



Assignment of master's thesis

Title:	Design of Commercial Bug Bounty Program
Student:	Jan Sedláček
Supervisor:	Ing. David Knap
Study program:	Informatics
Branch / specialization:	Computer Security
Department:	Department of Information Security
Validity:	until the end of summer semester 2021/2022

Instructions

The goal of the thesis is to explore available solutions for Bug Bounty programs, and propose high-level design for a new Bug Bounty platform aimed for commercial B2B usage.

1. Discuss history and usage of bug bounty program as a cybersecurity measure and compare various approaches available. Compare bug bounty to other similar measures like penetration testing.
2. Explore available backend systems and platforms for bug bounty. For at least three different, experiment with them (if possible) and/or discuss the experience of their current users.
3. For potential users of your solution, discuss their needs and requirements, as well as limitations they face with solutions already available on the market.
4. Based on the knowledge gained, propose a high-level design of a bug bounty platform used for B2B and closed campaigns.



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Master's thesis

Design of Commercial Bug Bounty Program

Bc. Jan Sedláček

Department of Computer Security
Supervisor: Ing. David Knap

April 26, 2021

Acknowledgements

I would like to thank my supervisor – Ing. David Knap for his invaluable supervision and support during the writeup of my thesis. Alongside, I would like to express gratitude to my colleagues in CBS, namely Ing. Zdeňek Grmela and Ing. Petr Holeček for their treasured support and consultations which was really helpful to have an inside view of the problem. I would like to thank my friends, colleagues and family for their encouragement and support all through my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on April 26, 2021

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2021 Jan Sedláček. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Sedláček, Jan. *Design of Commercial Bug Bounty Program*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Tato práce se zaměřuje na bug bounty, objasňuje význam sekcí souvisejících s bug bounty programy a obecným procesem hledání bug bounty. Příspěvek poskytuje srovnání s jinými bezpečnostními postupy a zdůrazňuje podobnosti, rozdíly a výhody týkající se bug bounty. Na teoretickou studii navazuje praktická analýza čtyř dostupných platforem poskytujících lov bugů jako službu. Vedle osobních zkušeností je u všech čtyř poskytovatelů zahrnuta recenze uživatelů a společností. Práce končí high-level návrhem pro platformu bug bounty, která zohledňuje shromážděné informace během předchozích fází.

Klíčová slova Bug Bounty, platforma Bug Bounty, lov Bug Bounty, historie Bug Bounty, HackerOne, Bugcrowd, Intigriti, YesWeHack, Bug Bounty High-level Design

Abstract

This thesis focus on bug bounty, it clarifies the meaning of sections related to the bug bounty programs and the general process of bug bounty hunting. The paper provides a comparison with other security practices and highlights the similarities, differences and advantages concerning the bug bounty. The theoretical study is followed by a practical analysis of four available platforms

providing bug bounty hunting as a service. Alongside the personal experience, the review of the users and companies is included for all four companies. The thesis ends with a high-level design proposal for a bug bounty platform, evaluating gathered information during previous phases.

Keywords Bug Bounty, Bug bounty platform, Bug Bounty hunting, History of Bug bounty, HackerOne, Bugcrowd, Intigriti, YesWeHack, Bug Bounty High-level design

Contents

Introduction	1
Motivation and objectives	1
Problem statements	1
1 Bug Bounty	3
1.1 Bug	4
1.2 History	5
1.3 Content of bug bounty programs	7
1.3.1 Overview	8
1.3.2 Scope	9
1.3.3 Rules	9
1.3.4 Reports	10
1.3.5 Rewards	10
1.4 Closed and public programs	12
1.5 Time-bounded and ongoing programs	13
1.6 Self-hosted bug bounty	13
1.7 Bug bounty as a service	14
1.8 Benefits of bug bounty program	15
1.9 Summary	16
2 Alternatives to Bug Bounty	17
2.1 Vulnerability disclosure program	17
2.1.1 Bug bounty comparison	18
2.2 Vulnerability scanning, management, assessment	19
2.2.1 Bug bounty comparison	21
2.3 Penetration testing	21
2.3.1 Bug bounty comparison	24
2.4 Red and Purple teaming	25
2.4.1 Bug bounty comparison	27

2.5	Security auditing	28
2.5.1	Bug bounty comparison	30
2.6	Summary	30
3	Analysis of available products	31
3.1	HackerOne	31
3.1.1	Personal experience	32
3.1.2	User's experience	33
3.1.3	Company's experience	33
3.2	Bugcrowd	34
3.2.1	Personal experience	35
3.2.2	User's experience	36
3.2.3	Company's experience	37
3.3	Intigrity	38
3.3.1	Personal experience	38
3.3.2	User's experience	40
3.3.3	Company's experience	40
3.4	YesWeHack	41
3.4.1	Personal experience	41
3.4.2	User's experience	42
3.4.3	Company's experience	43
3.5	Unsatisfied needs	43
3.6	Summary	43
4	Product Design	45
4.1	Requirements	45
4.2	General Description	45
4.2.1	Assumptions	46
4.3	Interactions	47
4.4	Elements	49
4.4.1	Basic user	49
4.4.2	Company	50
4.4.3	Triage team	51
4.4.4	Programs	51
4.4.5	Report	52
4.5	Workflow	53
4.6	Future steps	55
	Conclusion	57
	Bibliography	59
	A Acronyms	65
	B Contents of CD	67

List of Figures

1.1	Poster created for bug bounty program. 1	6
1.2	Overview of the Mattermost bug bounty program on HackerOne 2	8
1.3	Overview of rewards by Playstation on HackerOne 3	11
1.4	Examples of self-hosted bug bounty programs and platforms 4	14
2.1	Cycle of vulnerability management process 5	20
2.2	Cycle of penetration testing 6	23
2.3	Comparison of red, blue and purple team 7	27
2.4	Common steps in security auditing 8	29
3.1	HackerOne logo 9	32
3.2	BugCrowd logo 10	35
3.3	Intigriti logo 11	39
3.4	YesWeHack logo 12	41
4.1	Overview of actions possible by actors	46
4.2	Relationships and Entities with attributes	48
4.3	General workflow for Public Managed programs	53
4.4	General workflow for Private Managed programs	54
4.5	General workflow for Closed Managed programs	54

Introduction

Bug bounty hunting is an IT security section, that exists for several years already, but it is not yet exactly defined and used just by several companies. Bug bounty hunting is the process of searching for product issues related to security. The hunting is done by hackers(The term “hacker” in this thesis is always used as the meaning of an ethical computer hacker, also referred to as “white-hat hacker.” It is not referring to a person supporting or doing any illegal activities.), or a security researcher.

Motivation and objectives

The bug bounty is built on the principle of paid per finding rather than time or supervised project. A usage of bug bounty allows the company to establish a new process to constantly look for bugs to keep the product and the company itself safe. It establishes a process of how to approach and resolved security bugs. To outsource the bug finding, the company can reach a bug bounty platform, which delivers it to many hackers and specialists. This introduces a new business possibility, create a bug bounty platform to group people hunting bugs and charge companies for delivering their product to the group of specialists.

Bug bounty hunting needs to be compared to other used practices for IT security to see the differences and benefits introduced by a bug bounty platform. The rise of the bug bounty as a service introduces a new platform to be developed and analysis is needed, to include all necessary aspects, to create a competitive product.

Problem statements

A bug presented in the product introduces a problem, especially from the security perspective to the company and the product itself. To find a bug

during project development is hard due to the complexity of the product, used tools, principles, etc. To deliver the product to many hackers and security researchers can be complicated, especially for smaller companies.

The bug bounty needs to highlight its advantages and needs concerning other security approaches. The existing platforms need to be analysed to understand the current market and its need by receiving feedback from companies, hackers and security researchers.

As bug bounty is a fairly new term and approach in IT security, the first chapter will define the bug bounty, briefly mention the short history, and observed similarities of bug bounty programs. This will be followed by the next section focused on other more common IT security approaches to find security-related bugs within a product. These will be as well compared to the bug bounty to state the advantages presented in the usage of the bug bounty. The last part will provide an analysis of existing bug bounty platforms together with feedback from companies and users. This will be used to provide a high-level design of a platform for a bug bounty.

Bug Bounty

At a high-level, the bug bounty program, referred to as a vulnerability reward program as well, is just published document specifying the scope and rules available for testing. The principle of the bug bounty is based on the Vulnerability market (by [13] it is defined as a place to trade discovered vulnerabilities for a reward). The public crowd of security researchers, penetration testers and hackers are free to test the target, report the findings and potentially earn a reward. The report is supervised either by a triage team or by the company itself. The possible exploit and impact of the finding are verified and forwarded for remediation or mitigation to the responsible team. The reward received depends on the company publishing the bug bounty program. It can be in the form of reputation, company's products, so-called non-monetary or financial ranging from few dollars to several hundreds of thousands of dollars, so-called monetary.

Bug bounty, as stated in [14], has increased in popularity in recent years. Increased popularity and related interest of companies introduced a new concept of platforms associating bug bounty programs of companies in one place and partially taking care of the triage for submitted reports. The popularity growth is so significant, that even organisations as the Department of Defense (DoD) of the United States have their bug bounty programs, moreover, some are using the services of the 3rd party platforms¹. The company can create a bug bounty program for many areas such as software, firmware, hardware, multiple different platforms, infrastructure or even data policies.

The interest of companies in bug bounty is supported by continuous monitoring by a security specialist, who is paid only for findings. Some bug bounty hunters had in the past problem with the law as black-hat (in [15] is defined as a malicious actor doing illegal operation and causing damage to a company) or grey-hat ([15] consists of actors doing illegal activities, but try to report the findings to the company in exchange for monetary reward) hackers, who are

¹<https://hackerone.com/deptofdefense> - accessed on 2021-02-23

now helping to improve the security. The bug bounty is for them a possibility to use their skills and know-how for legal earning.

In recent years international companies realised the need for enhanced security and started inviting the external testers more publicly and being more opened for externally acquired vulnerability information and related exploits. The possibility of income from the bug bounty program is a new motivation for previously malicious actors, who earlier shared this information on hackers forums with low profit and with a potential threat to companies.

An important remark formulated in [16] “There exists no established terminology to describe the current crowd-sourcing patterns for vulnerability discovery and disclosure. Bug bounties, vulnerability reward programs, security challenges, vulnerability hunting campaigns, and related terms are used more or less interchangeably to describe the same phenomenon.”

1.1 Bug

The objective of hunting bugs in bug bounty programs is usually related to software. A so-called software bug is related to a code and can compromise the security of the system, bugs non-related to the security exist as well, but are not a huge threat from a security perspective and are not being awarded or the pay bill is unworthy, therefore, they are often not included in bug bounty programs. The software, policies and product design contain security vulnerabilities, that have not been intended and are the result of invalid logical flow, incomplete or wrongly interpreted design, mathematical error or invalid assumption. Not all bugs are considered as security issues or vulnerabilities, this depends on the vendor and the intended usages of the product. A bug may exist in a product for a long period before it is discovered and may not be observable under all circumstances. Bugs are not generated only during initial software development but any update, including bug fixing, may introduce even multiple new bugs or reopen patched bugs.

As a bug, stated in [17], introduces a vulnerability of a system, this vulnerability may be exploited by a malicious actor, who will gain the unintended abilities that may be abused. The bugs are not fully resolved during the development phase, as the cost would increase dramatically, as well as there are many target platforms or scenarios that are hard to be tested within the accepted scope.

Companies selling their products are aware of bug possibility and to protect their customers, reputation and earnings multiple methods for bug fixing have been developed. Companies employ security experts, researchers, engineers, software and policies to minimise the number of bugs available in their product. With the increasing complexity of these products, with multiple smaller companies, external researchers, white-hat (*in [15] defined as an actor with a legal approval to compromise a system within agreed boundaries*) and

grey-hat hackers realized the need for identifying the security-critical bugs and the related economical value.

1.2 History

The origins of Bug bounty hunting are in the Old West, where financial rewards were published for criminals death or life. Even in the 19th century, these hunters were part of the U.S. government and law system. These hunters had the right to imprison a person or force him to step forward to the sheriff or any other law institute. The modern bug bounty hunters of the IT world have their origin at the end of the 20th century.

The very first public campaign, documented in [18], was launched in 1983 called “Get a bug if you find a bug“ published by company VRTX inviting specialists to search for a bug in microprocessors sealed in silicon. The reward for a valid bug was a Volkswagen Beetle or \$1,000 cash. The next bug bounty program, as archived in [19], came twelve years later in 1995 from Netscape’s software engineers with the release of Netscape navigator 2.0. This bug bounty as well as the VRTX from 1983 contained exact scope. For the finding of a security bug in the navigator, there was a cash reward. The idea was to encourage an open review by the public and create a product of higher quality. The program lasted till the final release of the product and multiple findings were fixed due to this program.

Unfortunately, this methodology was not immediately followed by other vendors and there were no publicly documented programs till the beginning of the 21st century. In 2002 came a huge milestone for the Bug bounty as the idea is common nowadays. The organisation iDEFENSE, as described [20], announced its Vulnerability Contribution Program, the program was the first type of Bug Bounty as a service. The company collected vulnerabilities in software not related to their products. They verified the vulnerabilities, their exploits and impacts, rewarded the finder with a cash value and acted as a middleman of the researcher and affected company. This idea was followed by the Zero Day Initiative² in 2005 who took over and continued with the same idea.

Finally, in 2004, as summarized in [21], the first well-known company started with its bug bounty program. Mozilla announced a security bug bounty program for critical vulnerabilities in the end-user software developed by the company. This program is still active nowadays³. Hackers and security researchers can be still awarded for bug reporting. In 2010 was a huge kick-off to the public, as described in [22], as Google similarly started their bug bounty programs as Mozilla did and many corporations implemented the bug bounty programs as well in the same year. Mozilla expanded the program

²<https://www.zerodayinitiative.com/> - accessed on 2021-03-15

³<https://www.mozilla.org/en-US/security/bug-bounty/> - accessed on 2021-02-27

1. BUG BOUNTY



Get a bug if you find a bug.

Show us a bug in our VRTX® real-time operating system and we'll return the favor. With a bug of your own to show off in your driveway.

There's a catch, though.

Since VRTX is the only microprocessor operating system completely sealed in silicon, finding a bug won't be easy.

Because along with task management and communication, memory management, and character I/O, VRTX contains over 100,000 man-hours of design and testing.

And since it's delivered in 4K bytes of ROM, VRTX will perform for

you the way it's performing in hundreds of real-time applications from avionics to video games.

Bug free.

So, to save up to 12 months of development time, and maybe save a loveable little car from the junkyard, contact us. Call (415) 326-2950, or write Hunter & Ready, Inc., 445 Sherman Avenue, Palo Alto, California 94306.

Describe your application and the microprocessors you're using—Z8000, Z80, 68000, or 8086 family. We'll send you a VRTX evaluation package, including timings for system

calls and interrupts. And when you order a VRTX system for your application, we'll include instructions for reporting errors.*


But don't feel bad if in a year from now there isn't a bug in your driveway.

There isn't one in your operating system either.

**HUNTER
♦ READY** 

VRTX
Operating Systems in Silicon.

*Call or write for details. But, considering our taste in cars, you might want to accept our offer of \$1,000 cash instead. © 1983 Hunter & Ready, Inc.

Figure 1.1: Poster created for bug bounty program. 

and Federal agencies and corporations started the same trend. Shortly other technological giants followed, as Facebook in 2012 to enforce the security of provided services.

The final important steps happened in 2012 when Casey Ellis founded the Bugcrowd^[4] and Jobert Abma with Michiel Prins founded HackerOne^[5]. These organisations are nowadays one of the biggest public platforms offering bug bounty as a service. These platforms connect world-class hackers with companies and support the education of new specialists.

Nowadays, as describes ^[23], there are unions of organisations, those even created bug bounty programs to search for bugs in non-profiting products and open sources, such as Ruby on Rails^[6] and other common frameworks.

1.3 Content of bug bounty programs

As the bug bounty lacks an exact definition of boundaries, the same holds for the published bug bounty programs structure. It can be freely customised, and even the 3rd party services do not use standardised templates. However, even as there is no exact definition, there are similarities, that can be seen as required, these include scope and rules. Findings presented across this section have been observed based on multiple HackerOne programs (Mattermost^[7], PlayStation^[8], Xiaomi^[9]), Bugcrowd programs (1Password^[10], Netflix^[11]) and self-hosted programs of Facebook^[12] and Microsoft^[13]. The naming of these sections may vary, and companies can include additional sections such as reward scaling and description to enhance the provided information and attract more testers.

Some 3rd party software providers standardised these rules as a standard for all programs, as long as the customer does not specify differently. The Bugcrowd in this document^[14], for example, lists a set of vulnerabilities that by default will not be considered as the worth of bounty, Reward explanation, or confidential obligations to be followed by any user of Bugcrowd.

⁴<https://www.crunchbase.com/organization/bugcrowd> - accessed on 2021-02-22

⁵<https://www.hackerone.com/company/leadership> - accessed on 2021-02-15

⁶<https://hackerone.com/rails?type=team> - accessed on 2021-03-05

⁷<https://hackerone.com/mattermost?type=team> - accessed on 2021-03-05

⁸<https://hackerone.com/playstation?type=team> - accessed on 2021-03-05

⁹<https://hackerone.com/xiaomi?type=team> - accessed on 2021-03-05

¹⁰<https://bugcrowd.com/agilebits> - accessed on 2021-03-05

¹¹<https://bugcrowd.com/netflix> - accessed on 2021-03-06

¹²<https://www.facebook.com/whitehat> - accessed on 2021-03-06

¹³<https://www.microsoft.com/en-us/msrc/bounty> - accessed on 2021-03-03

¹⁴<https://www.bugcrowd.com/resource/standard-disclosure-terms/> - accessed on 2021-02-12

1. BUG BOUNTY

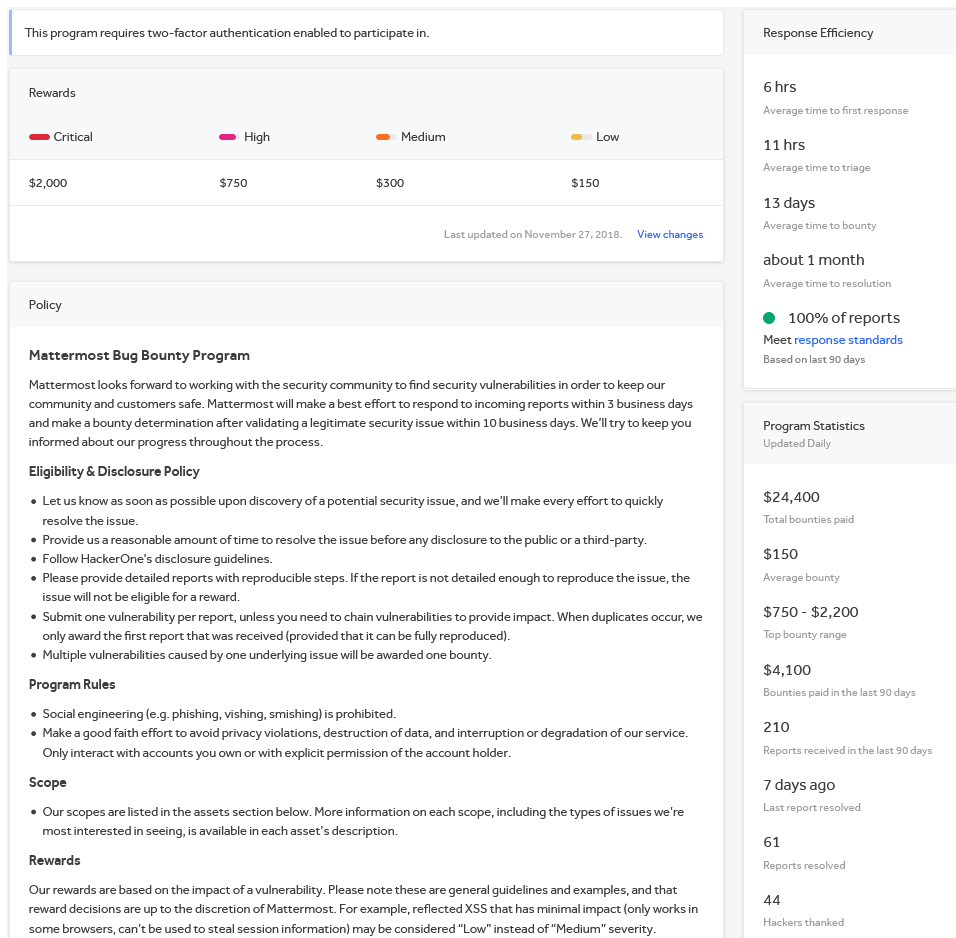


Figure 1.2: Overview of the Mattermost bug bounty program on HackerOne [2]

1.3.1 Overview

This section has various names, such as Purpose or Details, and is more common to the programs hosted on 3rd party services. It is not that common for self-hosted solution. The content varies as well, but it serves as a brief description of the company, motivation of the bug bounty and links to additional resources. It is common, if the company runs an additional vulnerability disclosure policy ([24] and [25] explain the vulnerability disclosure policy as a process of handling newly reported vulnerabilities by external actors. In more details described in next chapter), to state that discovered vulnerability that is out of scope can be reported using the VDP.

1.3.2 Scope

Every single bug bounty program should have a section called scope, alternatively, the section can be named as a target or any similar name not changing the meaning dramatically.

The scope of the program contains details of what can be scanned with the related program. The listed targets can be scanned by the security researcher or hacker, but in case of additional sections, such as out of scope, rules, etc. these must be satisfied as well. The scope does not only contain what can be scanned but as well what kind of vulnerabilities are accepted and should be reported.

Most commonly the scope contains a set of URLs, even with regular expressions for sub-domains, for the scanning of the web applications. Besides, it becomes increasingly more common to include testing of a software products developed by the company. Some companies such as PlayStation, for example, created a program¹⁵ to test the hardware they created, “We are currently interested in reports on the PlayStation 4 and PlayStation 5 systems, operating systems, accessories and the PlayStation Network.”

The out of scope section is used to briefly list the types of vulnerabilities and targets that are prohibited to be tested for a given bug bounty program. This section is partially overlapped with the following section specifying the rules for the testing.

1.3.3 Rules

This section is quite rare, but if it is present, it contains important information. Some companies have additional requirements, that for some reason have been moved outside of the scope section. A common requirement in this section is a set of rules on what not to perform as testing, who can conduct the testing and how.

Custom Examples from analysed programs with generalised naming:

- Submit one vulnerability per report unless you need to chain vulnerabilities to provide impact.
- Do not commit privacy violations, destruction of data, or interruption or degradation of our service.
- Create test accounts or test content to avoid affecting real users
- Do not test vulnerabilities on user accounts that you do not own or have rights to access or control.
- Provide details of the vulnerability finding, including information needed to reproduce and validate the report

¹⁵<https://hackerone.com/playstation?type=team> - accessed on 2021-02-17

1. BUG BOUNTY

- Please make sure to use the User-Agent string `companyName_BountyProgram_UserName` while testing
- Tools without the User-agent string `companyName_BountyProgram_UserName` may result in ban
- Social Engineering (e.g. phishing, vishing, smishing) is prohibited.
- Automated requests/scanning must be kept to under 45 requests per minute.

The rules are crafted by the company to control the testing procedures and distinguish the testing procedure from the potentially malicious attack.

1.3.4 Reports

The section specifying how to report the findings to the bug bounty program. It is common for self-hosted bug bounty programs. The section usually lists the general best practices on how to write a qualitative report. Highlights to include technical details of the vulnerability, description of its impact and involved product with the version number. Besides, it is reasonable to provide steps on how to reproduce the vulnerability. This section briefly specifies as well where should the report be submitted and what kind of security measures should take place, for example, email encryption, password protection attachments and supporting material to accomplish this.

The hosted bug bounty programs usually have any kind of fill-in form for the users to submit the report^[16] with the correct form and to provide a common template for all the findings.

1.3.5 Rewards

The concept of the reward is important as a motivation for users to participate in the program. The rewards usually have two different forms. A reward for the vulnerability finding can be either monetary or non-monetary.

The monetary rewards are more popular among testers. They are being publicly discussed using social media, where researchers boast themselves with high payouts^{[17][18]}. This encourages new researchers to join the bug bounty community. The amount paid by the company has a general rule, for higher severity the higher payout, but no threshold is guaranteed to be paid for any severity range. Therefore, this introduced a big difference in the market. The PlayStation will pay up to \$50,000 for critical severity in PlayStation 5 or 4, on

¹⁶<https://docs.bugcrowd.com/researchers/reporting-managing-submissions/reporting-a-bug/> - accessed on 2021-02-18

¹⁷<https://twitter.com/alaa0x2/status/1372598185810669570> - accessed 2021-03-19

¹⁸<https://www.polygon.com/2021/3/16/22334214/gta-online-loading-times-t0st-update-bug-bounty> - accessed 2021-03-18

1.3. Content of bug bounty programs

Rewards			
Critical	High	Medium	Low
PlayStation 5			
\$50,000	\$10,000	\$2,500	\$500
PlayStation 4			
\$50,000	\$10,000	\$2,500	\$500
PlayStation Network			
\$3,000	\$1,000	\$400	\$100

PlayStation will determine, in its sole discretion, whether a bounty will be awarded. Reward amounts will differ based on vulnerability severity, as well as the quality of the report. Sony will only award a bounty to the first researcher to have reported a previously unreported, vulnerability.

The amounts listed above represent the minimum bounty for each severity category.

Last updated on October 24, 2020. [View changes](#)

Figure 1.3: Overview of rewards by Playstation on HackerOne [3]

the other hand, smaller and less popular company Mattermost will award the researcher for critical vulnerability up to \$2,000 as both companies defined these ranges in their programs using HackerOne. This section as well may contain how the money is transferred or what are the additional conditions for the rewards to be paid out. Usually, the reward is provided after the bug is considered closed. The payout methods are specific to the provider of the bug bounty and consist of some kind of bank transfer, or via different services such as PayPal, Currencycloud, cryptocurrency etc.^[19]

The non-monetary rewards are slightly less popular but still appreciated. This reward has two general concepts, in the form of fame and product, voucher, or any other swag reward. From these two categories of non-monetary rewards the products, vouchers, or any other form of swag reward are not that appreciated concerning fame. The fame is usually in the form of some ranking points for bug bounty platforms and a hall of fame for the companies running self-hosted bounties. At first glance, this looks like not a valuable achievement, but it can be well-used in CV especially in the security area. Fame is a kind of proof of the user's knowledge, know-how and proficiency that can be

¹⁹<https://docs.bugcrowd.com/researchers/payments/setting-up-payment-methods/> - accessed 2021-03-04

handy for further career^{[20][21]}.

1.4 Closed and public programs

Bug bounty programs, as state [26] and [27], can be divided into two major groups by availability. The program can be either publicly accessible, and anyone can contribute, or the program has been established as private and the owner can define researchers and hackers who have access. Historically the programs were public, but with the rise of bug bounty platforms concentrating the specialists on one place and with some additional rating to these users, private programs came in place.

The public program is simply published on the Internet and anyone can access it, regardless of the company having a self-hosted solution or used the 3rd party as Software-as-a-service. As anyone can access the program, read the scope and perform testing research, the public program has more submissions of findings and is inspected from multiple perspectives. More reports generated mean, there is more work and human resources required to verify the findings or mark the report as duplicate. This extra work, no matter which deployment option is used, increases the cost of the bug bounty program with possibly higher security. Besides, the company's name is associated with an interest in IT security, which improves the image of the company and could introduce new opportunities.

Private or closed programs, as [28] explains are based on the invitation of researchers and hackers into this program. This concept has become increasingly popular with the rating of hackers and researchers based on various metrics such as the number of findings, testing tools contribution, experiences and successes in CTF and other practices, that the organisation concentrating hackers and researchers implemented. The advantage is that the invitation may filter out the inexperienced users who may incorrectly report findings and generally, it decreases the traffic and stressing of the target.

From the perspective of researchers and hackers, the closed campaign gives a possibility to check for a target with a lower concentration of users, and therefore a better chance to have findings concerning a report to the public program where thousands of users act and test on a daily basis. The invitation to a private program may be seen as a privilege and reward^[22] for hard-working. Generally speaking, the private programs introduced an interesting aspect, but even the best hacker has no guarantee to match the constraints of the private

²⁰<https://www.indeed.com/viewjob?jk=f00bb3f7dc70dcba&tk=1f1ht10bh3kkj002&from=serp&vjs=3> - accessed 2021-03-15

²¹<https://www.indeed.com/viewjob?jk=4af45f7f7d054a0f&tk=1f1ht10bh3kkj002&from=serp&vjs=3> - accessed 2021-02-16

²²<https://docs.hackerone.com/programs/private-vs-public-programs.html> - access on 2021-03-15

program with relation to obtained ratings. Most companies still use public programs as they prefer more researchers.

1.5 Time-bounded and ongoing programs

The program can be as well time limited as unlimited, depending on the aim of the company. Short time programs²³ are usually with the special aim or as a trial of a bug bounty for a new company.

Programs without time limitation are published and remain active for continuous submissions of reports. These programs are usually associated with a company that evolves the scope of the program to include or exclude objectives as the business progress. As an example, Google²⁴ initially started with a scope containing `www.google.com` as the company kept expanding additional web applications have been included in this scope as well. The time-unlimited programs are more often used, but this does not mean, that the company can not withdraw from the program. Simply saying during the publishing time, there is no known termination of this program.

Time-bounded bug bounty programs are usually of two different types. The private programs are usually listed for a limited period and after this period they either become public or terminate. The second type of time-limited programs is related to the releases. In case the company has a new product, it can create a time-limited campaign to test its security and functionality before it is delivered to the final customer. These programs have more detailed scopes to test the critical features of the new product. The same approach is used with software and hardware versions and their support.

1.6 Self-hosted bug bounty

In case the company decides to take care of the bug bounty program itself, it must define an exact procedure on how the reports of discovered vulnerabilities should be submitted to the company and how to attract and deliver them to the security public. This approach is not much popular, but some companies^{25,26}, especially the bigger ones do it. As a simplified version of a self-hosted bug bounty service may be seen a vulnerability disclosure policy or program (VDP) described later as an alternative to a bug bounty.

As Jason Pubal described in [29] and H. Fryer with E. Simperl in [30], the security researchers or hackers need to have a clear guideline on how to submit

²³<https://www.hackerone.com/product/challenge> - accessed on 2021-02-22

²⁴<https://www.google.com/about/appsecurity/reward-program/> - accessed on 2021-02-

16

²⁵<https://www.avast.com/bug-bounty> - accessed on 2021-02-24

²⁶<https://lisk.io/bug-bounty-program> - accessed on 2021-02-24



Figure 1.4: Examples of self-hosted bug bounty programs and platforms [4]

the finding report which must be exactly defined and provide approval. Moreover, it is an invitation the researchers and hackers to freely check within the included scope for vulnerabilities. The submitted reports should be delivered to a team, that will verify the existence of the exploit, check the described impact and decide on how to proceed further. As for any other vulnerability, it must be decided if the vulnerability should be mitigated, remediated or not resolved with relation to the possible impact and costs related to the remediation or mitigation process. To complete the process as a bug bounty program, the response should be provided to the researcher with information, if the vulnerability has been already reported, if it is seen as impactful to be rewarded or to clarify some misunderstanding.

The company must take care by itself to proceed with payments and deployment of the information to the public crowd regarding the scope of the testing, out-of-scope targets and additional information to provide the researchers starting point. A common practice is as well to define a financial boundary with relation to the severity of the vulnerability.

1.7 Bug bounty as a service

As summarised in [31], originating from the idea of the Zero Day Initiative the company does not need to create its bug bounty program, but it can be outsourced. The increase of bug bounty popularity introduced a new company that provide bug bounty platforms used by 3rd party companies, to deliver their bug bounty programs to security researchers and hackers or completely outsource the process to a provider of Software-as-a-Service (SaaS).

These companies introduced an enhancement to the bug bounty programs by offering a triage team, closed campaigns, a huge community of hackers and researchers and ranking of users by success. Besides, there are no requirements for deployment resources used for hosting the bug bounty program, and it decreases the initial investment into technologies and human resources.

The offered triage team can either fully manage the triage or just partial manage. Triage team duties described in [32] clarifies, that the triage team is responsible for the submitted reports, it will do the validation and verification of discovered findings. In case of a duplicate or reports not matching the customer's requirements, the case is closed by the triage team. The team as well takes care of the communication with the submitter. For more serious findings the triage team may collect additional data for the vulnerability, propose a plan on how to fix the vulnerability or rate its severity and submit it to the company for confirmation and further steps.

The provider has a community of hackers, that may be rated based on findings, contribution, etc. and an organisation may profit from such information by establishing a closed campaign to control the number, quality and specialisation of hackers and security researchers testing the program. The software provider ensures, that details of this bug bounty program are not publicly accessible and are delivered only to the users matching the constraints.

1.8 Benefits of bug bounty program

As stated in [33], [34] and [29], the bug bounty is a highly competitive environment, which keeps pushing the abilities and knowledge of the users. This produces more skilled individuals or groups of teams that provide professional testing, introduce a new point of view and another point of control of the entire life cycle of the product. Massive progress in digitalization took place, many companies implemented either formally or informally software development life cycle (SDL). The bug bounty included in the SDL increases the overall security of the delivered product. It may point out a specific area of the product that is vulnerable and may need to be redesigned, it helps the developers to sharpen their skill and consider as well the security view. The bug bounty supports a security-aware culture across the developers as they work with reports submitted by specialists and the business department develops an understanding of the risk and the importance of feedback from the public crowd. The bug bounty if used in correct circumstances is not expensive and provides control of the product that is comparable with penetration testing or red teaming.

1.9 Summary

A Bug bounty is a new security approach, that provides enhanced security by supervision of experts in related areas. The possibility to outsource the bug bounty programs to 3rd party software providers makes it available even to the smaller companies, that could not afford to create the self-hosted solution. The method of pay per finding is beneficial if the company took security measures before the publishing of the program. The investment is low and provides real experts control. The most important thing remains a precise definition of the scope, out of scope objectives, and well-formulated requirements and rules of the program to avoid legislation problems and attract attention. The possibility of time-limited or private campaigns allows the company to select a group of specials conducting the testing and include bug bounty as a part of the deployment process of a new product.

Alternatives to Bug Bounty

A bug bounty is a security tool improving security by continuous monitoring and testing by the public crowd. There are multiple different tools, that search for vulnerabilities and misconfigurations in a given scope, some of them are listed below with a comparison to the bug bounty programs based on collected information in the previous chapter.

2.1 Vulnerability disclosure program

Megan Brown said in [35], “Companies that lack a clear vulnerability disclosure program are at increased risk should a security researcher find a vulnerability.”

As described in [36] and [37] Vulnerability disclosure programs, or for short VDP, have been evolving alongside the bug bounty. The VDP is there to be used by security researchers or any actor, who discovered a finding regarding a company’s product. It offers a straight forward, secure and simple way to report the finding with all the evidence to the organisation. It carries additional information regarding triage, workflow for remediation or reward for this finding. This program as well contains information regarding the accepted scope²⁷ for the findings which are extremely important to define a legal scope of testing for vulnerabilities or security findings. The VDP as well allows any company to have so-called responsible disclosure, a period when the vulnerability is not known to the public. This gives additional time to the company to fix this vulnerability and protect its assets or customers. In the case of missing VDP, the vulnerability may be publicly disclosed or exploited by a malicious actor and cause damage to the company.

[38] claims, that recently the need for VPD has raised in popularity and government services in the US are already implementing and using the VDP

²⁷Range of URLs, products, software versions, IP addresses, etc.

2. ALTERNATIVES TO BUG BOUNTY

to strengthen their securities. Multiple big international companies^[28]^[29] are running VDP alongside Bug Bounty programs. It has been recommended even to the public sector to use a VDP as support of standards ISO 29147^[30] and ISO 30111^[31]. With such a trend, there are as well as companies offering the VDP as a Service to simplify the process for other companies.

The complete process of VDP, summarised in [36] and partially in [38], consists of four steps. Collection of the report with a description of the vulnerability, exploitation impact and additional information provided by the researcher. Security together with the development department performs an analysis of the report, verification of the vulnerability and propose suggestion to mitigate or remediation the vulnerability. For approved mitigation or remediation, corresponding steps are performed to remove the vulnerability. With the vulnerability being resolved, it will be disclosed and published, to other companies^[32] that may scan its assets for this new vulnerability.

2.1.1 Bug bounty comparison

In general, the vulnerability disclosure program and bug bounty provide enhanced security to the organisation. In both cases, multiple actors are checking the security of the product to protect the company's assets for no, or limited initial investment and provide long-term protection.

From the company's perspective, the VDP and Bug bounty have the same procedural requirements. Announce what can be checked, how the findings should be reported, verification and fix reported findings. Besides, both VDP and Bug bounty shows publicly the security interests of the company and invite security researchers to participate.

One of the main differences is that the vulnerability disclosure program is a methodological workflow, how vulnerabilities or security findings should be reported and delivered to the organisations to be resolved, which is a sub-part of the bug bounty. The VDP are not always associated with a reward of any kind, or the reward may not be financial. The bug bounty, on the other hand, is usually provided with a financial reward, and it is more public to invite the crowd to participate in the testing.

²⁸https://www.tech.gov.sg/report_vulnerability - accessed on 2021-03-01

²⁹<https://www.avast.com/coordinated-vulnerability-disclosure> - accessed on 2021-03-01

³⁰<https://www.iso.org/standard/72311.html> - accessed on 2021-02-18

³¹<https://www.iso.org/standard/69725.html> - accessed on 2021-03-01

³²<https://www.bugcrowd.com/resource/standard-disclosure-terms/> - accessed on 2021-02-12

2.2 Vulnerability scanning, management, assessment

The scanning of a system, described in [39] and in [40] is an automated process of scanning for vulnerabilities. The scanner performing the testing looks for weaknesses or misconfigurations in networks, web applications or the whole machine. The scope of such testing is defined at the level of the scanner. During the scanning, the scanner is connected to multiple databases referencing known security flaws, misconfigurations and vulnerabilities. After the scan is complete most of the vulnerability scanning tools generate a report summarising the security level of the scanned target, list of vulnerabilities and their references in some public databases and their severity, using for example CVE(Common Vulnerabilities and Exposures system of reference-method for publicly known vulnerabilities and exposures)^[33], NVD(U.S. Government repository of standards-based Vulnerabilities)^[34], CVSS(Free and open industry standard for assessing the severity of vulnerabilities.)^[35]

The scan can be performed in multiple different configurations to significantly impact the final result of the total number of vulnerabilities and their validity. The scan can be either intrusive or non-intrusive, this defines how aggressively will the scan try to match the vulnerability and exploit it afterwards. For more intrusive scans the results are more likely to be without false-positives and a better impact description is provided, but some damage can be done, or instability may appear during the testing.

Besides this, as explained in [39], the scan can be executed from different locations. The external scans are testing only the part of infrastructure exposed to the internet. External scanners provide vulnerabilities that a potential attacker may use to penetrate the organisational infrastructure. Internal scans, on the other hand, provide an overview of vulnerabilities in case an attacker would gain access to the system. Generally speaking, the internal scan can provide a much more detailed overview.

In addition to these, vulnerability scans can be either authenticated or non-authenticated. An authenticated scan requires additional configuration for logging in to the system during the scanning and provide much more details in the report, as it can interact with the target in a closer fashion.

The vulnerability scanning is just the first step in a more complex proactive approach described in [40], to manage security by reducing the potential of the system compromise. The entire process of vulnerability management should establish control and process flow to identify and work with vulnerabilities. Vulnerability management consists of several steps and should be

³³<https://cve.mitre.org/> - accessed on 2021-02-15

³⁴<https://nvd.nist.gov/> - accessed on 2021-02-15

³⁵<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator> - accessed on 2021-02-15

2. ALTERNATIVES TO BUG BOUNTY



Figure 2.1: Cycle of vulnerability management process [\[5\]](#)

performed periodically. It has been standardised in ISO 27000³⁶, ISO 27001³⁷ and ISO 27002³⁸, and it is a common business requirement to be performed.

The next step of vulnerability management is to identify and verify the vulnerabilities. The scanner may produce some false-positive findings, and it requires additional control. This may result in the exploitation and verification of reported vulnerabilities by an expert. The verified vulnerabilities must be further analysed to determine their actual severity and impact. The scanner will provide some high-level idea of the severity, but it must be further analysed concerning the company and the real-world risk exposure, policies, usage, etc. This includes how complex is the exploitation process, the business impact of such vulnerability, additional security measures related to this vulnerability and other factors defining the real threat to the organisation and based on this prioritise them. The next steps required to fix the vulnerabilities that should be solved. There are three possible approaches. Remediation, the process of fixing the issue by patching or correcting the vulnerability. Mitigation introduces a temporary solution by decreasing the impact of vulnerability or

³⁶<https://www.iso.org/standard/73906.html> - accessed on 2021-02-15

³⁷<https://www.iso.org/isoiec-27001-information-security.html> - accessed on 2021-02-15

³⁸<https://www.iso.org/standard/54533.html> - accessed on 2021-02-15

the likelihood of exploitation. The last approach is to ignore this vulnerability, because it may be too expensive to mitigate or remediate the vulnerability concerning the exploit. This should complete the process, and it should be restarted.

The process of a vulnerability assessment is a systematic review of an asset to find security weaknesses. The whole process is similar to vulnerability management. The assessment is often done by an external consultant, and it is a project with defined time scope. It consists of the same phases. The consultant within an agreed scope performs a vulnerability scanning, rates the finding according to technical severity and suggests steps and measures to resolve those security vulnerabilities, together with a business impact on the company. The vulnerability remediation or mitigation is out of the scope of the vulnerability assessment and should be done by the company itself.

2.2.1 Bug bounty comparison

The possibility to automate the process of vulnerability discovery introduces an upper hand concerning the bug bounty, which is still developing the process of automation. The vulnerabilities already published in shared vulnerability databases should be discovered by the scanning and resolved as soon as possible to minimise the risk. On the other hand, the bug bounty introduces the supervision of a human specialist and seeks new, unknown vulnerabilities. In case the company does not implement vulnerability scanning on a periodical basis, the bug bounty researchers will see many vulnerabilities available to the report even with high severity and earn a lot of bounties. The vulnerability management should be a continuous process, ideally the same should hold for a bug bounty program, but it depends on the company's policy. The common period for vulnerability scanning is one month or less. As vulnerability management introduces a process for fixing the vulnerabilities, it should exist for a company, even if the company has a bug bounty program, to have a clear process on patching the reported bugs. This should be set up in advance of the creation of the bug bounty program, to fix the simplest and publicly known vulnerabilities.

2.3 Penetration testing

The process of penetration testing, as described in [41] aims to test the security of companies, their software and infrastructure. The tester uses available vulnerabilities combined and the additional knowledge of the underlying system. The result of the testing is a detailed report describing possible attack vectors and possible threats to the organisation. Penetration testing is usually defined by a dedicated scope and with an initial set of information. The main difference with vulnerability testing is the exploitation phase of vulnerabilities and usage of vulnerabilities to gain access to the critical infrastructure of the

2. ALTERNATIVES TO BUG BOUNTY

company, which was not necessarily impacted by discovered vulnerabilities at first. The process of penetration testing is not automated, it requires a specialist who looks at the information, uses the correlation of information and interactions of the system to provide additional information concerning the vulnerabilities of the system. During the penetration testing, the results of vulnerability scanning are often used or the vulnerability testing is done as part of the penetration testing.

The tester may be provided with detailed information of the whole system, used software, network topologies and used security policies, so-called scenario is White box testing. The other side of the spectrum, so-called Black box testing, provides no information and the testers must gather this information themselves. In between, there exists a Gray box testing, with some additional, but not complete, information to the tested subject. The Black box testing aims to more real-world attack simulation, where the White box testing may provide more comprehensive details of the system and its security holes that would be hard or impossible to find without extended knowledge of the system.

The scope, described in [42], defined for the testing defines which assets should be tested during penetration testing. The scanning can focus on Web Applications, where the penetration testing may report even weak password policies, invalid business flow or sensitive data disclosure. Commonly the testing focus on the network infrastructure, where the tester checks firewall configuration, network segmentation and endpoints. Similarly, the test can scope the cloud environment, which rose in popularity³⁹ during recent years. Less often the testing consists of social engineering, which uses humans, their behaviour, to gain access and to disclose private data. A typical example of social engineering is impersonating and phishing campaigns. The testing of physical security is not that common, but it often discloses very serious problems of security.

Deviant Ollam said in [43] at the SANS ICS 2018 conference “The most important thing in the building, which is the key cabinet. You got all your expensive locks and keys, put them behind the worst lock ever.”

Independently of the scope or scenario used for the testing, it consists of multiple phases.

Reconnaissance, together with planning, starts the whole process. This consists of gathering as much information as possible, monitoring and analysing the publicly available information.

The follow-up is scanning the scope for vulnerabilities and possible security holes to identify the likely entry point to the system.

Exploitation and maintaining access to the process of compromising the system and gaining access to as many systems and data is the next step. Some

³⁹<https://techhq.com/2021/02/cloud-computing-spend-increased-by-a-third-in-2020/> - accessed on 2021-02-18

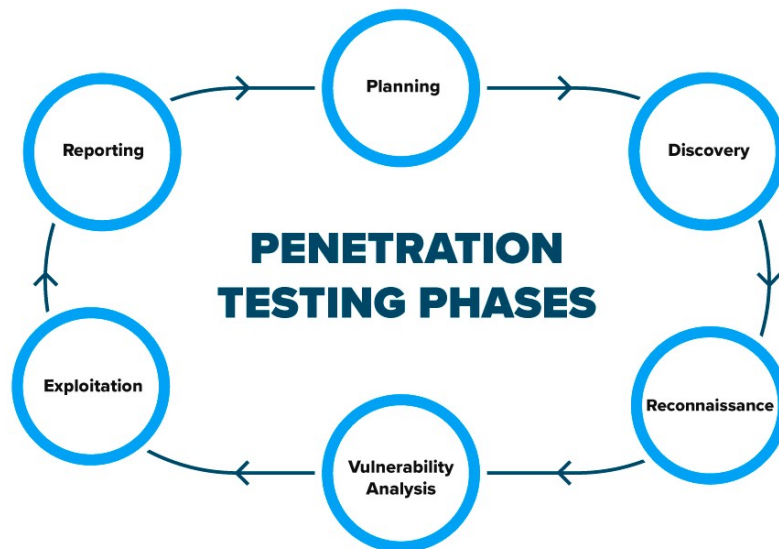


Figure 2.2: Cycle of penetration testing [6]

additional software may be installed to maintain the access or to demonstrate the impact of such an exploit.

The final step consists of reporting detailed steps used to gain access, the possible impact and disclosed data during the testing.

Penetration testing, is summarized in [41] and [42], provides many benefits to the company. It provides enhanced details of the importance of the asset, and the possible vulnerabilities with the real impact, allows the non-IT sector to see the critical need of the security. It provides protection of customers and improve the provided service, follows the policies and compliance, maintaining required security controls to auditors such as ISO 27001^[40]. It minimises the potential damage to the company image and financial loss caused by a real attack. It provides an in-depth analysis of the IT infrastructure and the ability to defend it and sustain incidents.

The requirement, stated in [41], of a trained expert, disables the possibility of big automation of penetration testing. One of the suggested solutions, that are slowly being adopted by companies, is Artificial intelligence. The development started several years ago, and nowadays, there are first serious products, conducting penetration testing with minimal or no human interaction. Still, there is a majority of the market advertising the products as automated penetration testing^[41], but the process is much more similar to the vulnerability assessment. Not only the AI can notice the context in more complex scenar-

⁴⁰<https://www.iso.org/isoiec-27001-information-security.html> - accessed on 2021-02-15

⁴¹<https://www.intruder.io/automated-penetration-testing> - accessed on 2021-02-19

2. ALTERNATIVES TO BUG BOUNTY

ios, but it avoids human error, which may be significant and with the growing scope and complexity is more and more likely. The time requirements and possibility for scheduling the test based on the company's need provide as well as an upper hand.

2.3.1 Bug bounty comparison

Bug bounty and penetration testing are one of the most similar processes. Both are usually outsourced to the external actors and test the target scope in great details, but extend the possibility of an automated scope. The most significant is the impact the actors search for. The bug bounty searches for bugs and uses this to produce a workflow, that would have a notifiable impact on the target. Submitted bugs without an impact will usually be rejected or not awarded by the company. The reasoning is, that this bug as it is presented in the report without an impact do not show the possible threat to the organisation. Therefore, the bug bounty hunter needs to step further, but still follow the scope and all the rules of the program and show that the bug may lead to something more impactful. In the case of penetration testing, the report will be composed, and it is not paid per finding, but for the whole project. Therefore, the penetration tester should as well report these bugs, that may not be currently exploitable or have no impact. Penetration testing is usually a part of compliance such as PSA⁴², PCI⁴³, etc.

The penetration testing is done by a specialist, usually having some certifications and therefore there is a guarantee of a certain quality level. This report may be presented to the stakeholders as a reasonable argument regarding the security of the product. From the bug bounty perspective, the testing is as well done by a specialist, but there is no guarantee, how much or how in details, who and what exactly was tested. Therefore, this can not be used for argumentation with compliance. The bug bounty introduces a long time testing, for the well-formatted and interesting program, the researchers and hackers will keep the testing of scope for a longer period. The penetration testing is usually conducted in a relatively short period and the cost is quite high even for no findings. Nevertheless, it is reasonable to conduct penetration testing periodically to have a complex document describing the security maturity of the tested scope. It is as well reasonable to have penetration testing before the bug bounty program and resolve the reporting, as the not significant findings by penetration testing may be exploited over time.

As well as the White box testing, the tester has an upper hand concerning the bug bounty, which is almost always a black box testing. The exploitation even though the impact is important can not exceed a certain threshold in bug bounty. The company is not aware of the testing of the specific element at

⁴²<https://www.investopedia.com/terms/p/public-securities-association.asp> - accessed on 2021-02-19

⁴³<https://www.investopedia.com/terms/p/pci-compliance.asp> - accessed on 2021-02-19

a specific time, when the bug bounty tester tries to exploit the vulnerability further. Therefore the testing should be stopped at a certain level and report, on the other hand, the penetration tester performs this in a certain scope and the technicians are aware of this and the penetration tester can exploit the vulnerabilities a bit further.

2.4 Red and Purple teaming

“The best defence is a good offence”, a famous quote by an unknown author defines the meaning and purpose of Red teaming, it helps business to remain competitive while securing their processes and interests. Red Team described in [45] has its origin in the military and its techniques to conduct a stress-test strategy, discover the weak point in the entire structure and test the countermeasures of the organisation. It is based on the group that plays the role of an intruder, enemy or competitor intending to evaluate the security status of an organisation. Red teaming is often confused with penetration testing as there is no exact definition of what is complex penetration testing and simple red teaming.

A Red team assessment, as generalized in [43], is a goal-based activity with a general scope, usually the entire department or whole organisation. The complexity and combinations of techniques result in a complex testing procedure, that concerning the penetration testing requires a longer period of few weeks to multiple months. The purpose of red teaming is to have a practical demonstration and testing of a complex real-world attack using seemingly unrelated exploits to achieve the penetration or control-takeover of the company. It is not used only in cybersecurity, but secret services, the army and many other subjects are conducting red teaming to constantly improve and test their ability to react to newly emerging threats.

The red team, as described in [26] and [46], often involves penetration testing as well, but as a simulation of the attack. The testing is less aggressive and takes longer, to not raise an alarm of the security teams operating in the company. It involves as well as the testing of the security perimeter, education and awareness of the staff. It often involves complex phishing campaigns to obtain access to devices within the organisations, preceded by software engineering to obtain as much information as possible regarding the internal functionality of the organisation. Regarding the security perimeter and awareness of the staff, an impersonation of the employees, 3rd party service providers or technicians is used. Due to this complex technique involved and simulation of an external intruder the testing is performed by hired specialised outside the company.

The testing procedure, stated in [47], is constantly evolving and threats and methods used by an attacker change with time, therefore the red teaming has not the exact procedure, but the testing can be divided into several

2. ALTERNATIVES TO BUG BOUNTY

phases. Outline the problem in the initial phase of the testing. This phase will precisely define the objectives and describe the target of the testing and make a legal agreement between the team conducting the testing and tested organisation. The second phase is usually the longest. It consists of a diagnosis of the target. This consists of gathering as much information as possible by any measures, but not triggering the alarm. It includes understanding of the corporate hierarchy and interactions with the external world and understand the permissions related to actions and actors. The next phase is used to challenge and verify the discoveries from the previous phase. It uses an interactive technique with employees and devices to exploit the potential security weak points. Even though this phase involves some confrontations with the organisation, its assets and employees, it should be conducted in a stealth mode to avoid triggering the alarms of any countermeasures such as physical alarms, firewall and access rules, guards attention or employees suspicion. This full-scope attack simulation targeting multiple layers of the company such as the network, employees and additional security measures, to check if it can withstand a real-world attack. The test exposes vulnerabilities and risks regarding technologies⁴⁴, people⁴⁵ and physical⁴⁶. Especially human error is often overseen in security testing, such as penetration testing, or vulnerability management. The InfoSec Institute concluded in [48] “6% to 28% of the attacks are conducted with the help of current or former employees of the infected organisations.”

The result of the red teaming is not just a report listing the security holes and methods used to exploit them. They will provide alongside this report a plan on how to improve the defence for future attack and cooperate with the security teams to consult the implementation of countermeasures and patching of the security issues.

As stated by Tim Bryant in [49], the red team is used to attempt to penetrate the system, there is a Blue team (*[49] performs an analysis of information systems to ensure security, identify security flaws, verifies the effectiveness of each security measure*) trying to protect the company from intrusions. Mostly speaking about the intrusion of technologies by using monitoring, filtering devices, educating the employees and working on security policies. One of the drawbacks of red teaming is its time demand to obtain any information on what to improve. This introduced a new concept, so-called purple teaming. The purple team consists of the red team and blue team representatives. The aim of the testing is again to see how the organisation will withstand a complex attack and improve the countermeasures. The purple team has enhanced knowledge of the system that is being tested in the blue team part, therefore the testing of the red team can start much faster and with many precise and

⁴⁴Network design, firewall, applications, routers, appliances, antiviruses, etc.

⁴⁵Employees, externalists, 3rd party suppliers, business partners, departments, etc.

⁴⁶Access control, locks, physical security of technologies, etc.

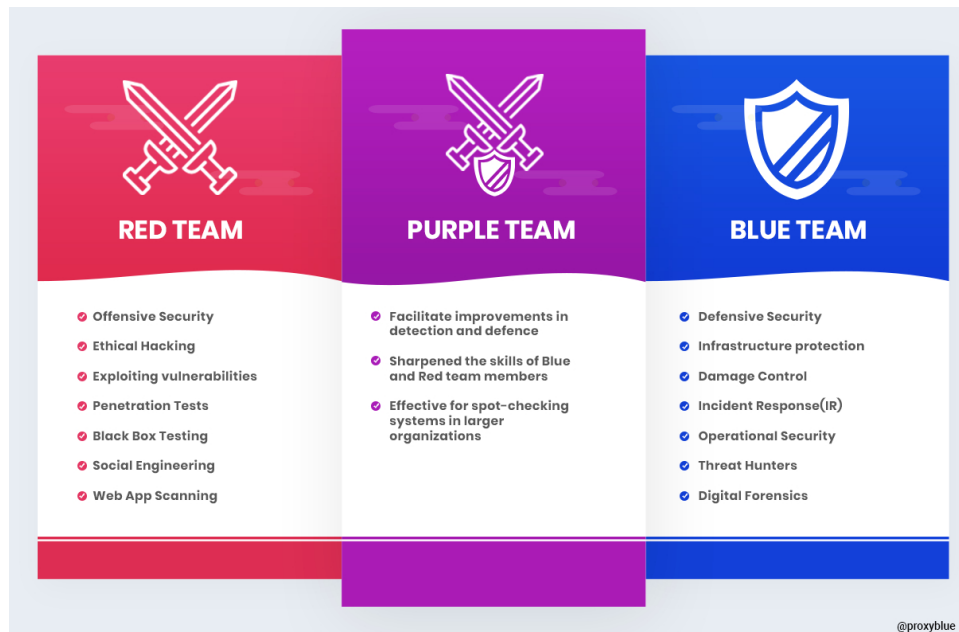


Figure 2.3: Comparison of red, blue and purple team [7]

sophisticated scenarios. After conduction, during an attempt to overcome any of the countermeasures the team immediately can check the logs and alerts that had been triggered, discussed the reasoning of the result and improve the protective mechanism. The whole process, therefore, maximises the effectiveness of the Red and Blue teams, to obtain to be the best result in a shorter period.

2.4.1 Bug bounty comparison

Comparison of the red and purple teaming against the bug bounty is not as trivial as it seems, the main point of non-triviality is the boundaries that should define activities as part of the teaming. In the case we consider the red teaming as an enhanced penetration testing, and include as well as an additional area of the security such as phishing attack or the physical security this form of red teaming is comparable in many factors to the bug bounty. The red teaming is the first security practice testing additional aspects, not only the software or network implementation. As well as a bug bounty, the red teaming considers testing of the hardware, but the red teaming often includes phishing campaigns and social engineering, which is usually defined as an out-of-scope subject for a bug bounty. It is as well common, that the red team is provided with the devices to be tested, but in the case of bug bounty programs, the hacker or security specialists must obtain the hardware on their own and even when the findings are significant, usually, there is no

2. ALTERNATIVES TO BUG BOUNTY

additional compensation to the expenses than the standard bounty. As well the physical testing of the security is not common in bug bounty, but the red teaming may abuse things such as keys or card readers. Therefore, the red teaming takes the testing step further by including additional attack surfaces, but usually, the red team generates one comprehensive report at the end of its testing, which may take even multiple months, where the bug bounty reports are delivered usually within a few days from discovery.

The Purple team is partially similar to the communication, that may be held by the triage team together with the researcher. But the purple teaming is more about the fixation of the issues and not only reporting. The closed campaigns allow the company to have a closer collaboration with the security researcher or hacker, and it may involve as well communication on the level of the purple teaming by the discussion of specific alert triggers and data modifications during the testing. However, due to the number of testers in public programs, this is not applicable and even in private programs, this is not common practice.

The practice of the red and purple teaming is similar to the bug bounty in multiple aspects, but as well use additional measures, that should not be conducted without exact written agreement, such as physical security testing. From the perspective of software these practices are similar, the bug bounty will not exploit the vulnerability in such details but will deliver the report much earlier, where especially the red teaming will take the exploitation as far as possible but will postpone the report till the end of the testing.

2.5 Security auditing

A security audit, defined in [50], is a top-down look high-level description of the business and all aspects of it. Security audit works with the technical aspect, such as Vulnerability scanning, password policies, disaster recovery plan and others, the non-technical aspect contains a design, policies a procedure used within the company. The audits should follow a workflow consisting of defining the assessment criteria, preparation, conduction.

As the process is described in [51], initially, the auditor determines together with the organisation representatives general objectives and assign them priorities. Agree on the method used during the auditing and track, define the scope and out-of-scope areas. The preparation consists of planning a schedule based on priorities defined in the initial phase and agreed methodology. A set of tools is selected, and a specification of methodologies is done, besides a survey is created as well to gather the additional data for the audit. The final phase consists of conducting the audit and report creation. The auditor supervises and controls the policies, up-to-date processes and infrastructure are used and the correctness of their usage. Besides, it conducts a series of tests to guarantee that the expectations and requirements match the observed



Figure 2.4: Common steps in security auditing [8]

situation.

The audits are related to the verification of policies and well-known standards as ISO 27001^[47] or PCI^[48]. The audits are conducted either internally, used for building up the policies, infrastructure, etc., or externally for bigger companies with well-established processes and policies, where audits are done by an external specialist. The audits are usually conducted continuously to follow the newest trends and to mitigate newly observed threats. The one-time assessment is performed on behalf of a trigger to check the correctness of the process or new compliance, policy or technology. The auditors may be as well automatized and provide an additional level of security to avoid humans errors.

The auditing by itself does not provide any security, it simply controls the established security procedures, protocols and technologies being used. There is no test of the correctness of established measures.

⁴⁷<https://www.iso.org/isoiec-27001-information-security.html> - accessed on 2021-02-15

⁴⁸<https://www.investopedia.com/terms/p/pci-compliance.asp> - accessed on 2021-02-19

2.5.1 Bug bounty comparison

At first, the security auditing differs from the bug bounty process. However, as the bug bounty keeps developing and new programs are being created with different scopes, there are slowly some similarities emerging. Few campaigns are already focusing on the compliance, data process task, business flow and additional process aside from the software testing. Currently, the security auditing is still focused on more details on the internal processes of the organisation, and it would be almost impossible to transfer this to the public bug bounty programs. However, the private programs may provide in the close future additional control to these. Similarly, as in the situation with the penetration testing, the report created during security auditing can be used for PSA⁴⁹, PCI⁵⁰, etc. or as part of the presentation for stakeholders. This form of report is something uncommon in the current bug bounty, but it may change with the evolution in the future.

2.6 Summary

There exist many roles and procedures to achieve a certain level of security. The bug bounty is introducing a new approach, that should be implemented alongside additional measures such as penetration testing and security auditing. An important thing to the company is not to look for bug bounty programs too early, as there exist tools that can even automatically discover vulnerabilities, such as vulnerability assessment, check the possible vulnerability exploits and conduct comprehensive security checks. The findings from vulnerability assessment and penetration testing should be handled earlier. The bug bounty stands out due to its simplicity, initial investment and a huge number of specialists in the various area checking the scope and are motivated in a better way than using an ordinary Vulnerability disclosure program. By having a bug bounty program the company should not stop doing the other security measures, but keep evolving the security sector to protect their business and customers.

The bug bounty brings a good wider range of test objectives, not only the software but even hardware or private data disclosure, which are not necessarily in the scope of red or purple teaming. In general the bug bounty is mostly similar to penetration testing and red teaming. Therefore, the bug bounty is something, that companies should consider implementing or outsource to 3rd party companies and use for a long time to have constant control of the security.

⁴⁹<https://www.investopedia.com/terms/p/public-securities-association.asp> - accessed on 2021-02-19

⁵⁰<https://www.investopedia.com/terms/p/pci-compliance.asp> - accessed on 2021-02-19

Analysis of available products

In the following section, I described two globally popular bug bounty platforms and two platforms with orientation on the European market with a smaller base. Besides my personal experience, I had a general discussion with users and companies using these platforms. The discussion had three sections, usability and experience, the positive points and the downsides. I have received requests to not publish the names of some interview, based on these requests I have decided to keep all names to myself only. The general information regarding the company was gathered from their respective websites.

3.1 HackerOne

HackerOne⁵¹ is a platform providing a bug bounty as a service alongside a multiple other services including penetration testing, etc⁵². The company, as summarized in [33] was founded in 2012 by Jobert Abma with Michiel Prins. The company was founded with motivation in a recent event of Hack 100, the discovery of security vulnerabilities in 100 prominent high-tech companies. The company started to offer an internet bug bounty project, today recognized as a public bug bounty program.

Nowadays, the HackerOne together with Bugcrowd is one of the biggest companies⁵³ offering a Software-as-a-Service for bug bounty, and to support the community of the hackers using their products additional educative and practising tools have been included. The company as well expanded the portfolio of offered services, therefore HackerOne now offers multiple different products and when possible the ISO standards are followed. Alongside the bug bounty program, the company offers penetration testing to further strengthen the security of their business partners.

⁵¹<https://www.hackerone.com/> - accessed on 2021-03-05

⁵²<https://www.hackerone.com/product/overview> - accessed on 2021-03-05

⁵³<https://www.linkedin.com/company/hackerone> - accessed on 2021-03-05



Figure 3.1: HackerOne logo [\[9\]](#)

HackerOne has two general approaches to any campaign and company regarding the involvement of the triage team. The triage team can either manage the program or provide just basic support, and it is up to the customer to do all the necessary things. The managed program outsources the communication process to the triage team and the customer cares only about verified findings and vulnerabilities with multiple details already gathered by the triage team. For not managed programs, as the communication must be done by a company, the involvement of SOC or any unit responsible for security is necessary and it must verify the vulnerability reports, check for duplicates and gather additional information from reporting users by themselves.

3.1.1 Personal experience

HackerOne is an open community and anyone can create an account. After the user logs in, the website itself starts to be much more clear and organized. Besides the customisation of a profile and connection with multiple accounts of other platforms such as GitHub, LinkedIn or Bugcrowd, the user can see the overview of its activity and obtained a rating, which is important for an invitation to private programs. To increase this rating composed of Reputation, Impact of findings and Signal, which is an average reputation per report the user should positively contribute to the bug bounty programs, or complete some exercises.

As the reputation is gained by the submitted reports and thousands of users are searching for the vulnerabilities, the gain may take a lot of time and frustration. The HackerOne introduced a learning section, that is used to train the hackers by educative content and more importantly by exercising the Capture the Flag. A successful participant in these exercises provides some basic rating to a newcomer hacker and serves as proof of knowledge. The possibility of practicing is handy to maintain or improve the skill set.

In the case of public programs, the filtering by program features and assets type is supported. In the detailed description of the chosen program, the user is capable of reading all the detail provided by the company, such as scope, reward range, etc. The user can interact with a public program by bookmarking it, to have easier access to the program from its profile, subscribe to receive updates and submit reports. In my opinion this is handy feature to have a nice overview of programs on the profile.

The report submission may get trickier for a new user. Some programs

require additional constraints, such as two-factor authentication. Besides, the user has four trial reports after the account creation. This is related to the rating calculation, a new user needs to obtain certain fame before there are no limitations to report submitting. The report creation guides the user through the entire process, with a list of available scope, that is impacted, the weakness type discovered and severity. Besides, the user creates a title of the report, description and impact in a separated text field, optionally files may be attached. Then the submission composes the report automatically. Generally speaking the report creation is intuitive and simple.

3.1.2 User's experience

Based on several discussions the users have commonly agreed, that the HackerOne besides the standard bug bounty provides reasonable educational content, which is well explained and documented. They as well agreed, that the capture of the flag is handy in the case of a newcomer, they can sharpen their knowledge and more importantly to obtain a certain rating to receive some private programs invitations. They saw as a very positive the number of programs available, the simplicity and clarity of the GUI and nice guidance during report creation. The report tracking process and its reliability were commonly identified as positive.

As negative points, some of them identified the triage team. They complained about inexperienced behaviour, asking to exploit the vulnerability more than necessary, even at the edge of allowed scope and common practices of a bug bounty. As well as additional information from the communication with the triage team or directly with the company is not well, if ever, presented within the report. Hackers would as well appreciate some additional fame generation for example tutoring or code development. As a last noticeable point, some of them noticed, that the huge crowd has a disadvantage in the users' perspective, as more people scan for vulnerabilities it is harder to have a finding and especially for the new users this may get tough.

In a general point of view, the HackerOne is seen as a good company, with a good community, that will be expanding in the future, and they would recommend it for usage, with a side note, that finding the bugs in a real application is not easy, but it should not stop the users from doing so. Important to say, that none of the respondents does the bug bounty as a full-time job. They reasoned, that it takes a lot of experience and practice to reach such a point without guaranteed income.

3.1.3 Company's experience

Two companies provided answers with their experience. Both of them enjoyed the possibility of additional products available alongside the bug bounty, especially the penetration testing. As well as the possibility of well-designed

3. ANALYSIS OF AVAILABLE PRODUCTS

private campaigns. The saving concerning penetration testing, red teaming, source code review, etc. was so significant and together with the saved human resources one of the key motivations to preserve the bug bounty program after the initial plan.

They appreciated the number of testers and the detail of the conducted tests on their web applications. The process of registering an establishment of the agreement was straight forward. Besides, the limitation of scope and all the vulnerabilities to be reported is easy to control and in general are these conditions satisfied, moreover, the triage team helps them to remove the out-of-scope reports, to save human resources and time.

On the other hand, the notice of occasional duplicates was quite often even though, the triage team was involved. In case, the reports are managed by the company, the severity is not always exact and therefore even less severe vulnerabilities consume quite a lot of time.

A major missing point is the possibility of integrations, for example with git services and servers of various kinds, companies would as well enjoy fetching reports and issues from additional applications, such as Git and visualise them in the HackerOne portal. More options for analytic view directly into the portal would be appreciated as well. Better support during the entire workflow and more possibilities would be appreciated as well with the collaboration of the triage team.

3.2 Bugcrowd

The Bugcrowd⁵⁴ is a platform concentrating a security researcher providing a various palette of products including penetration testing⁵⁵, bug bounty, vulnerability disclosure⁵⁶ and attack surface management⁵⁷. The platform is popular for the high number of skilled hackers and security researchers, together with a huge number of companies using the platform for hosting bug bounty programs. The company was founded in 2011 by Casey Elli⁵⁸. Initially, the project was started as a startup aiming to security and provide a possibility for hackers and security researchers to submit finding of a company and potentially earn a reward. The impulse was originating from the event of Hack 100, which was well accepted by the companies.

Nowadays, the platform is one of the biggest providers of the bug bounty

⁵⁴<https://www.bugcrowd.com/> - accessed on 2021-03-07

⁵⁵<https://www.bugcrowd.com/products/penetration-test/> - accessed on 2021-03-07

⁵⁶<https://www.bugcrowd.com/products/vulnerability-disclosure/> - accessed on 2021-03-07

⁵⁷<https://www.bugcrowd.com/products/attack-surface-management/> - accessed on 2021-03-07

⁵⁸<https://www.bugcrowd.com/resource/bugcrowd-founder-and-ceo-casey-ellis-on-the-future-of-crowdsourced-security/> - accessed on 2021-03-07



Figure 3.2: BugCrowd logo [10]

platform as a service with more than 1000 employees⁵⁹. The company developed as well a different product, to serve as a more general security platform used for outsourcing multiple services. The security researchers and employees of the Bugcrowd are involved as well in products such as penetration testing. Concerning competitors, the Bugcrowd aims more to the community of hackers and researchers to have an upper hand. The community is driven by Bugcrowd University, a project used to educate and connect hackers and researchers. It is supported with news and blog posts related to IT security and by a global leaderboard to keep up the motivation for bug hunting. There is as well automation in security workflow and enhanced crowd analytics provided as the usage of AI in the process.

3.2.1 Personal experience

The original page is in my experience a bit more well organized and more clear concerning the HackerOne. The account creation is simple, and besides the confirmation, you will be receiving some emails explaining the general concept of bug bounty or online resources for news feed and educational content. These emails are nice for beginners as they will share with them useful resources and practical examples to guide them in the initial steps. For more experienced hackers and researchers, the email may get annoying, and automatic unsubscribe is possible at the bottom of every single email.

The initial login will introduce to the user simple decision branches, to describe its current knowledge and experiences, to guide the user in the first steps. This process may be skipped as well, to not bother uninterested users. The user can then select more specific areas of interest. These skills and interests can be edited later on in the profile. The information of interests and skills are used to propose programs matching these constraints, to highlight programs more suitable to the users. This is in my opinion great to initially set up the main area of focus.

Besides the standard editing of the profile, the dashboard shows an overview of the payments, recent programs and recommended programs. In addition, the dashboard shows an overview of the tasks and activities done by a user and announcements made by a companies, to show the latest changes of programs. A user can connect platforms as GitHub, Stack Overflow and Pentester lab, to obtain additional fame and increase the prestige for private

⁵⁹<https://www.linkedin.com/company/bugcrowd> - accessed on 2021-03-07

3. ANALYSIS OF AVAILABLE PRODUCTS

programs. The user may as well develop a simplified resume to provide an overview of skills and experiences to companies to have a higher possibility of receiving invites to the private programs. The idea of rewarding the users for activity on platforms such as Stack Overflow is great and supports the community greatly.

The programs can be filtered by properties, that must be typed into the search box, the suggestion of such properties is provided, however not all are listed and for new users, this is not ideal. Every program has details, with all the necessary information related to the program as a scope, reward, safe harbour and others. In case some section is missing, the program may follow Bugcrowd's standard terms. An overview of announcements is provided as well as all the changes announced during the program lifetime. An interesting feature is the possibility to join a program with extended criteria such as ID verification or two-factor authentication.

The report is guided and provides dropdowns for scope and vulnerability locations. Additional text boxes allow to provide further information and dumps/requests used for the exploration. It as well offers a preview of the section used for formatted text. As the last provided data, files can be attached. The last is a checkbox for confirmation of the program brief, terms and conditions. The section contains as well as the disclosure policies used for this program links to rules, scope, etc. The entire process of reporting finding is well understandable and all important information are highlighted during the reporting process.

The section of CrowdStream provides an overview of disclosed and accepted submissions. The same is for the leaderboard with an overview of the best users. But other than that, there is no interesting data. It is nice to have possibility to check other findings, but it does not contain much details.

The Bugcrowd does not provide a limitation to the new users and the rating requirements are not put as high importance. The Bugcrowd University does not provide a possibility to obtain a rating, or at least it is not so clear as with the capture of the flag provided by HackerOne. The university consists mainly of webinars and podcasts with a topic related to bug bounty and IT security. The educational webinars used to introduce tools and methodology to the new users also contains the link to the GitHub of Bugcrowd, with an overview of lectures, with webinars and additional resources. The lower importance of fame is good to decrease the pressure, but it is harder to obtain the fame to start receiving invitations to private programs based on my experience.

3.2.2 User's experience

A general discussion was held with several users having long experience with bug bounty, however, none of the users does do currently the bug bounty as a full-time job. Most of the people work in some kind of IT security company or cybersecurity team and using the bug bounty to earn an extra bounty and

sharpen the skills. They agree that the dark mode, enabled by default is pretty nice, although they do not see it as a positive argument for users to join.

The users have spoken positively of the support team, the quality of provided answers, as well as the time required for the answer. The same was true for the triage team during solving the reported vulnerabilities and their validation. The default policies and rules applied to all the programs not specifying their constraints are seen as something important and positive, as it ensures a legislative cover to the users. The target listing based on filled specialisation provides a straight-forward possibility to initiate testing right away. The academy and additional resources are nice to have, but from their perspective, it provides no additional value, as they are more experienced. The same holds for the news feed, as they have custom feeds via Twitter or different media. The last positive point was about whitelisted programs. For such program a user must apply and match the requirements, they lack such possibilities on other platforms.

Despite the simplicity of the reports and the graphical user interface, there can be a reasonable improvement to make it more intuitive and with more information included. The most significant criticism was about filtering. A huge improvement should be in the filtering possibilities. The current is not as intuitive and especially if the user does not use it often it is hard to write an exact filtering option. They would appreciate having a more straight-forward possibility to participate in additional products of Bugcrowd, especially penetration testing, paid per finding as well. For some of them, this would be an argument to start doing it as a full-time job.

3.2.3 Company's experience

The company can use nice additional services, as there is no need to search for an additional provider. The bug bounty offers two general concepts regarding the triage team. The managed service that provides comprehensive reporting and feedback to the company. The second possibility is just partially assistance and on demand consultations. The company can as well select three different options for deploying their programs, the public, most common and available to anyone, whitelisted, where users can apply for it if they match constraints and private program, where the company sends the invites to the chosen candidates based on selected constraints.

The most often mentioned advantages are without doubts the huge number of researchers working on the programs. The help of the Bugcrowd with customisation of the entire process, as well as the suggested remediation included with the reported vulnerabilities is highly appreciated across the companies. The support with the project management and bug tracking is well organized and handy to all the companies. Another positive point is the relative cost, concerning the HackerOne. The last point worth mentioning is the possibility

of integration with multiple tools. There are still tools, that would be nice to have integrated, but concerning the competitors, the offered tools satisfied the companies.

The success of the program is strongly influenced by the moderator and based on the discussions, not all moderators are performing approximately at the same level. In contradiction, customer support tries to resolve this issue with high priority, therefore the companies do not see this as a huge problem. Enrichment of the interface for a more detailed overview and reporting would be appreciated as well, but the possibility of outsourcing the data via an integrated tool satisfied most of the companies.

3.3 Intigriti

The company⁶⁰ focus on the security testing of the companies assets provided by the crowd of hackers. Concerning the big companies, Integriti is more known and focused on the European market. The company was founded by Stijn Jans in 2016 and currently employs a few hundreds of customers⁶¹. The company does not provide additional services other than bug bounty but propose some additional services to researchers to match all the constraints of customers, for example, a VPN connection.

3.3.1 Personal experience

Immediately when the home page is opened, the content is clear, the menu is simple but easy to navigate. This is in my opinion caused by the simplicity and limited service offered by the company. There is only hosting of the bug bounty, no additional security services provided to the customer, no dedicated section for education, nor practice. In the section called Bug byte, the cybersecurity news is not intrusive, yet well presented. Moreover, the post of Bug byte contains usable resources and links to compensate for the lack of an educational section. This in my opinion is good enough for the smaller platform which Intigriti is.

For users, the dashboard displayed after the logging provides a nice, simple overview of lastly update programs, together with an overview of upcoming bounties payments, reputation and global ranking. The Dashboard is simplified but contains important information, this helps the user to quickly understand and notice all the information displayed. The overview of updated programs is good and important, as a new not tested target to try and I like it.

The profile does not allow complex customisation. It provides a nice and simple overview of the activities, containing ranking and reputation, as well

⁶⁰<https://www.intigriti.com/> - accessed on 2021-03-10

⁶¹<https://be.linkedin.com/company/intigriti> - accessed on 2021-03-10



Figure 3.3: Intigriti logo [\[11\]](#)

as links to the accounts of LinkedIn and Twitter if provided. The leaderboard exists as well, but the profiles there do not provide you with a lot of information regarding the findings, to use as study material. As the platform is not so popular, the most known names of bug bounty hunters are missing. In my opinion, this needs an improvement in future.

The browsing program section provides an overview of the programs. The default ordering of the programs is alphabetically, but in my opinion, this is not ideal. The filtering of programs is done nicely by dropdowns with multiple checkboxes. This filtering is therefore intuitive and straightforward even for new users. A nice feature is a short text for each program, it usually briefly describes the company and the program in general. Moreover, the programs are associated with information of the last update and last submission from any researcher. Some programs require a check of an ID. The programs can be viewed, but there is only a description and a bounty range, no further information of the scope, rules, etc. but the information of the ID check required is not well highlighted. For programs without ID check, the same place is occupied by another banner with the same style having just some informative value, which is in my opinion, not the best solution.

The program is associated with a standardised format containing all the required information. Besides, it provides a list of users who did the most recent submissions with a date as well and the biggest contributors. The option to subscribe to the program is available, as well as the possibility to ask an additional question regarding the scope for clarification. This is in my opinion handy to avoid any problems in future caused by misunderstanding.

The report submission works as other bug bounty programs with dropdown lists of available scope and vulnerabilities. The text boxes to enter a detailed

3. ANALYSIS OF AVAILABLE PRODUCTS

description, impact and recommended solution, these sections can be nicely formatted, but there is a lack of previews, therefore it may be hard to visualise. To visualise these sections, as well as the report as a unit, it is required to proceed to the next step containing only the preview. This is not ideal in my opinion. At the very bottom is an optional section to provide an IP address used for the exploitation, which is in my opinion good for the company, to check the logs and filter the activities. As a nice feature, the report is being automatically saved as a draft of the reports to allow a modification and submission later on.

3.3.2 User's experience

The user experience of the platform integrity is overall positive, but some points for improvement were discovered and some controversial points emerged as well. The point of the number of programs and users is not clear whether it is positive or not.

The smaller number of users based on the asked participants provides an advantage of the better chance to have a finding, but it goes hand in hand with the lower popularity of the platform and fewer users to learn from. The same is true for the number of programs. Few programs do not attract a huge crowd of users and again the users see this as a better chance for findings, but they as well realise and note, that the selection of programs is really limited. Mixed feelings are as well for the page layout, not everyone likes the simplified view with just a few items.

A positive aspect of the platform is the section dedicated to frequently asked questions to most programs. The possibility to have automatically computed CVSS during report creation based on the finding and impact is handy, as well as the automatic saving of the reports. In case the report is classified as a duplicate the rating is still defined, but only one-fifth of the original findings, but the users like this fact.

A major disadvantage of the platform they highlighted the display of the programs per page, having only 10 programs is slightly annoying. Some additional tags can be shown on each program as well. Also the community needs some improvement, to attract more users.

3.3.3 Company's experience

The feedback obtained is generally positive for the hacker's community, even though it is much smaller concerning the most famous providers, the community is filled with professionals. A lot of the users participate as well on Slack and allow the company to communicate directly with them. This was one of the most positive things mentioned by the companies. The onboarding process is well established and supported by the triage team, which is also having positive feedback on reliability and quality of cooperation.



Figure 3.4: YesWeHack logo [12]

The thing to be improved is the overview of reported findings, the visibility at the management level, better highlight remediation and the lifecycle of the finding, as the companies would appreciate some more information. Integration with more external tools would be handy as well, but improvements to the overview are not crucial.

3.4 YesWeHack

YesWeHack^[62] is a company with a main focus on the European market, it is based in France and follow all the constraints for compliance with the European legislative. The company was founded in 2013 by Guillaume Vassault-Houliere and Romain Lecoivre.^[63] Currently, the company employs a few tens of workers. The company offers an education to the users enhanced by real-life situations, and it aims to raise the awareness of the public crowd in the security field. The educational platform offers different levels to sharpen the skills in a particular area. To enrich the community, the company provides as well the job board for the cybersecurity and non-profit platform for Coordinated Vulnerability disclosure alongside the bug bounty platform. The company is raising quickly and therefore in future it may become a new big player in the cybersecurity area, as the building block is solid.

3.4.1 Personal experience

The main page of YesWeHack has a professional view, with a lot of fancy animations and scripts running, which I dislike. It disturbs the attention from the main points, as well as not for all browsers it works so smoothly. The most important information is at the very top, to provide the user with an immediate possibility to reach the desired section. After the login, the user interface gets much clearer with a view of the submitted reports.

The list of the programs provides a good overview of submitted reports, which is nice. The lack of filtering, only searching by name is not ideal at all,

⁶²<https://www.yeswehack.com/> - accessed on 2021-03-10

⁶³<https://www.linkedin.com/company/yes-we-hack> - accessed on 2021-03-10

3. ANALYSIS OF AVAILABLE PRODUCTS

however as the number of programs is limited it is not a big problem overall, but I would appreciate advanced filtering anyway.

The details of each program are well formulated with all the information necessary, also the overview of the activity done by users with a program is nice. The information of the response time and reports in some period is handy, and the list of accepted languages increases the possibility for more hackers to participate, which I like. The language contains English or French mostly. I have not seen another language. This is expected, as the company is based in France and the triage team is mostly composed of French and English-speaking people. But if the language variation will increase in future it could provide a great feature.

The report is compact with all the required information and drop lists for bug type and scope. The calculator of the CVSS is handy as well. The requirements to provide the IP address is nice, but not necessarily appreciated by all the users. The section to the bug description has a template on all the formatting such as headers, list, markdown, etc. but it looks chaotic and should be separated into two or more separated sections in my opinion. On the other hand, the immediate preview is handy to visualise the report, and it simplifies the formatting a lot. The last possibility to chain this bug with another is really handy and may save up a lot of writing and discussion with the triage team.

The ranking of the users in the global scope of the platform is nice, but immediately the notice is in the domination of the French users. This may be caused due to the low popularity outside of France and can change with the increasing popularity of the platform. The overview of users in the leaderboard is nice and provides handy information regarding the reports and a few additional details, such as Twitter or GitHub of these users.

3.4.2 User's experience

Users appreciate the graphical composition of the platform useful for researchers, the presentation of available programs is great as well. The possibility of a different language is, based on the discussion, good idea, but now it is limited mainly to the French speakers.

As a negative point is seen in the limited offer of programs, as well as the variety of the different types of bug bounty. This is based on the discussion with the users the most significant problem. The missing filtering is not so big a problem due to the low number and variety of programs. The users believe, that as the company is growing in popularity this will change soon with the arrival of more companies.

3.4.3 Company's experience

The companies speak positively about the possible modification of the process and support during the establishment of the VDP, management of vulnerabilities and the entire process of fixing the findings. As well as the assistance during the assessment and scope selection, to construct an effective program is appreciated. An exposed API using JSON format allows integration with any arbitrary system is highly valuable by the questioned companies.

The major disadvantage is the dashboard from the reported bugs and general findings, either more detailed and clear information would be required or the possibility of the customisation. The location and focus of the YesWeHack are not ideal as well, as some bigger companies would like to include a department outside the EU. The possibility set as well as an additional language for the report is nice but is not appreciated for companies not located in France.

3.5 Unsatisfied needs

According to the research conducted and based on the answers gathered I identified several weak points common to all bug bounty services. Few unique insufficiencies were identified, but those are flaws regarding specific services rather than flaws in the overall design. Human related work was mentioned in the majority of interviews. Triage team's incompetence is a major problem on all four researched platforms. Mainly triage team not being able to identify duplicates, decide severity factor or support companies with program creation and users with submitting reports. In the not human related category two main flaws were identified. Not enough integration with external tools and not exposed API calls are viewed as negatives by respondents. Being able to directly create Issue on Git repository or have possibility to automate this process would be very appreciated. Some services were defined by having overpopulated webpages or having way too simple interface with not enough features.

3.6 Summary

As the bug bounty is relatively new and is increasing in popularity, it is a new possibility for companies to expand the portfolio of offered services. The market is not yet overloaded by many companies and the current solutions still have holes that need to be improved.

One of the most significant deficiencies of the current platforms excluding human resources is the possibility to integrate it with other tools. Alternatively to consume these data by external tools using some standardised format via API calls. The last of the most significant downsides of existing solutions

3. ANALYSIS OF AVAILABLE PRODUCTS

is the consistency of the available information, as the provided data differ over program by program significantly.

The conducted discussion resulted in the findings, that the triage team and generally the performance of the employees are one of the most influencing factors for the platforms. Besides this, the platform should provide a simple overview, which is easy to navigate. Any data input should be prefilled if applicable.

The triage team should be as well ready to support the company during setting up vulnerability disclosure policies and the platform should have an easy possibility to deliver to the customer's additional security products, such as penetration testing or red teaming. To allow automatisation, the platform should provide integrations to common tools.

To keep the better interest of the users, the community should be strengthened by news feeds and the possibility to participate in other products if applicable. Any form of material with an educational purpose to practise the skills will be beneficial to the community as well, however, these sections are already exceeding the scope of the bug bounty platform itself. To have a successful bug bounty platform, it is not enough to provide only with the bug bounty, but rather a more comprehensive and general platform should be used alongside other offered products, to be competitive to existing products.

Product Design

This section provides a high-level design for a bug bounty platform based on the information and experience gathered in previous sections.

4.1 Requirements

Requirements specified before the analyses contain only the points for the B2B platform with a closed campaign. Based on the gathered information, the platform should have additional requirements, to be competitive on the market. The bug bounty platform should contain a private program as well and additional material for the users, to prove their skills, gather the rating and have a better chance of having an invitation to any non-public program by a Capture the Flag. A requirement out of scope for this document is the establishment of a high-quality triage team and a good advertising policy, to attract enough companies and hackers and support the community.

4.2 General Description

The bug bounty platform will consist of several components. Some of these components will be programmed, while others will be implementations of pre-existing programs. The language used will be decided based on the requirements stated in this chapter and the tools used by the developers. Multiple users will be able to log in and interact with the program at the same time. It will be set up using three different user roles as seen in the figure 4.1. First is the basic user, the hacker or security researcher, which can browse programs and submit reports. The second type of users are companies, customers of the bug bounty platform. Companies are the only users that can create new programs. Programs are a request made by a company to find bugs in their software, hardware or infrastructure. The third group of users is the Triage or Admin team member. Their purpose is to moderate programs, check if

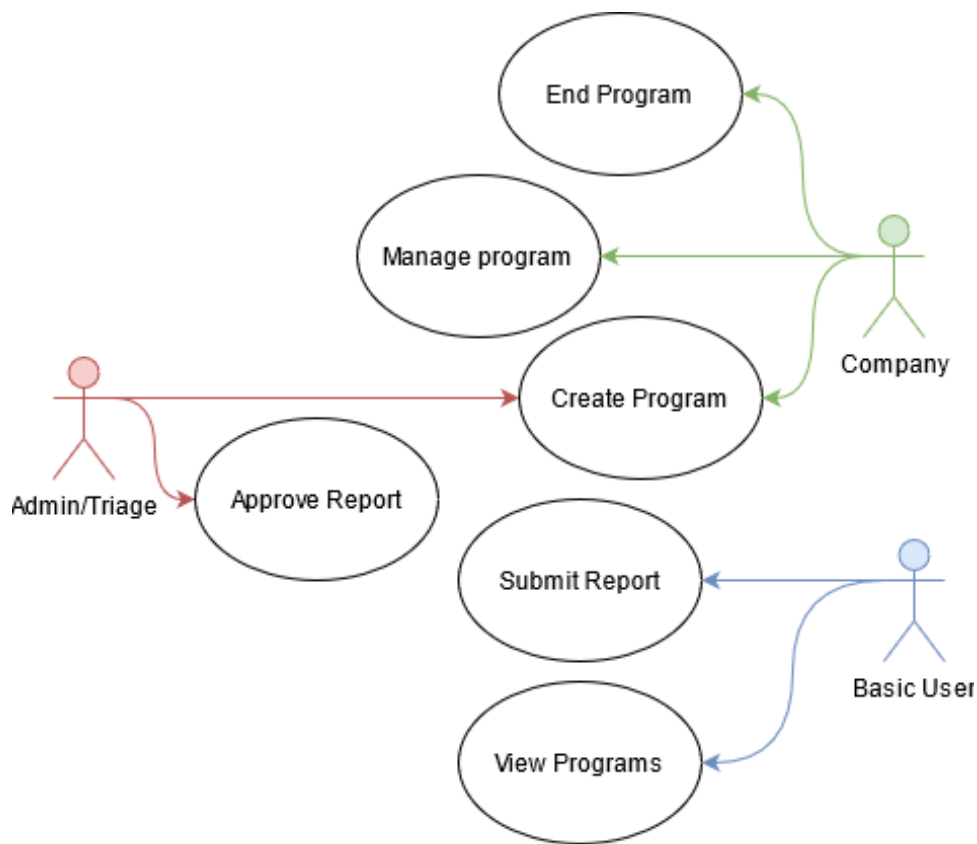


Figure 4.1: Overview of actions possible by actors

reports are valid and certify companies. All user roles can only be attained by logging into the system with proper authentication information.

4.2.1 Assumptions

It is assumed that all aspects of this project can work together in the way the designer is expecting, as real-life examples already exist. The details of integration with other tools, or parts added to increase the competitiveness of the platform are expected to be functional, but the implementation is left to the developers.

The bug bounty website must be user-friendly and all parts that don't require human interaction as automated as possible. The aim is to simplify the user's interaction as much as possible. Administrators should not be required to do anything besides certifying companies, validating programs and checking reports. Without logging in, the user will only have the ability to view public programs, but won't be able to report, demand rewards, create programs or do

any of admin related work. The logged user should be able to submit a report to accessible programs and continue communication with a company or triage team regarding the already submitted report. The company can create a new program or interact with an existing program, as well as with received reports of findings. A set of external tools possible for integration should be available, to allow companies to integrate tools they are using within the platform and for received reports to simplify the workflow. Alternatively, the platform should expose API calls to provide bug related information in a structured format such as JSON, to allow the company to consume it by tools of their choice.

Bug hunting would be described as a request from an owner of the target to find security flaws in it. The most reduced idea revolves round concepts following.

Companies can create, manage and close programs. All programs before being published must be checked by an admin/triage user to avoid unauthorised programs publication. Companies have exclusive control over the program, so any information may be changed at any time with the possibility of admin approval required and with notification to users. Unfilled information in program should be enriched by the standard terms set up by the platform, to ensure the safety of the users and companies and to avoid legal issues.

Basic users can view public programs, and private programs they have been invited to by the company and apply to a non public program which requirements they fulfil. Basic users can submit a report for programs. A report is a description of a discovered bug. If a program was created with a report checking option, the report is first sent to the triage team, which has to check if the report is valid and follows the program rules. If the triage team approves it, it is forwarded to the company to fix the bug. For programs not managed by the triage team, the report is forwarded directly to the company.

The admin's scope is to validate the creation of new programs and to check reports sent to checked programs. They should as well support the companies with creation of Vulnerability disclosure policies.

4.3 Interactions

Registered companies can create new programs. Programs are key blocks, they represent a request from the company to the users to find any security flaws and solve the legal approval for testing. When creating a program, several fields need to be filled in to provide the necessary detail for the user. If everything is made as requested anyone from the admin/triage team can authorise it and therefore publish the program.

For a company to be considered registered, it has to create a company type account and request validation. Validation can be provided by a member

4. PRODUCT DESIGN

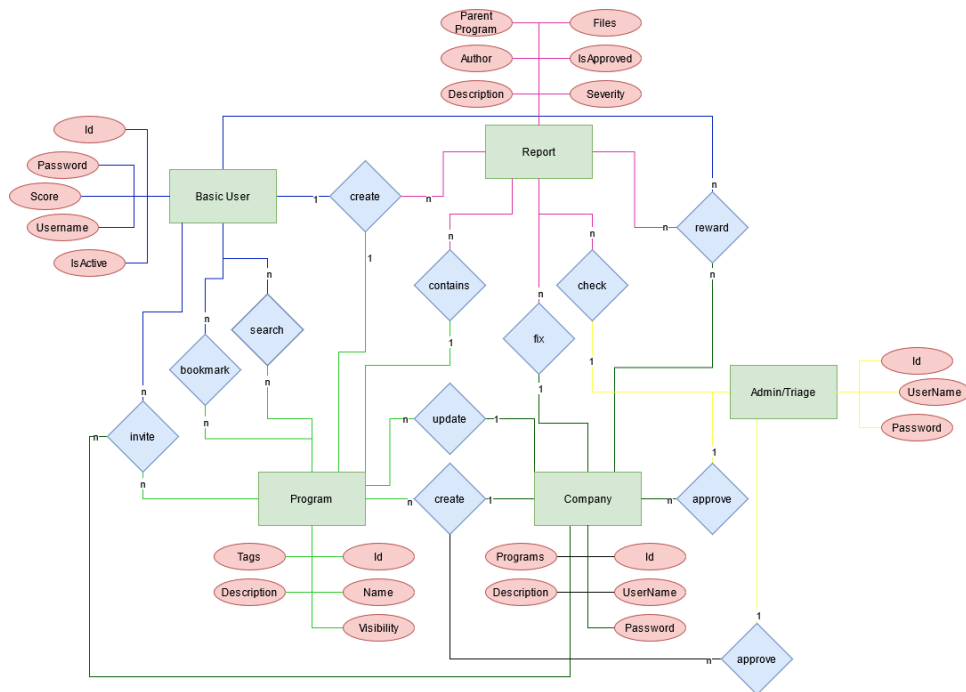


Figure 4.2: Relationships and Entities with attributes

of the admin team. It should be evaluated based on information provided on the register to validate the company and authorisation of the user to create a bug bounty program. This should ensure that all companies are genuine and the testing follows the laws.

Users can search for programs but will be shown only programs they have access to. If a user has enough permission to view a program, he will be able to see the program's information, but still cannot view other users' reports. Users can report only to programs that they can contribute to. By default, that would be only public programs. To view or submit to private or closed campaigns the user must be invited by the company that created the program. Closed programs can be viewed by anyone, but to be able to contribute, the user has to request contributor access, that can be granted by the company.

Users can also bookmark programs they have access to. Doing so will add the program to a watchlist and whenever the program is updated the user is notified. No max limit for bookmarked programs should be endorsed.

When users find a bug in any of the programs search areas defined within a scope, they can report it. When a report is filled, it is assigned to the program it belongs to. Reports should be a brief description of the bug, that the user found, and files may be added to demonstrate the bug or provide additional information.

Companies can view all reports for their program. By default, reports are

ordered by severity excluding already closed reports, so the company does not miss any critical flaw. When the company authorises a report, the author user's statistics are updated and a low priority message is sent, at the same time a payment request is initialised. Payment is decided upon several factors stated in the program's description. Bug severity, impact, area of effect or bug type could be used to determine the reward.

If report checking has been selected when creating a program, before every report reaches the company, it passes through the triage/admin team. Admins should evaluate if the report has been filed correctly and if provided information checks out, or check for the duplicity of the bug. If not a notification is sent to the user, and an extension of provided information should be requested. If the report has been approved by an admin, it is sent directly to the company to fix it.

4.4 Elements

Elements are actors/entities that are used to preserve data. There are five main elements required, based on the previously done analysis. Three of which are representations of end-users, and two are the main building blocks around which all processes regarding the bug bounty platform revolve.

4.4.1 Basic user

Basic users are the more common form of account as it does not require any complex authentication during the creation phase. When creating a new user account nickname, email and password field are required as they are used to identify the user. Alongside these fields, others are automatically generated.

Each user is represented with a score ranging from zero to 100 defining the skills, performance and activity of the user.

- 0 – 20p. for certificates. Acknowledged certificates are CEH⁶⁴, CISA⁶⁵, CISM⁶⁶, CISSP⁶⁷ and Security+⁶⁸. These certificates count 5 points each. Any not mentioned certificates related to IT are valid 2 points up to 10 points total (i.e. you can't get all 20 points from software design certificates). The points are awarded for the valid period of the certificate.
- 0 – 10p. GitHub's account, amount of points depending on activity and contributions. Contribution to multiple projects, private or shared, once

⁶⁴<https://www.eccouncil.org/programs/certified-ethical-hacker-ceh/> - accessed on 2021-03-15

⁶⁵<https://www.isaca.org/credentialing/cisa> - accessed on 2021-03-15

⁶⁶<https://www.isaca.org/credentialing/cism> - accessed on 2021-03-15

⁶⁷<https://www.isc2.org/Certifications/CISSP> - accessed on 2021-03-15

⁶⁸<https://www.comptia.org/certifications/security> - accessed on 2021-03-15

a week is awarded by 10 points and contribution to one project monthly is the equivalent of 1 point, the scale between these thresholds should be approximately linear regarding both frequency and number of projects.

- 0 – 30p. Capture The Flag. CTF is a small private contest, where the user is given a playground environment, that reassembles real-life examples and has to find as many “flags” as possible. Flags are usually represented by recognisable String patterns. Flags can be scattered across the entire playground (i.e.: source code, database or even log files). For implementing CTF, a stand-alone project should be considered, as its implementation requires a more complex codebase and preparation.

The main purpose of the CTF is for users to have the possibility to get points even if there are few bugs in all the available programs and to provide a possibility to receive invitations to non-public programs.

- 0 – 100p. Reports are the main income of points. The more you submit, the more points you can get. Report points are calculated based on the proportion between successful and failed reports, and their severity. Every report counts by 5 points and is multiplied by a criticality as 1,2,4,7 for low, medium, high and critical respectively. For an invalid report, the points are multiplied by a -1. For a duplicate bug, the points are divided by 4. Points for the reports will last up to 3 months, to reflect the activity of the researcher.

All 100 points should be achievable by reports, as it is believed, that anyone can become a good hacker even without certificates or so. But to obtain 100 points from the report section only, one would need a nearly perfect report streak.

To facilitate the sorting process a count of all reported vulnerabilities is kept for each user individually as well as the average severity of these findings. Failed (non-valid) reports are also kept, as they should indicate how active a user is.

Users can follow/subscribe to any, by them accessible, program and add it to their list. This list is kept together inside this user entity. And is visible to companies for a decision of invitations to non-public programs.

4.4.2 Company

Companies are simpler than users, as they need to save basic authentication information (company name, email, password). But unlike users, they require more validation, as the registered company must be an existing company.

Validating, if the company is genuine is required to be processed by a human. The company needs to provide basic information: website, owner, headquarters (or address of some sort) and under which authority the company

is registered. This information should be validated by the members of the admin team.

Company accounts have two nearly tied purposes. They are used for creating or updating owned programs, and for validating reports if the program is not managed by the triage team. The update of scope or creation of the program is again validated by the admin team, to ensure, that the company owns the content of the scope.

4.4.3 Triage team

The triage/admin team consists of users responsible for the smooth workflow of the site. They are the only way how to create and validate a new company account. The same goes for creating new programs. End-users cannot create admin accounts, as admin accounts should be created via direct contact with the server. Admins should be internally divided into groups assigned to specific tasks like checking reports, allowing programs creation or verifying company accounts.

4.4.4 Programs

Programs are the basic building block of the bug bounty platform. Programs are demands made by the registered companies to discover vulnerabilities in the selected assets. Each program needs to provide basic information and a description of tested software, to facilitate this process when creating a program a template is used. The template contains several text areas that need to be filled in. Other optional fields where companies can provide any additional information, that does not fit in any of the required fields, can be added.

Template fields are:

Policy – A brief description of the given program (what it is supposed to do). Ground rules on how deeply you can examine the software should be stated in this section.

Scope – A List of all subcategories of the target where users can search for bugs. Each element should be associated with a priority level, indicating how promptly a given category should be treated.

Reward – Map of expected bugs (by impact, by type) and their respective range of reward. There should be four supported levels of impact: Low, Medium, High and Critical.

Any additional sections mentioned earlier in this document may be added to provide users with the necessary information, such as out of scope targets or reporting of bugs not matching the constraints.

In addition to the description document, at least two tags must be associated with each program. Tags are used for limiting searches to specific categories of programs. Tags are not case-sensitive and should be created

by the triage team to better categorise the companies and their respective programs, to deliver them to the experts in the relevant field.

To allow for private or closed programs, visibility can be chosen when creating a program. Three visibility options are possible: Public, Closed, Private.

Public – Default behaviour of programs, all users can see these programs and the registered users can report bugs.

Closed – The program is listed with public programs, but to view its entire content (or to submit reports) a join request has to be processed by the author company. The user can apply to this program if all the constraints applied for joining are matched.

Private – These programs are not listed anywhere, and only people that have accepted an invitation can see them. For a user to be able to enter the private program, the author company must send an invitation. The invite can be provided based on the points and are of interests to the user.

While aiming to ease the report validating process, and so the companies don't have to dedicate their people to do so, a managed program option can be selected. If so, all reports submitted in that program pass through the triage team, which tests whether the reports are correct, if it is the report is forwarded to the company. The triage team should for managed programs decouple the communication between the company and users and take care of invitations for private programs as well.

4.4.5 Report

Reports are objects tied to a specific program. Reports are used to notify the company that created a program of existing bugs in their scope. They can be created by users and edited/forwarded by admins. Only key information is kept, as there will be many reports for each program. As well, the program should have a dropdown of scope related to the program and a list of known vulnerability codes.

Automatically filled data, that can not be modified by the user.

Parent program - Reference to the program the report belongs to.

Author - Reference to the author of the report. This will be used when evaluating score and processing payments.

Sections filled by the reporting user.

Description - The key component, contains a subsection with all information about the bug (where it occurred, how to reproduce it).

Scope - What scope of the program this report is about. Multiple values may be entered if the bug spans over more scopes.

Severity - A field that notates how severe is the impact of the bug. This is the default value for sorting responses. The severity can be corrected by the company or the triage team.

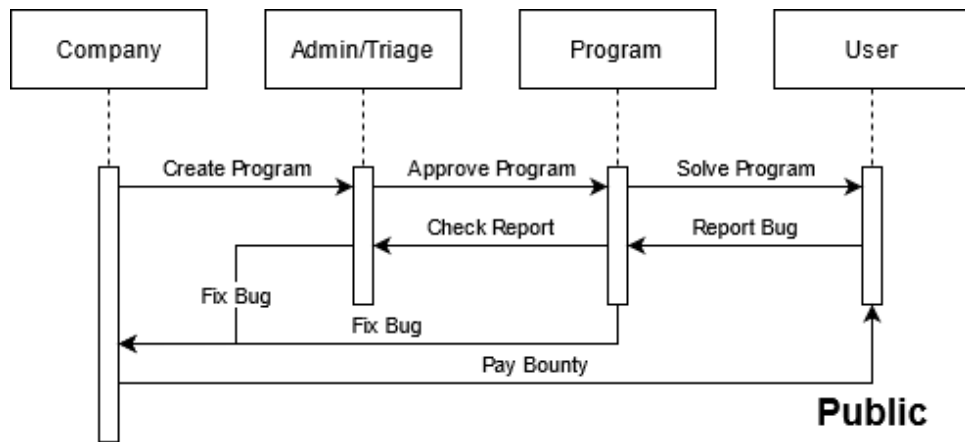


Figure 4.3: General workflow for Public Managed programs

Optionally the report may contain attached files, IP addresses used for the testing to simplify the investigation for the company and a section with suggested fixes.

Triage team comments - A value that only reports belonging to managed programs. Its value is dependent on whether the admin responsible for the report checking has approved this report or additional information gathers from the user during communication with the user. The reports not forwarded to the company should in this section have an explanation for such a reason and the company can check these reports as well.

4.5 Workflow

There are three workflows expected by the design. Even though the general flow is the same, some extra steps are involved in private and closed programs. A company creates a new program, fills in the required information, chooses visibility and report checking. Then the program must be approved by an admin. If so it is published. Users with sufficient authentication can view this program and try to find bugs in the defined scope. If a bug is found, a report is filed.

Depending on whether the option to manage the program was selected during the program's creation or not, the report can be sent directly to the company to evaluate it. Otherwise, it's first sent to the triage team that checks if the bug can be reproduced and is genuine. If a report is successfully controlled, it is forwarded to the company, that has to process a reward payment. Otherwise, if the report is not found valid, the user is notified of the failed report and the company does not receive anything.

In private campaigns, before the user can submit reports or even view the program itself, he must be invited by the company directly. The company will

4. PRODUCT DESIGN

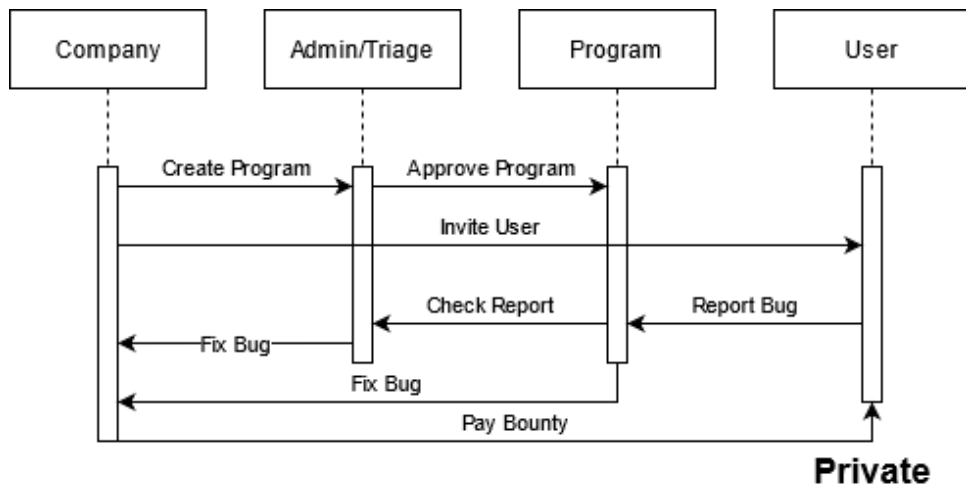


Figure 4.4: General workflow for Private Managed programs

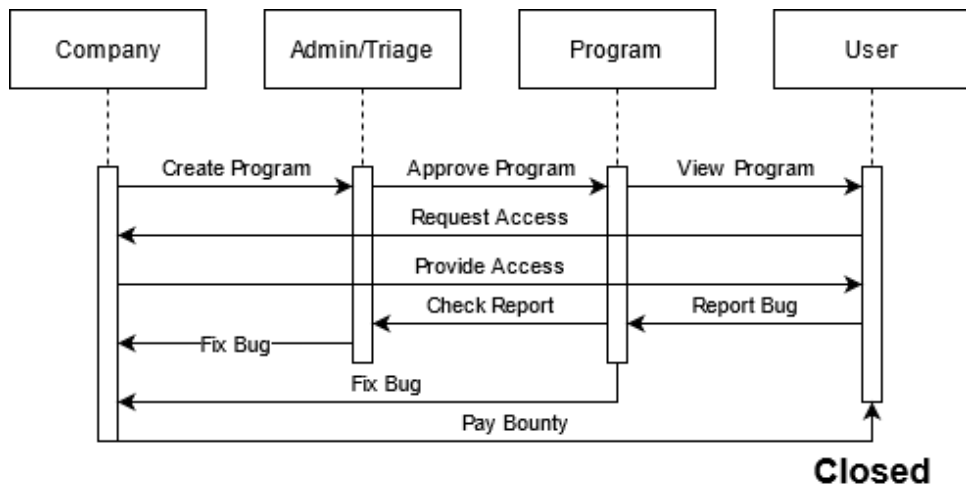


Figure 4.5: General workflow for Closed Managed programs

have a list of users at its disposal and could sort them by tags and points.

In closed campaigns, the user can view its description, but to be able to contribute to it, the user must request access to it. Access can be granted by the company, or for managed programs by the triage team.

The difference between private and closed campaigns is the actor who sends the invite/request. In private campaigns companies ask users if they want to join. As opposed to closed where the users request access from the maintainer.

4.6 Future steps

The sections in the last chapter provided a high-level design of the bug bounty platform. As the implementation details may differ for anyone using this document even at the high-level the design ends at this level of abstraction. The following points provide an overview of general and even more detailed steps, how to make the most out of this document, by anyone, who is using it as a reference for the design of the bug bounty platform.

- Define the usage of the platform.
 - Choose types of programs to be used on the platform.
 - Make a list of other functionalities to be bounded with the platform.
- Extend the provided design with modules to match your requirements.
- Select the technology used for the platform.
 - As front-end, any framework to create a clear interface should be selected based on the developer's skills.
 - Selection of back-end engine should be compatible with the selected database and based on the skills of developers.
 - For the database a non-SQL database is recommended, as the data format may vary significantly.
- The design of the front-end must remain clear and understandable to the public.
- Include your bug bounty platform as one of the programs.

The following list provides objectives to consider including, or ensuring to achieve, to be competitive and reasonable on the current market. The market requirements may develop during the time, therefore a reasonable analysis of the market is always recommended extending and correcting the following list.

- Provide integration to other tools.
- Provide API to allow custom integrations by users.
- Provide sections to educate or inform users in related fields.
- Supports the competitiveness of users to keep them motivated.
- Ensure that new companies get a helping hand with onboarding.
 - Correct setup of the program.
 - Reasonable establishment of fixing the bugs.

4. PRODUCT DESIGN

- Ensure high quality of the triage team.
 - Good communication skills with companies and users.
 - Limitation of false-positive and duplicated findings.
 - Assistance with bug fixing.
- Provide the possibility of other IT security testing.
- Assign users with a rating based on the triage team/company as another metric invitations to non-public programs.

Conclusion

I have conducted, that bug bounty hunting provides a unique way of testing the security of the target concerning the closest techniques, which are penetration testing and red teaming. I have deduced that the Software as a Service for bug bounty is rising in popularity. Based on the general discussion in total with ten users and eight companies using bug bounty platforms I have understood the positive points and negatives of the aspects of the bug bounty platforms. I have proposed a high-level design of a bug bounty platform based on the discussion and my personal experience with four different platforms.

The first chapter provided an overview of the history of the bug bounty. It as well summarises and collects a list of common sections presented in the bug bounty platforms, that is hard to find in other documents. An overview of the possible types of bug bounty programs such as public, private, time-bounded and unlimited programs. As well as the self-hosted option and using bug bounty as a service completes the summary of bug bounty possibilities. The chapter was ended with the benefits of the bug bounty and a summary to provide a comprehensive description of the bug bounty missing in other documents, articles, or books.

The second chapter provided a list of tools(vulnerability disclosure program, vulnerability scanning and management, penetration testing, red and purple teaming and security auditing) used in IT security with a brief explanation and each with comparison to the bug bounty. By the listed tools, the Vulnerability disclosure is a part of a bug bounty procedure, other than that the most similar processes are penetration testing and red teaming. It is important to say, that bug bounty is evolving quickly and bug bounty may become similar to other tools as well.

The next chapter introduced four different providers of the bug bounty platform. I had a personal experience with every single of these platforms and questioned several users and companies for their feedback regarding the usability advantages and drawbacks of the platforms. The most important part of bug bounty based on this chapter is the triage team and from the

perspective of the companies the assistance during the establishment of bug bounty, control and filtering of the reports. For the users, it is important to have a clear and easy to navigate clear webpage. As a last important part, is the possibility to have a chance to join even closed programs by proving the knowledge other than finding bugs in a public program.

The last chapter summarises the gathered information from previous chapters and proposes a high-level design of a bug bounty platform. The chapter ends with suggested steps to be taken and considered, to complete the design creation of a competitive product based on the current market situation. The design includes features to match the common requirements of the bug bounty platform. Besides, the design provides a guideline, on how to implement the main feedback obtained from users in the previous chapter. This is composed of rating, that the user can obtain by certificates, git contribution and completing challenges of the Capture the flag. The final lists provide suggestions on how to be competitive in the current market and stand out of the crowd by exposing API to be consumed by users for example.

Bibliography

- [1] Rice, A. Get a bug if you find a bug. <https://twitter.com/senorarroz/status/783093421204393985/photo/1>, accessed on 2021-02-28.
- [2] Mattermost. Mattermost bug bounty program. <https://hackerone.com/mattermost>, accessed on 2021-03-20.
- [3] Stationr, P. Play Station bug bounty program. <https://hackerone.com/playstation?type=team>, accessed on 2021-02-28.
- [4] Aboukir, Y. Bug bounty logos. <https://image.slidesharecdn.com/bugbountyprograms-171120131328/95/bug-bounty-programs-7-638.jpg?cb=1511183698>, accessed on 2021-01-28.
- [5] Cremen, L. Cycle of vulnerability management process. <https://www.datasecc.com/wp-content/uploads/2019/09/Vulnerability-Management.png>, accessed on 2021-02-15.
- [6] DataSecc. Phases of penetration testing. https://miro.medium.com/max/2400/0*dKYJh1XCm9HjXiSH.jpg, accessed on 2021-02-15.
- [7] Jani, H. Comparison of red, blue and purple team. <https://hackernoon.com/images/JTw2M3rQabaxNg3EFoNIxjmC1ZB3-6han3z1b.jpg>, accessed on 2021-02-15.
- [8] Carlisle, B. Internal Auditing Service. <https://blaircarlisle.com/wp-content/uploads/2019/10/auditing.png>, accessed on 2021-02-15.
- [9] HackerOne. HackerOne logo. <https://www.hackerone.com/assets/images/logo.png>, accessed on 2021-03-01.
- [10] Bugcrowd. Bugcrowd logo. <https://www.bugcrowd.com/wp-content/uploads/2019/04/bugcrowd-logo.svg>, accessed on 2021-03-01.

BIBLIOGRAPHY

- [11] Intigriti. Intigriti logo. <https://www.intigriti.com/assets/img/intigriti-kumkn.webp>, accessed on 2021-03-01.
- [12] YesWeHack. YesWeHack logo. <https://gsec.hitb.org/sg2019/wp-content/uploads/sites/6/2019/07/YesWeHack-Logo.png>, accessed on 2021-03-01.
- [13] Karina, E.; Suárez, G. Vulnerability Management Expert System. *Universitat Politècnica de Catalunya*, Jan 2011, accessed in 2021-02-20. Available from: https://upcommons.upc.edu/bitstream/handle/2099.1/12372/Master_Thesis-Eliana_Gonzalez.pdf?sequence=2&isAllowed=y
- [14] Ring, T. *Why Bug Hunters Are Coming in from the Wild*. ComputerFraud & Security, 2014.
- [15] Levy, S. *Hackers: Heroes of the Computer Revolution: 25th Anniversary Edition*. Novel Audio, 2015.
- [16] Jukka, R.; Luca, A. A Bug Bounty Perspective on the Disclosure of Web Vulnerabilities. *Novel Audio*, May 2018, accessed in 2021-03-09. Available from: https://weis2018.econinfosec.org/wp-content/uploads/sites/5/2018/05/WEIS_2018_paper_33.pdf
- [17] Yaworski, P. *Real-World Bug Hounting*. No Strach Press, 2019, ISBN 978-1593278618.
- [18] Netscape. All there is to know about bug bounty programs. <https://orange cyberdefense.com/uk/blog/cybersecurity/get-a-bug-if-you-find-a-bug/>, Nov 2020, accessed on 2021-02-20.
- [19] Netscape. NETSCAPE ANNOUNCES NETSCAPE BUGS BOUNTY WITH RELEASE OF NETSCAPE NAVIGATOR 2.0 BETA. <https://web.archive.org/web/19970501041756/www101.netscape.com/newsref/pr/newsrelease48.html>, Oct 1995, accessed on 2021-02-20.
- [20] iDEFENSE. The iDEFENSE Vulnerability Contributor Program. <https://web.archive.org/web/20020812035333/www.idefense.com/contributor.html>, Aug 2002, accessed on 2021-02-27.
- [21] Mozilla. Mozilla Security Bugs Bounty Program Launched. <http://www.mozillazine.org/talkback.html?article=5121>, Aug 2004, accessed on 2021-02-28.
- [22] Friis-Jensen, E. The History of Bug Bounty Programs. *Cobalt*, Apr 2014, accessed on 2021-02-10. Available from: <https://cobalt.io/blog/the-history-of-bug-bounty-programs>

-
- [23] Goodin, D. Now there's a bug bounty program for the whole Internet. *arstechnica*, Nov 2013, accessed in 2021-02-10. Available from: <https://arstechnica.com/information-technology/2013/11/now-theres-a-bug-bounty-program-for-the-whole-internet/>
- [24] Unit, C. A Framework for a Vulnerability Disclosure Program for On-line Systems. *U.S Department of Justice*, Jul 2017, accessed in 2021-03-12. Available from: <https://www.justice.gov/criminal-ccips/page/file/983996/download>
- [25] Zhao, M.; Laszka, A.; et al. Devising Effective Policies for Bug-Bounty Platforms and Security Vulnerability Discovery. *Journal of Information Policy*, Jan 2017, accessed in 2021-02-15. Available from: https://www.jstor.org/stable/10.5325/jinfopoli.7.2017.0372#metadata_info_tab_contents
- [26] Allsopp, W. *Advanced Penetration Testing*. John Wiley & Sons, Inc., 2018, ISBN 978-1119367680.
- [27] Chatfield, A. T.; Reddick, C. G. *Cybersecurity Innovation in Government: A Case Study of U.S. Pentagon's Vulnerability Reward Program*. International Conference on Digital Government Research, 2017, ISBN 9781450353175.
- [28] Burrell, B. YOU'VE GOT MAIL! – RECEIVING BUGCROWD PRIVATE PROGRAM INVITES. *Bugcrowd*, Dec 2020, accessed in 2021-03-10. Available from: <https://www.bugcrowd.com/blog/bugcrowd-private-invites/>
- [29] Pubal, J. Bug Bounty Programs:Enterprise Implementation. *SANS Institute*, Dec 2017, accessed in 2021-03-05. Available from: <https://www.sans.org/reading-room/whitepapers/application/bug-bounty-programs-enterprise-implementation-38250>
- [30] Fryer, H.; Simperl, E. *Web Science Challenges in Researching BugBounties*. WebSci, 2017, ISBN 9781450348966.
- [31] Gorenc, B. 15 YEARS OF THE ZERO DAY INITIATIVE. <https://www.thezdi.com/blog/2020/8/19/15-years-of-the-zero-day-initiative>, Aug 2020, accessed on 2021-02-27.
- [32] Abma, J. HACKERONE'S APPROACH TO TRIAGE. <https://www.hackerone.com/blog/HackerOne-Approach-to-Triage>, May 2017, accessed on 2021-02-25.
- [33] Perlroth, N. HackerOne Connects Hackers With Companies, and Hopes for a Win-Win. *The New York Times*, Jun 2015, accessed in 2021-02-21.

- Available from: <https://www.nytimes.com/2015/06/08/technology/hackerone-connects-hackers-with-companies-and-hopes-for-a-win-win.html>
- [34] Newman, L. H. The Pentagon Opened Up to Hackers—And Fixed Thousands of Bugs. *WIRED*, Oct 2017, accessed in 2021-02-22. Available from: <https://www.wired.com/story/hack-the-pentagon-bug-bounty-results/>
- [35] Rice, A.; Brown, M. L.; et al. An Invitation to Hack- The Benefits and Risks of Vulnerability Disclosure Programs. <https://www.hackerone.com/resources/wistia-webinars/an-invitation-to-hack-the-benefits-and-risks-of-vulnerability-disclosure-programs>, May 2018, accessed on 2021-02-28.
- [36] Finifter, M.; Akhawe, D.; et al. *An Empirical Study of Vulnerability Reward Programs*. USENIXSecurity Symposium, 2013, ISBN 978-1-931971-03-4.
- [37] SEC. Vulnerability Disclosure Policy. <https://www.sec.gov/vulnerability-disclosure-policy>, Mar 2021, accessed on 2021-03-15.
- [38] Zhao, M.; Grossklags, J.; et al. An Exploratory Study of White-Hat Behaviors in a Web Vulnerability Disclosure Program. *Scottsdale research institute*, Nov 2014, accessed in 2021-02-19. Available from: <https://sites.psu.edu/mingyi/wp-content/uploads/sites/11890/2014/04/wooyun.pdf>
- [39] Underwood, R. Impact of Network Security Vulnerabilities Management. *Student East Carolina University*, Mar 2016, accessed in 2021-02-15. Available from: https://www.researchgate.net/profile/Robert-Underwood-2/publication/303856184_Impact_of_Network_Security_Vulnerabilities_Management/links/5758497f08ae414b8e3f5749/Impact-of-Network-Security-Vulnerabilities-Management.pdf
- [40] Magnusson, A. *Practical Vulnerability Management*. No Starch Press, 2020.
- [41] Council, P. T. G. S. I. G. S. S. Information Supplement:Penetration Testing Guidance. *Security Standards Council*, Sep 2017, accessed in 2021-03-15. Available from: https://www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf
- [42] Bacudio, A. G.; Yua, X.; et al. An Overview of Penetration Testing. *International Journal of Network Security & Its Applications*, Nov 2011, accessed in 2021-03-09. Available from: <https://>

https://www.researchgate.net/publication/274174058_An_Overview_of_Penetration_Testing/fulltext/5737e04108aea45ee83db7dc/An-Overview-of-Penetration-Testing.pdf

- [43] Ollam, D. You're Probably Not Red Teaming... And Usually I'm Not, Either. <https://www.youtube.com/watch?v=mj2iSdBw4-0>, Jun 2018, accessed on 2021-02-12.
- [44] Fraser, G. Tunneling, Pivoting, and WebApplication PenetrationTesting. *SANS Institute*, Aug 2015, accessed in 2021-02-21. Available from: <https://www.sans.org/reading-room/whitepapers/testing/tunneling-pivoting-web-application-penetration-testing-36117>
- [45] Kim, P. *The Hacker Playbook 3 ReadTeam Edition*. Secure Planet LLC, 2018, ISBN 978-1980901754.
- [46] Shah, S.; Gianarakis, M. Catch Me If You Can. <https://www.youtube.com/watch?v=C85Z0Jgufuw>, Mar 2019, accessed on 2021-02-28.
- [47] Alleyne, N. *Learning By Practicing - Hack & Detect: Leveraging the Cyber Kill Chain for Practical Hacking and its Detection via Network Forensics*. Independently published, 2018, ISBN 978-1731254450.
- [48] Dimov, D. US Regions Most Vulnerable to a Cyber Attack. *Infosec resources house*, Aug 2019, accessed in 2021-03-05. Available from: <https://resources.infosecinstitute.com/topic/us-regions-vulnerable-cyber-attack/>
- [49] Bryant, T. *PTFM: Purple Team Field Manual Paperback*. Independently published, 2020, ISBN 979-8682974061.
- [50] Hajdarevic, K. *Cyber Security Audit in Business Environments*. International Burch University, 2018, ISBN 978-9958-834-64-6.
- [51] Moeller, R. R. *IT Audit, Control, and Security*. Wiley; 2nd edition, 2010.

Acronyms

VDP	Vulnerability disclosure policy
URL	Uniform Resource Locator
CV	Curriculum vitae
SaaS	Software-as-a-Service
SDL	Software Development Life Cycle
AI	Artificial intelligence
PSA	security certification scheme for Internet of Things hardware, software and devices
PCI	Payment Card Industry Data Security Standard
SOC	security operations center
GUI	graphical user interface
VPN	virtual private network
IP	Internet Protocol address
CVSS	Common Vulnerability Scoring System
CVS	Concurrent Versions Software
NVD	National Vulnerability Database
API	Application Programming Interface
JSON	JavaScript Object Notation
CEH	Certified Ethical Hacker

A. ACRONYMS

CISM Certified Information Security Manager

CISSP Certified Information Systems Security Professional

CISA Certified Information Systems Auditor

CTF Capture The Flag

non-SQL not only Structured Query Language

Contents of CD

	readme.txt	the file with CD contents description
	src	the directory of source codes
	thesis	the directory of L ^A T _E X source codes of the thesis
	figures	the thesis figures directory
	*.tex	the L ^A T _E X source code files of the thesis
	text	the thesis text directory
	thesis.pdf	the Diploma thesis in PDF format
	thesis.ps	the Diploma thesis in PS format