were applicable for the vulnerability disclosure program. He then pointed out that if he was going to spend a fair amount of time on looking for vulnerabilities, a considerable reward at the end is preferable. Something else he wanted to point out was that a company needs to be open for receiving vulnerabilities, for example by presenting the ”security.txt”-file on the company web pages. This is a file with instructions for how and who to contact someone in the organization to submit a vulnerability, and if something, what the company is willing to offer in return. Most important of all is that there should be no negative impact when reporting a security vulnerability.

For the hunter to participate in a company's bug bounty program, he also explained that a the hunting had to be a challenge for him. He wanted to test systems that he normally doesn't get to test in his normal job as a penetration tester. Google was one of the organizations that was most mature regarding information security, and seemed to be a perfect fit. If he was to find something at Google, he had to come up with a creative vulnerability. This ultimately ended up being the case, where he found a vulnerability that was applicable on several services ending up earning him over 18 000 dollars in bounties.

I asked him if he ever had considered Hackerone as a potential source of income. All other three interviews had been very weighted towards Hackerone. He then answered that it was natural for them to be a focus point due to their size, and that he was registered, and had a few invites to programs. However, he had not found anything to report yet, mainly down to a shortage of time. With a full time job, and a busy life in general, there was not too much time left for bug hunting, and with a regular grant from google being available throughout the year, the time was spent mostly targeting services at Google. When I then asked if he thought this was an approach the company took benefit in, and not just an unnecessary expense, he was certain that it was. In fact, most of the vulnerabilities he had found in their program recently was down to the grant program. Another positive thing he could think of with Google's program was they they often upgraded the reports. When submitting a report a hacker usually set a risk level on the vulnerability. If Google then discovered that the bug was more severe on their end, than what the hacker could know, they would upgrade the risk and increase the reward.

Invalid reports can happen the best, and had also happened to bug bounty hunter number two sometimes with Google. On one occasion he had reported a duplicate which then was invalidated, but most often the reports contained bugs that were security bugs, but was not severe enough for the company, and some times he received a blunt answer that it was something the company did not want to fix because they were confident that the risk was mitigated elsewhere. Reporting invalid bugs was something that affected the hacker's rating on Google Hall of Fame negatively in the past. These downsides were later removed when they realized that there was a risk of a bug not being submitted out of fear of getting an invalidated report. He further explained that Google were a special case when it came to resources, and it would be hard to do the same for a smaller company where resources

are scarce.

When speaking about bug bounty programs in Norway, he did see a future for it. It was only a matter of time until it would be a trend, it had already proven to find more vulnerabilities than traditional procedures. Bug bounty had grown very much on the international stage recently, and when Norwegian companies become more mature with respects to information security, more organizations would begin with bug bounty programs. However, a few issues he could point out were the lack of resources. A company that wants to perform a bug bounty program would need a person qualified to receive reports and triage them. This could be solved by having a managed bug bounty program at one of the big service providers. He mentioned that competition was increasing in that field, which could lead to cheaper services. Another issue, if a company hosted a private program themselves, would be to reach out to hackers, a problem also solved by buying bug bounty as a service. If these problems weren't solvable, the minimum effort a company could do would be to at least have a responsible disclosure program.

The interview had covered a lot of information of what a company should do when hosting a bug bounty program, so I asked if he had some pointers on how not to handle a bug bounty program. He had not experienced anything extraordinary from companies that had a program, they are often eager to process anything the hackers report to them. However, he had experienced from large companies that should have at least had a procedure of processing vulnerability reports. The hunter had worked on a little side project out of curiosity, and found a vulnerability and uncovered a bug. The company had thanked him, and said they were fine with it, and nothing really happened. The worst thing though he could think of would be if a company threatened with legal actions if they received a bug, it is therefore best to have a "Safe Harbor"-policy, where vulnerability research is accepted if it is with good intent.

The hacker also had some good points when comparing traditional penetration testing with bug bounty programs. Ultimately, for ideal penetration testing, the pentester has a lot more information about the system. The tester has access to logs, source code, direct communication with developers, and additional information about the system from the customer. Whereas a bounty hunter only does black box testing. "They end up uncovering completely different bugs", he said. Another aspect was that he could not see any way where a bug bounty program could test internal infrastructure of a company, or systems from third party providers. He then concluded with that bug bounty programs would rather be a supplement to strengthening an organization's security, and partially replace penetration testing. "There is no assurance that the system is sufficiently tested", he continued. A company would want to be assured that every part of their system has been tested. He also said that he was positive to bug bounty programs because a company would receive attacks more attacking techniques. "One very good penetration tester can never match all the techniques of 50 bug bounty hunters".

The interview ended with him stating that he thought bug bounty programs would be a big part of the future, but for companies to start with it, all pieces need to be in place before starting a program.

4.4 Finn.no

The interview with Finn held at Finn.no's corporate offices in January 2020 with the one person who was solely responsible for managing their bug bounty program. Finn is a company that hosts a digital market place for people and businesses, where it is possible to search for jobs and travels, as well as sell second hand items that no longer are needed.

It was early on established that Finn hosted a private bug bounty program through Hackerone. The reason for this was that the bug bounty program as a service providers are normally similarly priced, but they liked some of the philosophies Hackerone had. One of them being the private disclosure policy. This policy allows vulnerabilities to be disclosed to the participants in the program in a private manner, allowing the hunters to view what bugs are found and how much the others were paid. Ultimately creating a transparency within the private program that generates trust between Finn and the hackers.

The program was initiated September 10 2019, where the scope in the initial phase was only the main website "finn.no", which is a gigantic platform with several micro services. However, the scope was not instantly well received by the hackers, where several of them declined an invitation through Hackerone due to a small scope. They started out small, and invited six attackers, and then 10 more, and after a month had 16 active participants in the program. Further underlining, that this was to perform a slow start, where you assess what the company can handle before you increase the scope of the program further. He wanted to be certain that he could handle all the reports they received, and that the team of developers were able to fix the bugs in a timely manner. The reports are disclosed when the report is closed, allowing bugs to be open for long increases the probability of duplicates occurring, which ultimately leads to dissatisfied hackers. He also said that the hackers are very satisfied with their response and resolution time, and that there are programs in other companies where bug cases can be open for months at a time.

An important focus that finn.no has on their bug bounty program is actually to fix the bugs very quickly. Although they do not have a fixed time, due to it also being contingent on the severity and complexity of the bug, the average time until a vulnerability is mitigated is only six days. One of their most critical bugs only took half an hour to mitigate. Emphasising that even bugs with medium and low impact are fixed as soon as possible to avoid duplicates that cause bad reputation. Being quick to respond to reports and fixing security problems quickly is not the only thing they do to run a successful program, he further described. It happens that they receive reports about vulnerabilities that are not covered by the scope. An example that he presented was that a hacker had reported a setting Finn had set on a repository on Github, an arena for sharing and developing source code. The hacker reported it as a security bug, and Finn agreed. Although the github-profile was not in scope, the bug was fixed, and the hacker was paid handsomely. The reason for doing so, was to operate in a manner that they themselves regarded as fair, and in an attempt to not make the hacker feel used or fooled. Underlining, that in their program, they do everything they can to keep the hackers satisfied.

After increasing the scope in January to include every domain, subdomain and IP-address currently in control by Finn, and inviting new hackers to participate, the program has 42 accepted invitations from 56 in total, where 32 are currently in the program. When asking about the demographics of the participants and if someone were from the Nordic countries, he said that four Norwegians were in the program as well as eight to ten Finnish

hackers. He elaborated further that they had invited a few Finnish hackers after contact with people working with bug bounty programs at Visma. These hackers were so impressed with the response time, feedback, that they invited their colleagues to join the program. When asked if there was any impression left from Nordic hackers, he said the communication with them had been a lot more pleasant than with the other hackers, and on the plus side they deliver very good reports.

However, inviting bug bounty hunters from Finland was not the only way they received participants, utilizing Hackerone's algorithm for inviting bug bounty hunters was also used. Through this algorithm, they experienced that only inviting the hackers among the top 100 ranked on the platform was not enough. These often declined the invitation due to a small scope and it would be difficult to invite them again in the future. He further elaborated that they went for the hackers ranked among the top 1000. Not only do they stay for longer in the program, they are also more enthusiastic, and thankful for what they get.

When asked if they had a big team working with the bug bounty program, he said that it was only him, and that it was sufficient for a company of their size. His job was to communicate with the hackers, and communicate with the software developers responsible for the service that the security bug covered. He further said that after a while, processing the reports did not take that long, as the reports started to get familiar after a while. In fact the most requiring part of the bug bounty program was to work with extending the scope, at setting up the program. He also said that inviting new hackers to the program resulted in more work. New hackers meant new types of vulnerabilities, hence new types of reports, and that it takes some time to digest the new information and to reproduce the vulnerability. However, due to his efficiency he felt that he had a lot of room to go on before the response time would get to a point where the hackers would react negatively. Even Hackerone's full managed bug bounty service normally takes two days to respond, and himself was nowhere near that. He further explained that the reports normally come in bulks, and even if it takes time to reproduce the bugs, and triage them, just a quick reply, thanking them for the report would be sufficient in the initial phase.

I then asked him, what the managed bug bounty did, since it seemed like he did all the work himself anyways. He explained that the managed part was a nice safety to have for week-ends and such, but he would consider going for an unamanged program in the future.

Regarding the question if they have received many invalid reports the answer was none, and only some duplicates had occurred. Only quality reports were submitted to them, and he stated that this was due to the fact of having a private program on Hackerone. The hackers are experienced, and only the top 1000 ones are invited. If they deliver a bad report, they would earn negative reputation, which could hinder them from being invited to future programs, he elaborated.

What they did to counter the duplicates was to use a setting called "private disclosure", which allows all members to that are part of the program to view reports that are submitted by other hackers. This provides a transparency in the program, ultimately gaining more trust from the participants. "Some other programs might have a private private program where nothing is disclosed", he continued. He did not like the idea of this, because it allows the possibility to fool the hackers that submit bugs. Further explaining that a scenario

might arise where a hacker submits a report, gets the answer that the bug has already been uncovered by someone else, either another hacker from the program, or someone working at the company. The company ends up not having to pay for a security bug, but still gaining valuable information that they can work with.

Another topic that arose, was the topic of bounties. Finn awards 3000 dollars as their highest bounty, which is handed out for critical vulnerabilities, and 1000 dollars for vulnerabilities categorized with high vulnerability, further on it is 200 for medium bugs, and 100 for low. When considering what to reward the reports they receive, most of the rewards are based on what previous vulnerabilities were rewarded. However, the ultimate deciding factor was impact. He explained an example where one vulnerability regarding HTML-injection received 200 dollars, and another one covering the same category, but with a much higher impact was rewarded 1000 dollars. On a question whether he thought the bounties were high enough, he said that he thought so, pointing out that all hackers declining or leaving the problem had commented that the scope was too small.

Finn also had a long history with penetration testing their software, but had recently shifted their focus on more towards the bug bounty program. In 2019 they had performed four penetration tests, including one new service had not been tested for several years they wanted to include in the scope for the bug bounty program. This was done by a provider that uses bug bounty hunters to perform security audits to test if there was any point in security testing services before adding them to the program. The results they received from the security audit was not anything that they would not be able to handle by just including the service in the scope. He then continued by explaining that Finn would need approximately 10 penetration tests a year to keep their security at a decent level. With a bug bounty program that would be reduced. However, he did underline that there are different premises when comparing bug bounty programs versus security testing, but by comparing dollar cost per vulnerability, bug bounty programs are a lot cheaper.

Something else that Finn are focused on regarding their transparency, was that everyone in the company that wanted, had access to the program portal. So that they can see the reports that are submitted. This had created a positive atmosphere where developers gained a better awareness towards security, as well as gaining valuable insight into security bugs. As well as if internal resources in the company found security bugs themselves, they had to upload them to the program, so that the hackers could see them in case they handed in a report that was a duplicate.

Finn also had some future plans for their program. When viewing the hackers' activity, they had observed that the hunters only discovered 20 of approximately 120 services on the platform. He wanted to counter this by creating a communication channel with the hackers, where he could help them by giving additional information about the system, or even information about software changes that were implemented that the hackers could set their focus on.

The interview was rounded off with his views on bug bounty programs versus traditional penetration testing. Where he pointed out that an unmanaged or managed program hosted at Hackerone would be expensive for small companies, due to a yearly fee of 20 000 to 30 000 dollars, and then the company needs to pay bounties on top of that, where Hackerone also takes a fee on the bounties. With these amounts, a company can purchase a very thorough penetration test for that sum. However, for Finn.no they would require

around 10 penetration tests, would be a lot more expensive. Through the program they received 50 individuals testing their systems, instead of two penetration testers. Ultimately they liked the diversity that several hackers brought.

Another important aspect that he pointed out, was that for a successful bug bounty program, you need a driven person. The individual needs to want to work with the program, due to reports that can occur at all the hours of the day. The one responsible for the program also needs to the qualified for the program. This involves being able to reproduce the vulnerabilities that are reported, communicate these with the team of developers, as well as be able to understand and estimate the impact. In fact this issue was one of the factors that the company owning Finn, Schibsted, did not implement the program for more of their services. They lacked people with the skills and passion for working with a bug bounty program.

Chapter 5

Analysis

The literature review and the semi structured interviews point out several similarities that should be present in a bug bounty program. In this chapter the results from the literature review and interviews will be discussed and analyzed the results from the two research methods, to uncover the the common features that good bug bounty programs have or should have. The section will also enlighten the other points from the master's thesis.

5.1 The Economic Impacts of Bug Bounty Programs

One of the points for this assignment was to attempt to answer if a bug bounty program would be a viable solution for securing a Norwegian company's computer systems without having to pay large amounts of money. The companies behind the bug bounty programs presented in this thesis are mostly grand, international companies where the security teams have large financial backing. While the companies in question are small to medium sized Norwegian businesses (SMB) that want to be as secure as possible with the means they have available.

When implementing a bug bounty program in Norway, the two Norwegian companies and the international company bounty hunter one worked for had both chosen to go for a private program on Hackerone. Finn also had the program managed by them, which costs even more. Both companies were also in discussions with Bugcrowd, but ended up with Hackerone due to familiarity, since both companies were evenly matched considering price. The price for a hosted bug bounty program was estimated to $20,000 for an unmanaged program and $10,000 extra if the company went for a managed, in yearly fees. So before a company has even started getting reports, an amount equal to approximately two thorough penetration tests (according to bug bounty hunter two, who also works as a penetration tester) has been paid. On top of that, bounties for up to $3,000 are paid for vulnerabilities, plus a cut of 25% extra for each bounty rewarded.

Bug bounty hunter two also pointed out that for SMBs, the maturity of security would not always be at the highest level, so that a penetration test would maybe would be more rewarding from a security point of view. So, that for a bug bounty program to be an

excellent choice for securing the company's software or systems, it needs to be cheaper than $30,000 a year and some sort of maturity level needs to be reached. Services such as Hackerone and Bugcrowd are out of the question, and the company needs to host their own program.

5.2 Public Or Private Bug Bounty Program

The research has shown that there are normally two ways in how to perform a bug bounty program, public and private. Finn, Visma, and the company that bug bounty hunter one worked for all chose a private program. However, companies such as Facebook, Google, Lime and Verizon Media, all have public bug bounty programs that handle reports from several hundred hackers every year. To help restrict all the reports the companies were getting, a penalty for reports lacking content or even lacking a vulnerability was used. Later on Google discovered that this might discourage hackers from submitting relevant vulnerabilities, and decided to remove these penalties. To compare what works for companies such as Google with what might work for a Norwegian SMB is not ideal. Google has massive amounts of resources, where as resources for the Norwegian companies might be scarce. On the other hand, the interviews uncovered that invalid reports were not normal for their bug bounty programs run privately through Hackerone, where only duplicates occurred from time to time. Another aspect to take from the interviews is that they were very uncertain on how the program would impact the operations of the company. Would they be bombarded by reports, unable to follow up on them all, and eventually spend way too much time on responding and paying more in rewards than they could afford. What seemed to be the common factor from the interviews was that when launching a program, control was the most desired factor. To achieve this control, the decision to go for a private bug bounty program was made.

Several advantages are gained by choosing a private bug bounty program, one of the factors that you gain control over is the participants. One example was through Hackerone where companies can set a threshold for what rating a hacker should have to be invited to their bug bounty program. A perk with this is that although the number of hackers participating in the program are few, they are very skilled and will hand in great reports.

Another positive aspect is that information can be shared with the participants, that the company might not have shared in a public program. An example for this can be what vulnerabilities were uncovered. It is not given that a company is comfortable with sharing what vulnerabilities existed where in their system to everyone, but by limiting the participants and sharing the information with them several additional upsides were mentioned by Finn. Not only does the hackers see what types of vulnerabilities or systems they should be searching for or looking at, it gives a transparency that creates a bond of trust between the hackers and the company.

Bug bounty hunter one pointed out another good reason for choosing a private program. If hackers start to make trouble, or argue on decisions that the company made, the easiest way to solve the situation was to remove the participant from the program.

However, one major problem arose during the thesis for attempting to build a private bug bounty program without relying on BBaS. The process of recruiting hackers to the program might become difficult. Whereas on Hackerone and Brugcrowd more hackers

Risk	Proposed Reward
Critical	$500
High	$250
Medium	$100
Low	$50

Table 5.1: The table shows the amount each impact category should be rewarded.

can be directly invited, and with a public one anyone is invited, the issue that needs to be solved is to get participants without these options. Something that can solve this in some degree would be to ask the companies such as Finn and Visma for help. Both of these companies had directly or indirectly reached out to hackers, that were now participating in their programs. Finn had even received contact information to a Finnish hacker, which had resulted in several hackers from Finland being part of the program. Another way that can potentially solve this issue partially will be to create an attractive program for hackers, a program that they want to become a part of.

5.3 Rewards

Although bug bounty hunter two reported vulnerabilities to companies that did not have any sort of reward system for security bugs, he made it very clear that for him to spend time researching vulnerabilities on a company's systems, a decent reward should be the end result. Meaning, that rewards need to be well communicated to the hackers in advance. However, deciding how big a reward might be difficult. Where Google operates with a max bounty of $31,337, and Facebook has paid up to $50,000, the programs studied through the interviews operated with a bug bounty for up to $3,000 for their most critical vulnerabilities. There was also no indication that this level of rewards was unsatisfactory, and that rather the higher rewards at the big public companies are due to the heightened competition bug bounty hunter two was drawn to. This means that a roof over what the reward to the SMB's most critical vulnerabilities should be is $3,000. However, the SMB's might have a more restricted budget where a bounty payout of $3,000 can be demanding. The research showed that there are no perfect way for a bug bounty program to start off at. All the programs had started at one point, and had to change the information on the go to further develop the program to a point where the results became satisfactory. An example is the information on the program that specifies what is especially not in the scope, where reports about vulnerabilities would not be accepted. This is information that the companies had to add later on, when gaining experience. An approach like this can also be used for deciding a reward level. The SMB should set a level of rewards that they can handle, and then change this after a while when experience has been acquired. For a company with limited resources, a proposed reward frame is:

This is only a proposition, and should be assessed to accommodate the company in question.

5.4 Scope

The scope is the part that explains to the participants what they are allowed to perform testing on, and varies from company to company. A scope can divided into multiple sections, such as mobile applications, web applications and hardware, where large companies have a very comprehensive scope. For small businesses, the possible scope might not be too extensive. From the interviews, I learned that a controlled launch was something the companies wanted, as well as bounty hunter one underlining that control needs to be kept throughout the bug bounty program's lifetime. The best way to stay in control, is by keeping control of the scope. Visma stayed in control, by having a thorough procedure for preparing services for being included in their program's scope, where new services were added when ever activity died down in the program, and they were ready to handle new surges of reports. For the SMBs in question, that does not have the ability to penetration test systems before inducting them into the program, this procedure cannot be completely followed. However, it is possible to control the size of the scope, where the company should start as small as possible and then proceed with increasing the scope. An example for a starting scope might be one single endpoint on their web-page, before gradually increasing to the whole web-page being in scope. When activity is under what the company is able to manage, new items can be added to the scope, such as mobile applications for the same service, or sub-domains used the website. One might ask if it would be responsible to put items that have not undergone penetration testing into a bug bounty program. Finn did an experiment on this, and decided on that the differences were not that severe, where future items would be added without performing testing on before hand. The company also need to remember that they are in charge, and that if situations go a bit out of hand, bug bounty hunter one taught us that it was not problematic to remove items from scope to regain control.

5.5 Out of Scope Items

A section of information about everything that is not wanted in the program is also a recurring feature in bug bounty programs. This is typically an area where the company can post vulnerabilities or reports that they do not have use for, or domains that they are not that interested in regarding risk. Some items seem to be covered by multiple programs, such as denying vulnerabilities:

- Regarding social engineering
- Regarding third party services
- Requiring unlikely interaction from the user
- Regarding disclosure of information that does not present significant risk.
- CSRF-attacks with minimal impact
- That does not affect the newest versions of web browsers.
- Issues that have been reported earlier.

- Regarding the availability of the service.
- Regarding version information disclosure without the version in question being vulnerable to attacks.

However, this list is not something that can be completed before the program is implemented, and should be extended throughout the program's life time. Whenever the company receives a report about something that they do not consider to be valuable information, add it to the list of items that are unwanted.

5.6 Program Rules

All the programs in the literature review includes a section where everything is made clear in terms of legal issues. This is also the part where you state the rules of the program that the hackers need to follow in order to qualify for the rewards. Some rules need to be specified in custom for the company in question, but some rules are very generic and can fit for multiple programs:

- The hacker is only to use his own accounts, or accounts that he is authorized to use for testing.
- Discovered vulnerabilities are not to be used to extract information further than what is the minimum requirement to prove the security flaw.
- If sensitive information is acquired by accident, it is not to be stored.
- Participants are not authorized to perform actions that may be disruptive, damaging or harmful to the company.
- The hackers are only allowed to perform testing on items that are covered by program scope.
- The researchers are not allowed to publicly disclose vulnerabilities without the permission of the company.
- The amount to be rewarded for a vulnerability is solely decided by the company.
- The hacker is to follow any applicable law for companies and private individuals in Norway when performing testing.
- The hacker needs to by best effort avoid violating other user's privacy when performing testing.
- The hacker needs to follow the program's guidelines for submitting reports to qualify for the maximum impact based reward.
- Where possible, the hacker should use identifiable input, such as his username and email.

Additionally, some companies might feel the need to reserve testing from people that are close to, or related to employees of the organization, such as Verizon Media has done. Due to the size small size of Norway, and the even smaller security community in the country, this might prove to be more damaging, than positive. The list stated above should not be a final list, and also here it is important to be adaptable and learn from experiences. The rules are meant as a framework to protect the company from difficult situations that might occur. However, a new never thought of before scenario might come up in the future, where a disagreement with a hacker might cause some turbulence. After handling the situation in the best possible manner with the hacker, the program rules should then be updated with conditions that will keep the same situation from happening again in the future.

5.7 Safe Harbour

One item that all programs should have is a section called Safe Harbour. This section will let hackers know that your company takes reports concerning vulnerabilities in the best possible manner, and that legal action will not be pursued if the hacker had his best intentions when researching

5.8 Reports

The end product that program will receive from the hackers are the vulnerability reports. From the interviews and literature review it is apparent that these reports are as complete as possible to give valuable information to the company. By receiving good reports, several advantages are gained:

- Less resources have to be spent on communicating with the hacker to gain the knowledge needed.
- The foundation for what to base the risk and impact calculations on are as thorough as possible.
- When reports on vulnerabilities that haven not been received before are submitted, it is easier to consume the new information if it is complete.

Although only Verizon Media and Facebook propose instructions to how they want the reports to look from the program's assessed, this would be a valuable effort for the SMBs. The program is new, and the experience is low, so even for an incomplete report the company might not even know what additional information to ask for. Visma had on occasions dialogues with hackers to obtain all the information needed, though their team was experienced not only in working with bug bounty programs, but working with cybersecurity in the organization as well. Finn on the other hand meant that the level of the hackers that were included in the program was so high that the reports were good no matter what, where Nordic hackers submitted some of the best reports. This shows that when the program consists of ideal participants, instructions for reports might be excessive. However, until that situation is reached, instructions should be present.

Reports that the company wants to be submitted should contain:

- Which part of the scope that the vulnerability affects e.g the domain, or mobile application.
- What kind of vulnerability is it.
- The title for the report.
- A thorough description of the vulnerability, where also the impact is stated.
- The tools that was used to reproduce the vulnerability, such as web browser and operating system.
- Other details the hacker feels would be useful.
- A numbered list of how to reproduce the attack, where each number is limited to a maximum of two sentences.
- Pictures that can help with the description of the vulnerability.

Some companies penalize hackers that do not follow these instructions by reducing rewards, or even removing the hacker from the program. However, due to the issue of gaining participants being a major problem to overcome, it would be recommended to hold back on such actions until the program is well established.

5.9 Communication

From the interviews with Finn, and both bug bounty hunters a valuable point became clear. The way that the company communicates with the security researchers is essential for the program becoming successful or not. First of all, submitting a report should be a straight forward procedure for the hacker. For platforms like Hackerone and Bugcrowd, this is done through a portal where the participants have access to everything they need. However, for small businesses, creating a web portal for the sole purpose of submitting reports would excessive, and a waste of resources. A simpler but yet as effective means would be to have a designated email address for submitting reports, e.g. ”bounty@[company domain]” that can be portrayed along with the other information for the bug bounty program.

Being able to receive reports in a simple manner is not the only course of action that a company should take. Both bug bounty hunters favoured programs where the company was experienced as being responsive, where a response within the first few days were a sought after trait in a program. Finn had taken this one step further, and usually replied within a few hours. A quality that was so positive for the program, that hackers went on and vouched for the program to their friends. If the SMBs were to copy this, it would benefit the program greatly in terms of participants, and it does not always mean that the report needs to be understood before replying. A quick reply will show of the image that this the program is something that the company takes seriously and are enthusiastic about, which will again lead to engaged and enthusiastic hackers that want to help with improving the company’s security. In fact, the worst thing the bounty hunters knew were companies

that took the reports lightly, attempted to come up with excuses for the issue, or even ignoring the matter completely.

Another point that became apparent through the interviews was how the companies focused on making the hackers feel good. They are human beings spending time researching the companies' systems, without being 100% certain that they will be rewarded for effort, so maintaining a positive language goes a long way. This includes being thankful, and showing gratitude for the work they perform, even if it ends up being somewhat a waste of time. Positive language helps, and the interviews also made it apparent that the company gains a lot by generally attempting to make the hackers happy as long as they can. This includes, being helpful when they ask questions, such as when the hacker approaches for permission to disclose the vulnerability. The immediate instinct might be to decline the request. However, when the bug has been reported, the hacker has been paid, and the vulnerability has been fixed and verified, declining the request might do more harm than good. Bounty hunter one even said that there were times their company had to consult and solve logistics with their legal team just to allow a hacker to disclose his findings in a conference talk.

5.10 Triage

When the report has been received, and the hacker thanked for their contribution in a response, the vulnerability needs to be assessed, and the risk and impact decided, a process the hacker expects to be done within a few days of submission. In the interviews, the companies revealed that they had a list of all previous vulnerabilities that the company had rewarded, which they actively used to decide the risk and impact of new reports that were submitted. The list combined with the table portraying the impact's value was sufficient for deciding how much the vulnerability would be rewarded. Whenever they would receive a new vulnerability, a more detailed and demanding procedure had to be gone through. A problem for the SMBs arises when this process starts. First of, when the program is new there will not be a list of previous vulnerabilities to base the future payments on, and the employee responsible for the program will have to conduct a lot of decisions based on his own knowledge of the company's system and the vulnerability. Second, the interviews uncovered that the person responsible for a program needs a broad understanding of information security, a competence that is a scarce resource. Finn even said that the lack of competent security people was one of the reasons for a bug bounty program not being established in other parts of their organization Schibsted. So for a program to be possible, a driven person with a passion for information security is needed in the company. The individual also needs to be trained in the process of deciding a vulnerability's impact for the company. Bugcrowd's Vulnerability Rating Taxonomi system where they categorize security bugs in severeness P1-P4 might be a tool that can aid in the process of triaging. However, as Finn pointed out that vulnerabilities are not always straight forward, when explaining about the case where the same technical vulnerability had ended up being rewarded differently by several hundred dollars due to the gap in risk and impact. A possible solution can be to make the hackers also submit the impact level they regard the issue to have, where this combined with with a Bugcrowd's framework, and ultimately the encyclopedia of vulnerabilities the company will get, can create a total basis for how to triage

bugs. The approach requires the ability to trust the hackers, which might feel controversial, but the company always has the final saying, and the company needs to keep it mind that the participants are ethical hackers that want to participate in the program to help secure the organization.

5.11 Payment

Both Bugcrowd and Hackerone perform the process of paying the bounties to the hackers that after the company has triaged the vulnerability, and decided on a fitting reward. If the program would only have Norwegian hackers, this might not be a problem where a regular bank transfer would solve everything. However, this will become more problematic with foreign participants, and a plan for executing payments need to be in place if this is ends up being the scenario. From the interviews, and the section about existing bug bounty programs, it was revealed that one of the factors a program can be measured in is the time it takes until the hacker gets his money into his accounts, and that most hackers prefer a quick process where the money is received within a few days of the bug being triaged. Although there are probably several solutions to this problem, this thesis has not attempted to solve this problem due to it being a feature that arose during the assignment. However, there are several platforms that are cheap to use, can can solve this easily, services such as PayPal. The most important thing to underline is that a plan for payment is required for a good bug bounty program.

5.12 Resolving Vulnerabilities

Throughout the research it became clear that a bug bounty program is not only about uncovering vulnerabilities, in a successful program the pressure to fix the security flaws quickly also exists. The incentive for bug bounty hunter was the reputation, only parts of the reputation on Hackerone was handed out after triage, and the rest after the vulnerability was fixed and the case closed. Although this concern regarding reputation should not be a focal point for the SMB's bug bounty programs, there are several other reasons for mitigating the vulnerabilities. The major reason is also the reason for the having the program. The vulnerability is discovered, and mitigated quickly, removing the risk of a potential exploitation. The possibility of receiving duplicates is also reduced, ultimately making the hackers more satisfied. From the interviews it became apparent that duplicates are troublesome for both parts. For the hackers it ends up being labor that they will not be rewarded for, and for the program it can end up in more work without increasing the security. If vulnerabilities are not disclosed in some way, and a lot of duplicates occur because of slow resolution time, it can also create distress among the participants if this becomes a pattern where the hackers suspect that the company attempts to fool the participants in an attempt to reduce expenses. From the existing bug bounty programs and the interviews, the common consensus is that from submission to resolution the process should take no longer than 90 days. However, if possible the companies should try to make this process as quick as possible. This is a trait that might even suit the SMB's in question. Small companies do not have the organizational structure of large companies, where changes to

computer systems often take longer.

5.13 Disclosure

Although the disclosure of vulnerabilities should be completely up to each individual company, the interviews uncovered several reasons for having some policy for disclosing vulnerabilities. Through Hackerone Finn utilized a private disclosure functionality that allowed all participants of the program to view the vulnerabilities that had been uncovered by other hackers, creating a transparency in the program. Performing this without a BBaS-platform can be hard to do for an SMB with limited resources. An alternative can be to only disclose the original vulnerability to the participant that submits a duplicate through direct communication, which will result in a section covering the part about disclosure of vulnerabilities when duplicates are submitted, being added to the program policy. Something else that might encourage the companies to be more open is something that can help pave the way forward for bug bounty programs in Norway. During the interviews I asked how bug bounty programs would become more established in the Norwegian market, where there was common agreement that this would be through sharing experiences. Companies that perform bug bounty programs need to share their experiences with having a bug bounty program, so that when the community sees that companies have success with the programs, others will follow. A great way to do this can be to publicly share the issues that the program uncovers. The vulnerabilities will have come for a good price, and when they are fixed the overall risk of the company will have decreased significantly.

5.14 Responsible Disclosure Policy

A common trait that came up through the interviews was that a company should at least have a responsible disclosure policy. Although Visma had a private bug bounty program, they still made it possible for individuals to come to them with security bugs they had found. Furthermore, the bug bounty hunters, meant that this was essential for every company that expose services to the Internet. By not having the policy, the company risks the possibility of security researchers not disclosing vulnerabilities they are aware of due to the fear of facing legal actions. Sometimes the researchers did not even attempt to find vulnerabilities on a system, but happened to discover them by chance, or they tested something similar and discovered that the weakness was also present in the computer systems of other companies.

Chapter 6

Results

In this chapter a table presenting the most important aspects that a Norwegian SMB should focus on when implementing a private bug bounty program is presented. The table is meant to be guidelines, and should be adapted to the company after the program has started.

Along with the table 6.1, the companies should also keep in mind that the hackers are a scarce resource, and that the SMBs would do wisely in treating them fair and attempt to keep them happy at all times, unless they are acting unreasonable.

Private Bug Bounty Program Essentials	
Trait	**Notes**
Policy	A detailed list of what is permitted to do, and what is not permitted.
Program Scope	Include only as many services into the scope as the company can handle.If the activity is too much, reduce the scope.
Out of Scope Items	The items that the company does not accept reports from. Expand this section if you receive reports on items you do not want in scope.
Non Qualifying Vulnerabilities	A list that will grow with the program. Add items when reports have been submitted, that you did not want
Report Setup	A detailed list of how a report should look like.
Reporting Method	The hackers should have a clear and easy procedure for submitting reports.E.g an email address: bounty@<company domain>
Rewards	A table showing what each impact category will be rewarded with.
Payment Method	A simple way to pay the hacker. Bank transfer for Norwegian hackers,PayPal for foreign.
Time to First Response	After the hacker submits a report, he will expect a reply shortly. The company should attempt to respond within a day.
Time to Triage	Should not take longer than a couple of days. The hacker will also receive the reward when this is done.
Triage	Use Bugcrowd's VRT to decide impact.
Time to Fix	A maximum of 90 days should be spent until the vulnerability is mitigated.
Disclosure Policy	Needs to be assessed for each company, but it is recommended to allow hackers to disclose vulnerabilities after the vulnerability has been fixed.
Safe Harbour	A policy that states that if the activity is done with good intentions, legal actions will not be pursued.
Recruitment	Ask peers such as Visma and Finn to invite their nordic hackers.

Table 6.1: The table lists up the essential traits a bug bounty program needs to focus on.

Chapter 7

Conclusion

During this master's thesis has explored existing bug bounty programs, and performed interviews with Norwegian companies that have implemented bug bounty programs with the help of BBaS-platforms. I have also interviewed two experienced penetration testers that have also spent a vast amount of time researching vulnerabilities for bug bounty program, where one of them even had experience with running a bug bounty program. The study uncovered that bug bounty programs are a great way to uncover vulnerabilities, and that implementing the program correctly will most likely result in paying less for each security bug than with a traditional penetration test. The project also uncovered that the option BBaS would most likely be too expensive for a small to medium sized company due to yearly fees that would cost as much as approximately two annual and thorough penetration tests, and that for a budget of this size penetration testing would be more rewarding for the organization.

However, the thesis also proposes focal points for a private bug bounty program that can be economically prosperous for small companies if executed well. The thesis also points out two major problems with this program that needs to be solved, the problems being that running a bug bounty program requires an employee with a certain level of competence within information security, and that running a private bug bounty program on your own will make it problematic to recruit penetration testers. Although, when those problems are resolved, a bug bounty program that can prevail in the Norwegian market have reflected on the following topics and propose the following items:

- Program Policy
- Program Scope
- Qualifying Vulnerabilities
- Non-qualifying Vulnerabilities
- Report Setup
- Reporting Method

- Rewards
- Payment Method
- Disclosure
- Safe Harbor

In the analysis part of the thesis, some standard values for these points sections are proposed, and can be used in setting up the program. However, the project also uncovered that there is not really a perfect setup for a bug bounty program that can be utilized for all companies. As soon as the program is up and running, the process of adjusting the program for each individual company starts. This process involves discovering what to expect from the program, what the company can handle in terms of rewards and submitted reports, how fast vulnerabilities can be fixed, how open the company want to be regarding the program, and several other aspects. The most important thing is to create a program that suits your company that can also manage to hold on to the hackers' trust and interest.

One final point is that if these options ruled out due to capacity, the least effort a company can do to improve their overall risk lever, is to implement a responsible disclosure policy. The policy will make it more comfortable for hackers to report vulnerabilities they discover in the organizations system, even if it is targeted research or a vulnerability discovered by chance.

References

Address, W.M.I., 2020. What is cidr notation? https://whatismyipaddress.com/cidr/. Online; accessed 26-February-2020.

Bugcrowd, 2018. 2018 state of bug bounty. https://view.highspot.com/viewer/5cdb1ca4628ba2681bdb20eb. Online; accessed 09-September-2019.

Bugcrowd, 2020a. Glossary of cybersecurity terms. https://www.bugcrowd.com/resources/glossary/. Online; accessed 03-November-2019.

Bugcrowd, 2020b. Lime. https://bugcrowd.com/lime. Online; accessed 20-January-2020.

Bugcrowd, 2020c. Programs. https://bugcrowd.com/programs. Online; accessed 26-November-2019.

Bugcrowd, 2020d. Vulnerability rating taxonomy. https://github.com/bugcrowd/vulnerability-rating-taxonomy. Online; accessed 10-Februar-2020.

Bugcrowd, 2020e. Vulnerability rating taxonomy table. https://bugcrowd.com/vulnerability-rating-taxonomy. Online; accessed 10-Februar-2020.

Cloudflare, 2020. What is a malicious payload? https://www.cloudflare.com/learning/security/glossary/malicious-payload/. Online; accessed 25-February-2020.

EdOverflow, Shafranovich, Y., . Security.txt. https://securitytxt.org/. Online; accessed 20-February-2020.

Facebook, 2020a. Report a security vulnerability. https://www.facebook.com/whitehat/report/. Online; accessed 11-February-2020.

Facebook, 2020b. Whitehat. https://www.facebook.com/whitehat/. Online; accessed 10-February-2020.

Google, 2019. Google cloud platform vulnerability reward program. https://static.googleusercontent.com/media/www.google.com/no//about/appsecurity/reward-program/GCP_PRIZE_2019.pdf. Online; accessed 7-December-2019.

Google, 2020a. Google hall of fame. https://www.google.com/about/appsecurity/hall-of-fame/archive/. Online; accessed 18-February-2020.

Google, 2020b. Google vulnerability reward program. https://www.google.com/about/appsecurity/reward-program/. Online; accessed 18-February-2020.

Hackerone, 2016. Visma enterprise oy. https://hackerone.com/vismaenterpriseoy. Online; accessed 17-November-2019.

Hackerone, 2019a. The 2019 hakcer report. https://www.hackerone.com/sites/default/files/2019-02/the-2019-hacker-report_3.pdf. Online; accessed 22-February-2020.

Hackerone, 2019b. Challenge and client. https://hackerone.com/sites/default/files/2018-06/Challenge&client/testimonials/V2_0.pdf. Online; accessed 5-December-2019.

Hackerone, 2020a. 90 day leaderboard. https://docs.hackerone.com/hackers/90-day-leaderboard.html. Online; accessed 25-February-2020.

Hackerone, 2020b. A fully-managed hackerone bug bounty program. https://www.hackerone.com/product/bounty. Online; accessed 24-February-2020.

Hackerone, 2020c. Hackerone programs. https://hackerone.com/bug-bounty-programs. Online; accessed 5-December-2019.

Hackerone, 2020d. Most resolved reports. https://hackerone.com/directory/programs?order_direction=DESC&order_field=resolved_report_count. Online; accessed 15-January-2020.

Hackerone, 2020e. Verizon media. https://hackerone.com/verizonmedia. Online; accessed 16-January-2020.

Inc, F., 2020. Common vulnerability scoring system v3.1: Specification document. https://www.first.org/cvss/specification-document. Online; accessed 14-December-2019.

Mingyi Zhao, A.L., Grossklags, J., 2017. Devising effective policies for bug-bounty platforms and security vulnerability discovery. http://www.aronlaszka.com/papers/zhao_devising.pdf. Online; accessed 15-November-2019.

Newman, L.H., 2018. Facebook, under scrutiny, pays out largest bug bounty yet. https://www.wired.com/story/facebook-bug-bounty-biggest-payout/. Online; accessed 10-February-2020.

NHO, 2020. Fakta om små og mellomstore bedrifter. https://www.nho.no/tema/sma-og-mellomstore-bedrifter/artikler/sma-og-mellomstore-bedrifter-smb/. Online; accessed 26-February-2020.

Norton, 2020. Zero-day vulnerability: What it is, and how it works. https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work.html. Online; accessed 25-February-2020.

Project, O.W.A.S., 2017. Owasp top ten 2017. https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_Top_10. Online; accessed 13-October-2019.

Project, O.W.A.S., 2020a. Owasp risk rating methodology. https://owasp.org/www-project-risk-assessment-framework/. Online; accessed 26-October-2019.

Project, O.W.A.S., 2020b. Owasp top ten. https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. Online; accessed 10-October-2019.

Relik, 2016. Penetration testing execution standard. http://www.pentest-standard.org/index.php/Main_Page. Online; accessed 15-September-2019.

Rouse, M., 2018. Searchsecurity. https://searchsecurity.techtarget.com/definition/sandbox. Online; accessed 22-February-2020.

of Standards, N.I., Technology, 2020. Computer security resource center. https://csrc.nist.gov/glossary/term/vulnerability. Online; accessed 10-October-2019.

Unviversity, C., 2020. Cambridge dictionary. https://dictionary.cambridge.org/dictionary/english/bounty. Online; accessed 27-October-2019.