Supporting Data-driven Software Development Life-cycles with Bug Bounty Programmes

Thomas J. Walshe

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Abstract

A growing number of organisations are utilising the skills of a global base of white-hat hackers in order to identify pre- and post-deployment vulnerabilities. Despite the widespread adoption of bug bounty programmes, there remain many uncertainties regarding the efficacy of this relatively novel security activity, especially when considering their adoption alongside existing software development lifecycles. This dissertation explores how bug bounty programmes can be used to support data-driven software development lifecycles. To achieve this outcome, the dissertation presents four distinct contributions.

The first contribution concerns the usage of Crowdsourced Vulnerability Discovery (CVD) (of which bug bounty programmes are a part) within organisations. This includes the presentation of expert opinion pertaining to the benefits and shortcomings of existing approaches, and identification of the extent to which CVD programmes are used in software development lifecycles.

The second contribution explores the benefits and drawbacks of hosting a programme on a bug bounty platform (a centralised repository of programmes operated by a third party). Empirical analysis of operating characteristics helps address concerns around the long-term viability of programme operation, and allows for a comparison to be made between the cost of expanding a security team and the cost of running a programme.

The third contribution examines the extent to which participating in the search for vulnerabilities is a viable long-term strategy for hackers based on bug bounty platforms. The results demonstrate that participation is infeasible, even on a short-term basis, for significant numbers of hackers, highlighting the shortcomings of the current approach used by platforms.

Building on the first three, the fourth contribution explores CVD programme policies, and the extent to which pertinent information, particularly in reference to legal constraints, is communicated to hackers. A systematic review reveals the commonplace elements that form current policy documents, enabling organisations to identify gaps within their own programme policies and form policies that are consistent with peers.

Acknowledgements

To those that have helped keep the DPhil on track, I offer my thanks.

This goes, first and foremost, to Andy. Thank you for offering your support and guidance throughout the unpredictable years. And, for allowing me the opportunity to explore this fascinating area of cybersecurity.

Second, to my parents. Although for a time during this journey it felt like I was stuck at the end of the beginning, your constant encouragement has enabled me to reach where I am now with the DPhil, the beginning of the end. For this, and for your indefatigable patience, I thank you.

Finally, to my friends and colleagues. Thank you for being part of the journey and sharing in the joyous moments along the way. To Dennis and Mark, despite rarely being in the office, your advice as office mates has been invaluable. And to Ben, Henry, Andy, Ryan, Jon, Grigore, Charlie, and especially Lucy, thank you for a memorable four years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In an increasingly connected world, the threats to individuals and organisations are global, and continue to grow each year [29, 225]. With the sophistication and frequency of attacks increasing, continual investment must be made into defences, placing a heavy burden on organisations to protect themselves, their employees, and users [258, 89]. Failure to defend assets and private user data can lead to record fines, criminal investigations, and an irrecoverable loss of trust and reputation [18, 30, 21]. There exist myriad practices that an organisation can adopt in order to improve security [163]. However, in a budget-constrained situation it is impossible to enact all defence measures and achieve perfect security [243]. Organisations must therefore carefully allocate resources to those areas in which they believe the most benefit will be gained [235]. Little success has been found trying to measure the benefits brought about by particular security practices, leaving managers with only anecdotal evidence upon which to base decisions [35, 251]. As a result, sub-optimum allocation of resources is common, and the number of security breaches continues to increase [234].

It is common for organisations to focus their security investment on addressing issues that arise post-deployment [112, 179]. While this approach may reduce risks in the short-term, it does not address the long-term need for cost-effective security solutions. Previous work has demonstrated the need to invest in security practices throughout the software development

life-cycle [201]. Investing early to prevent the occurrence of a vulnerability not only improves the final security of an asset, but also prohibits the need for costly post-deployment fixes [113]. Crowdsourced Vulnerability Discovery (CVD) has emerged as a viable security activity for many organisations, and utilises the skills of a global community of white-hat hackers (hereafter referred to as hackers) that are willing to search for bugs in an organisation's assets, often without the expectation of compensation, helping to improve pre- and post-deployment security [195].

Despite the rapid and widespread adoption of CVD programmes over recent years, there remain many uncertainties and problems faced by programme operators and participants alike [255, 75, 271]. For many organisations looking to launch a programme, particularly a bug bounty programme, the burden on resources (especially just after launch) is not well understood or effectively communicated, leaving many operators overwhelmed, over budget, and forced to cut back the programme [9]. Furthermore, existing guidelines accessible to operators fail to provide data-driven insights or advice that may be used to improve the productivity of a programme, instead recommendations rely on anecdotal evidence or opaque methodologies. As such, the one-size-fits-all approach may not be suitable for many organisations. Finally, the greatest benefit from CVD programmes is achieved via integration alongside an organisation's Secure Software Development Lifecycle (SDLC), enabling a programme to benefit security at all stages of development [166]. There exists little literature offering guidance as to how CVD programmes can be integrated alongside a SDLC, nor many examples of organisations currently following such an approach.

1.2 Research Questions and Contributions

Motivated by the issues outlined in the previous section, the overarching research question explored throughout this dissertation is:

How can bug bounty programmes support data-driven software development lifecycles?

1.2.1 Subsidiary research questions

In order to sensibly address the overarching question, it is necessary to decompose it into a set of subsidiary research questions that cover aspects affecting a organisation's bug bounty programme. The following research questions will be answered throughout the remaining chapters:

- RQ1: *How effectively are organisations utilising their CVD programmes?*

  Analysis of existing CVD programmes is required to better understand how organisations make use of their programmes, and allows for the challenges faced by organisations and programme operators to be identified. A mixed methods approach of quantitative data-driven analysis and qualitative surveys and interviews allows for the organisational perspective to be understood. For this research question the scope is not limited only to bug bounty programmes; instead both bug bounty and responsible disclosure programmes are investigated in order to explore the utilisation of white-hat hackers in a broader context. Although closely related, the nuances of each programme lead to different challenges being faced.

- RQ2: *Is it beneficial for a company to make use of bug bounty programmes, instead of hiring additional software engineers?*

  In order to better understand the current operating environment of bug bounty programmes, it is desirable to attempt to quantify the running costs and benefits received through programme operation. To deepen the insights provided, this research question also considers the factors that drive variations in costs and benefits between programmes, and moves beyond a 'one-size fits all' approach.

- RQ3: *How effective are bug bounty platforms at retaining the interest of hackers?*

  For those organisations choosing to make use of third-party hosting sites, bug bounty platforms play a pivotal role in the success of a CVD programme. Aside from the additional services and support offered by a platform, organisations are reliant on the successes of the constituent hackers in order to derive benefit from programme operation. It is therefore prudent to understand the behaviour of hackers on a platform, with an

ambition to uncover the extent to which platforms are able to retain the interest of hackers. If platforms are failing to retain hackers, particularly those that are skilled, it may put into question the role they play in the operation of an effective CVD programme.

- RQ4: *What information do organisations convey to hackers through public policy documents?*

  For organisations employing CVD programmes, it is desirable for both operators and participants to set out clear and comprehensive 'rules of engagement' (to borrow terminology from Laszka et al. [145]). In doing so, legal, economic, and technical considerations may be understood by both parties, aiding in the smooth operation of the security activity. A greater understanding of the current norms, and identification of the shortcomings regarding key policy components, has the potential to help organisations better construct policies for the benefit of the operators and hackers alike.

1.2.2 Contributions

Through answering the aforementioned research questions, four novel contributions are made:

- C1: *Survey of the current state of bug bounty and responsible disclosure programmes.*

  In order to assist organisations through the production of extended guidelines, it is first required to understand the current state of CVD programmes, both in the real-world and through existing literature. The first contribution of this dissertation is to survey the current state of CVD programmes through the use of mixed methodological studies and reviewing relevant literature. The results of interviews and surveys will help to answer RQ1, and serve as a background for the remaining research questions.

- C2: *Quantification of the costs and benefits to operating a bug bounty programme.*

  Through exploring cost-benefit data collected from present and former bug bounty programmes, the second contribution aims to quantify many aspects of programme operation and investigate the factors that may cause variations between programmes. The results of the empirical analysis address RQ2.

- C3: *Development of models to aid understanding of platform dynamics.*

  Helping to answer RQ3, the third contribution of the dissertation provides models to aid programme operators, allowing for a better understanding of the dynamics of a given programme in the context of the behaviour of hackers across a platform. It is hoped that by understanding the viability of participation for the hackers that reside on a platform, the expected returns from a programme can be better quantified.

- C4: *Development of models to aid understanding of programme policies.*

  The final contribution, addressing RQ4, proposes models that allow for programme policies to be analysed, and weaknesses identified. By better understanding the components of a policy in the context of industry peers and existing norms, it is hoped that operators are afforded an opportunity to improve upon their own policies.

1.3 Publications

The aforementioned research questions and contributions are addressed the following publications that accompany the dissertation:

- P1: Walshe, T. and Simpson, A., 2022, December. Coordinated Vulnerability Disclosure programme effectiveness: Issues and recommendations. In Computers & Security (p. 102936) [254].

- P2: Walshe, T. and Simpson, A., 2020, February. An empirical study of bug bounty programs. In 2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF) (pp. 35-44). IEEE [255].

- P3: Walshe, T. and Simpson, A., 2022, April. A longitudinal study of hacker behaviour. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (pp. 1465-1474) [256].

- P4: Walshe, T. and Simpson, A., 2023 Towards a greater understanding of Coordinated Vulnerability Disclosure policy documents. Digital Threats: Research and Practice.

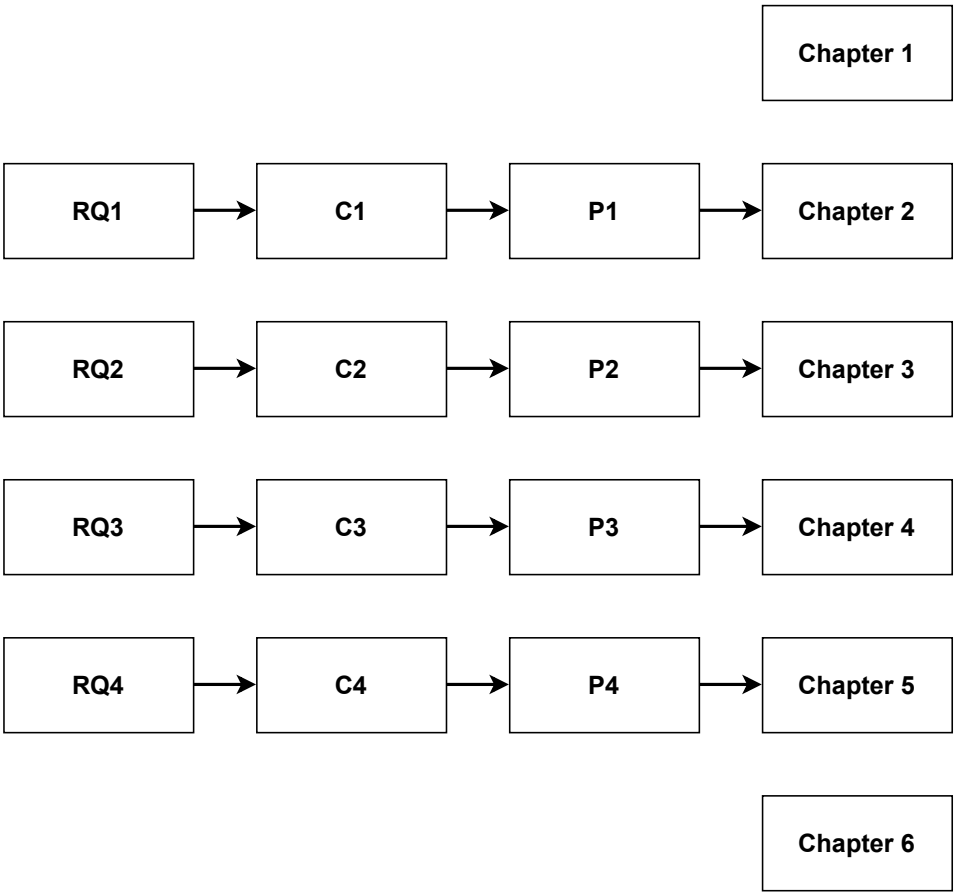

Figure 1.1: Mapping between research questions, contributions, publications, and dissertation chapters.

1.4 Dissertation Structure

A diagrammatic representation of the links between each subsidiary research question, contribution, publication, and dissertation chapter is shown in Figure 1.1. As each chapter naturally follows on from the previous, and is somewhat methodologically self-contained, discussions of pertinent background information, and a presentation of the relevant research methodology, are found at the start of each chapter.

Chapter 2 considers the issues faced by the operators of CVD programmes, and uses the results from qualitative interviews and surveys to draw comparisons between the operating environments of responsible disclosure programmes versus that of bug bounty programmes.

Complementary to the previous chapter, the operating conditions of bug bounty pro-

grammes are examined from a data-driven perspective in Chapter 3. Of particular interest is an evaluation of the cost to operate a programme and the return that one can expect for a given level of investment.

Chapter 4 investigates the behaviours of the hackers that utilise bug bounty platforms, with an ambition to identify the real-world working patterns of hackers.

Chapter 5 provides analysis of programme policy documents, allowing for the key components to be identified, and recommendations to the operators of new programmes to be produced.

Finally, Chapter 6 concludes the dissertation by summarising the key contributions and findings of the research. The underlying assumptions and limitations of the research approaches presented within the dissertation are discussed, along with a consideration of potential areas of future work.

Chapter 2

Organisational Perspectives

The underlying organisational perspectives concerning the operation of CVD programmes are explored within this chapter. Despite the widespread usage of CVD programmes and the increasing maturity of the security activity, there are still many barriers to adoption, including: large volumes of low quality reports [9], lack of hacker motivation [256], high operating costs [255], and a general distrust of hackers [81]. Understanding current perspectives has the potential to uncover further difficulties faced by operators during the pre- and post-launch phases, and assess how effectively organisations are utilising their programmes. As such, within this chapter we consider the research question *How effectively are organisations utilising their CVD programmes?*

In Section 2.1, general background information on CVD programmes is presented, with later chapters covering the specifics of bug bounty programmes and platforms in greater depth. While CVD programmes continue to be an increasingly popular security activity for organisations to employ in order to address the need for greater pre- and post-deployment security, they are not without their shortcomings. The need to identify existing issues, and effective countermeasures, is emphasised alongside other pertinent research questions in Section 2.2. Surveys and interviews with key individuals involved in the management or operation of a CVD programme have the capability to provide insights that may answer the aforementioned questions. This process is described in Section 2.3. Throughout the following sections, respondents to the qualitative study are designated using [Px], where x is an integer between 1 and 39, and

corresponds to the answers for a given individual. A brief description of the background of the respondents is found in Section 2.4. Prior work conducted by Al-Banna et al. [9] is compared and contrasted to the qualitative results. In particular the issues and fears faced by programme operators surrounding the launch of a CVD programme are explored in Section 2.5. This is followed by a discussion as to the extent that SDLCs are being influenced by the findings of a programme in Section 2.6. In order to provide benefit to real-world organisations, it is perhaps most useful to identify actionable recommendations that can be employed to help mitigate some of the most prevalent issues that plague the operators of CVD programmes. As such, a series of recommendations is provided in Section 2.7. Finally, the chapter concludes with a summary in Section 2.8. The work presented in this chapter is based on [254].

2.1 Background to CVD and SDLCs

2.1.1 CVD within secure development lifecycles

Secure Software Development Lifecycles (SSDLs) are an extension to the tradition Software Development Life Cycle (SDLC) development paradigm, seeking to better integrate security activities and checkpoints throughout the entire product lifecycle [123, 27]. By considering security at all phases (e.g. during planning, development, and deployment), the usage of an SSDL framework is intended to prevent the presence of architectural flaws and software and hardware vulnerabilities in the deployed product [20]. Aside from the obvious benefit to security, the mitigation of errors during earlier stages in the product lifecycle has been shown to decrease the total cost of development and future maintenance [193].

Created in 2008, and first published in 2009, the Building Security in Maturity Model (BSIMM) presents a data-driven approach to SSDL framework creation [223]. The BSIMM has grown from measuring the security activities of nine organisations (BSIMM1, 2009) to using measurements from 128 major organisations (BSIMM12, 2021) [221]. Its current iteration (BSIMM12) presents a collection of 122 curated security activities across the domains of: governance, intelligence, SSDL touchpoints, and development. Each domain is decomposed into a set of practices, which are, in turn, decomposed into individual security activities.

There are three mature activities of particular interest to us within the Configuration Management and Vulnerability Management (CMVM) practice. First, CMVM 3.4 advocates for the operation of a bug bounty programme. The activity was first identified in BSIMM-V (2013) but not measured until BSIMM6 (2014) [222]. It has seen steady growth within the BSIMM: from being present in 4% of measured organisations in BSIMM6 to 15.6% of organisations in BSIMM12 [221]. Second, the new addition of CMVM 3.7 in the BSIMM promotes the streamlining of incoming responsible vulnerability disclosure reports, and encourages organisations to improve the ease-of-access of reporting information to hackers (such as visibility of security email addresses and security.txt documentation [192]). Third, CMVM 3.2 encourages a systematic refinement of the SSDL framework using the feedback from operations teams. While this security activity has been included in the BSIMM since inception, it has seemingly failed to gain popularity within organisations [221]. Within the context of this chapter, this activity is of particular interest. In order to fully utilise the information contained within vulnerability reports (and any metadata), it would be prudent for an organisation to analyse reporting patterns and make necessary changes to the SSDL / SDLC processes to prevent re-occurrence. The inclusion of CVD programmes within data-driven SSDL frameworks, together with recommendations relating to the use of data in a 'feedback-to-SSDL step', demonstrate the growing acceptance and utilisation of CVD programmes within modern organisations.

2.1.2 Bug bounty and responsible disclosure

We define a CVD programme as the broad security activity that involves the disclosure of vulnerability information from an external white-hat hacker (individuals that 'positively' impact security through the disclosure of vulnerabilities [209]) to an organisation [119]. Bug bounty and vulnerability disclosure (often called responsible disclosure) programmes can be considered as two commonly employed types of CVD programme. For the purpose of this work we define a BBP as a CVD programme that explicitly offers monetary rewards — often outlined in the programme policy — for the submission of eligible vulnerability reports. In contrast, we define a Vulnerability Disclosure Programme (VDP) as a CVD programme that does not explicitly offer monetary rewards — and nor are such rewards implied. However, intangible rewards (e.g.

listing in a Hall of Fame, reputation points, etc.) and 'swag' may still be rewarded to successful hackers. Furthermore, both types of programme may offer one-off monetary rewards (not defined by programme policy) at the discretion of the programme manager for particularly insightful or severe reports.

As highlighted by Formosa et al. [83] and Manjikian [160], BBPs may be ethically questionable if the operators simultaneously encourage participation but fail to provide adequate authorisation for the activities. In an effort to provide some legal assurances to hackers, organisations may choose to publish 'Safe Harbor' guidelines that permit 'good faith' vulnerability research. A 'Full Safe Harbor' (see the Disclose.io project for templates [66]) is achieved if explicit authorisation is given to hackers conducting research an organisation's assets, and may help alleviate the risks that arise due to certain anti-hacking or anti-circumvention laws, (e.g. the Computer Fraud and Abuse Act (CFAA)) [72]. Further discussion of the legal risks and implications of bug bounties can be found in [70]. This is further discussed in Section 2.4.4.

Many CVD programmes are hosted on bug bounty platforms, such as HackerOne[1] and Bugcrowd[2], and, despite the name, bug bounty platforms often allow organisations to host a VDP. These platforms act as centralised directories for programmes, allowing user-bases of hundreds-of-thousands of hackers to view pertinent information about an organisation's CVD policies and the extent to which hackers have participated in the past. As Trusted Third Parties (TTPs), platforms help to overcome many of the obstacles that may be inherent in transactions involving the exchange of vulnerability information [170]. For example, acting as a centralised directory, platforms reduce the search costs for buyers (organisations operating programmes) and sellers (hackers looking to disclose a vulnerability for a reward). The inclusion of 'gamified' metrics such as leaderboards, reputation points (awarded for valid reports), and 'halls of fame' are thought to motivate hackers to search for vulnerabilities [183].

It is often advantageous for an organisation to host on a platform. The centralisation allows for vast numbers of hackers to see the programme and participate. It was suggested by Raymond that it was beneficial to have many individuals searching for vulnerabilities: "Given

[1] `https://www.hackerone.com`
[2] `https://www.bugcrowd.com`

enough eyeballs, all bugs are shallow" [195]. However, many large organisations (particularly technology companies, such as Google and Apple) are able to leverage their reputations and offer large payouts to attract hackers to stand-alone programmes hosted on their own domains. Furthermore, some small organisations, particularly those operating a VDP, are unable to justify the additional costs that may come with using a platform.

2.2 Motivating factors

2.2.1 Qualitative studies surrounding CVD

There have, in recent years, been numerous qualitative studies exploring the backgrounds and the relationships between hackers, security professionals, and organisations.

Using survey data collected between 2011 and 2013, Hafiz and Fang [107] leveraged 127 responses to analyse the approaches taken by individuals to discover vulnerabilities in three prominent vulnerability classes: buffer overflows, SQL injection, and Cross-Site Scripting (XSS). Survey respondents included both security professionals and hackers, with individuals looking for vulnerabilities either as a hobby or as a professional responsibility. Interestingly, a majority of respondents were reported to favour the full-disclosure of vulnerability details — either due to a dissatisfaction with the communication between themselves and a vendor, or with the hope that the increased visibility will prompt a timely response by a vendor. However, the authors dismiss this (at the time) widely held belief, showing instead that a large proportion of fully-disclosed vulnerabilities remain unfixed. Instead, it is recommended that reports should be disclosed to vendors to get an early fix [23]. Hafiz and Fang [107] suggest that the large number of possible channels for disclosure inhibit the ability of an organisation to find and fix a vulnerability. With no single centralised repository, reports are instead posted to personal Twitter pages, personal GitHub repositories [252], mailing lists [156], public databases [236], and disclosure sites [271]. The fear of full-disclosure is discussed further by Al-Banna et al. [9] and in Section 2.6.2.

A study by Votipka et al. [252] presents the results from 25 semi-structured interviews in order to compare and contrast the experiences between internal security professionals ('testers')

and hackers. Similarly to Hafiz and Fang, the authors consider how individuals select areas to search, make use of automated tools, and communicate with other stakeholders. Witschey et al. [261] also consider the adoption and usage of security tools amongst developers. The contributions of Votipka et al. [252] and Hafiz and Fang [107] both identify groups of hackers that search for vulnerabilities as a hobby. Respondents explain that this is done for personal enjoyment, but also as a good opportunity to learn using real-world examples [252].

In HackerOne's 2019 report [99], learning is cited as one of the key non-monetary motivating factors for hackers on the platform. This conflicts with the views of Malladi and Subramanian [158] that monetary rewards act as the sole source of motivation for hackers to search for vulnerabilities. Votipka et al. [252] suggest that despite hackers and testers following similar processes to find and report vulnerabilities, the outcomes are significantly different due to imbalances in experience and depths of security knowledge. They argue that the expertise of hackers is greater than that of testers, and, as such, it is recommended that organisations improve their relationships with hackers in order to maximise the long-term benefits that can be attained through vulnerability disclosure.

An earlier study by Al-Banna et al. [8] used 66 survey responses and 32 interviews to understand how the expertise of a hacker is indicated to other hackers and security professionals. The authors highlight the value to an organisation that can be found through the invitation of hackers to private programmes, such as those hosted on third-party platforms (e.g. HackerOne), through the identification of expertise indicators. Public programmes do not allow for an organisation to restrict access to low-skilled hackers. As such, without the ability to filter by expertise, a programme may receive an overwhelming majority of invalid reports ('noise'). Supply shocks due to the COVID-19 pandemic may have exacerbated such problems [275]. Automated approaches to report validity checking, such the report classifiers by Fan et al. [74], or by extending the topic models proposed by Xia [264], may be beneficial to organisations looking to filter out valid reports from the incoming noise. However, practitioners may be wary of automated report management systems in practice due to reliability [274]. A study by Li and Zhao [151] explores how operators utilise governance mechanisms (such as definitions of in-scope assets) to control the quantity of submissions to a programme.

Smith et al. [210] explore the use and effectiveness of internal red teams (offensive security) at Microsoft using the results of 17 interviews with red team members. Unlike hackers, internal red teams may utilise insider knowledge and shortcuts (such as internal network access) to more quickly identify vulnerabilities or consider more advanced attack scenarios [115]. Alomar et al. [15] also use interviews to explore vulnerability discovery and management from the perspectives of internal testing teams, penetration testers, red teams, blue teams (defensive security) and purple teams (offensive and defensive), and through the employment of bug bounty programmes. The authors highlight the need for a mature security culture, effective internal communication channels, and acceptance by managers before the aforementioned security teams can be fully utilised.

Using the results of 36 interviews conducted between 2015 and 2016, Al-Banna et al. [9] investigated the organisation perspective on the usage of CVD programmes. The authors uncovered the fears that many organisations express towards the use of CVD programmes. Of the interviewees, 10 were actively involved in the operation of a CVD programme and provide insights as to whether these fears are realised in practice. In addition, ongoing issues during programme operation, as well as countermeasures, are presented. Many of the fears and issues presented in the study are echoed in other qualitative and quantitative work. A distrust of hackers by some organisations is also found by Tanczer [224].

2.2.2 Objectives

The focus of the work described in this chapter is, in part, inspired by the creation and usage of a Bug Bounty Lifecycle (BBLC) framework by Verizon Media [166]. The ambition of the operator is to maximise the value derived from the BBP using a lifecycle approach that incorporates the findings from hackers to help guide organisational change in respect to security efforts and activities [166]. Although there have been many studies that consider aspects surrounding the usage of CVD programmes by organisations, there has been rather less investigation into the extent to which organisations use the results of a CVD programme to influence change in their SDLCs.

Furthermore, the continued adoption of CVD programmes gives motivation to explore how

organisational perspectives have changed over time. Motivated by the work of Al-Banna et al. [9] (who conducted interviews between 2015 and 2016), it is useful to explore how the pre-launch fears and post-launch issues have changed as the usage of CVD programmes has matured in recent years. It is also useful to assess the recommendations of Alomar et al. [15] in the context of BBPs and VDPs.

It is hoped that, by identifying how organisations use the findings from their programmes to bring about change, as well as how they overcome fears and issues that arise, it will inspire other organisations to take onboard the recommendations put forward in this chapter and more effectively utilise their CVD programmes.

This chapter seeks to address the research question *How effectively are organisations utilising CVD programmes?* through the following set of subsidiary questions:

1. What are the current issues and fears faced by programme operators pre- and post-launch?

2. To what extent do organisations use the findings from their CVD programme to direct change throughout their SDLC?

3. How should organisations looking to leverage the work of hackers best set up a CVD programme?

Question 1 seeks to re-examine the findings of Al-Banna et al. [9] using new data collected up to five years after their original study. If the same fears and issues persist, it perhaps demonstrates the existence of systemic flaws in the current methods to promote and inform organisations about the usage of CVD programmes.

The aim of Question 2 is to identify the extent to which organisations are currently using the findings from hackers to influence changes within the organisation, helping to improve security. Furthermore, by identifying which activities are being influenced by the findings, it is hoped that more organisations may be inspired to reflect upon their own processes.

In conjunction with the recommendations put forward by Alomar et al. [15], Question 3 considers how best an organisation should set up a new CVD programme. In contrast to the

earlier contribution of Alomar et al. [15], the work presented in this chapter also considers the use of VDPs in addition to BBPs.

2.3 Qualitative methodology

2.3.1 Candidate selection

Organisations operating BBPs and VDPs were considered, as both are categorised as subtypes of CVD programmes (see Section 2.1). Organisations were selected based on their inclusion on a bug bounty platform (HackerOne [105], Bugcrowd [44], iNTiGRiTi [121], Cobalt [53], and YogOsha [265]), listed in the Disclose.io database [66], included in the *Awesome Bug Bounty* GitHub repository [124], listed on FireBounty [77], mentioned on the Google Play Security Reward Program (GPSRP) [16], or identified from any additional public mentions of a CVD programme on a company website.

Many of the bug bounty platforms provide the names of organisations operating a VDP, as well as those operating a BBP. This resulted in 950 unique organisations with a corresponding contact email address; manual searching and matching was required in many instances to identify a suitable line of contact for organisations listed without an email address. A large number of candidate organisations were selected to help the generalisability of the results, but also to mitigate the poor response rates that are typical with web-based surveys that recruit via email [204, 159].

2.3.2 Survey design and distribution

Based on the recommendations found in popular survey design literature [202, 152, 139, 76], a typical iterative design process was used to construct and design questions (and groups of questions) in order to improve readability, ensure good narrative flow, and avoid bias. Respondents were presented with a total of 24 open-ended and closed-form questions; branching was employed to prevent respondents from being presented with questions that would have been irrelevant to their organisation. Four of the questions related to obtaining consent from the respondents and the option to participate in a follow-up interview. Considerations were made

to ensure that the total time to complete the survey did not exceed the approximate duration of 15–20 minutes, as reducing the burden on the participants can help improve completion rates and may also impact the quality of the collected open-ended data [88].

Prior to distribution, care was taken to ensure that the study complied with commonplace professional and ethical codes of conduct, such as the ACM Code of Ethics [3]. A study by Buchanan and Hvizdak [40] found that issues around consent, confidentiality, security and data storage (among others) were of particular importance to web-based surveys and survey tools. As such, additional refinement to the survey was made to minimise any personal information collected about a participant. To comply with departmental requirements, the Jisc Online Surveys platform[3] was used throughout. Ethics approval was sought from, and granted by, the University of Oxford's Computer Science Departmental Research Ethics Committee (CS-DREC)[4].

The list of 950 organisations was split randomly into 10 sub-lists. Smaller batches of emails were used to ease the burden of processing submissions and to allow for the invitation email to be refined between batches. After the initial batch, additional clarification with respect to the anonymization of survey responses was included in the invitation email.

Organisations were contacted during February–October 2021. From the 950 organisations contacted, 278 individuals accessed the survey site, leading to 39 complete responses. Individuals were not compensated for either the survey or follow-up interviews. On average, approximately 6% of emails sent were undeliverable to the target organisation, suggesting that some of the information contained on public bounty lists may be outdated.

2.3.3 Interview process

The final section of the survey allowed participants to express an interest in participating in a follow-up interview and give consent to being contacted. For those willing to be interviewed, a contact email was collected, and the option to provide their name, organisation, and position was given, From the 39 survey respondents, 14 expressed an interest in the follow-up interview,

[3] `https://www.onlinesurveys.ac.uk`

[4] Approval number CS_C1A_20_029 and approval date 9th December 2020

leading to 8 interviews being conducted. (Unfortunately, 6 individuals failed to respond to interview invitations.)

Semi-structured interviews, lasting 27 minutes on average, were held on the Microsoft Teams platform shortly after the participant had completed the survey. The shortest interview lasted approximately 18 minutes and the longest lasted 40 minutes. Information provided in the survey response and public information about the organisation's CVD programme was used to guide the discussions. Following the completion of an interview, the audio was transcribed by the author.

2.3.4 Data analysis

The approach utilised for the data analysis was twofold. First, the responses to all closed-form questions were separated from the survey data. Many of these questions were followed up by open-form questions, as suggested by Schuman and Presser [207]. However, on their own, the categorical data were analysed to produce insights into sentiment and programme structure. Despite the use of semantically ordinal scales (strongly agree, agree, neither agree nor disagree, ...) for the reporting of a respondent's sentiment towards a question, previous studies have highlighted the suitability of scales for use in parametric statistical analysis [130, 61]. As such, statistics from the Likert scales are presented where meaningful [216]. A 'sentiment' score is computed using the arithmetic mean from the Likert scales following numerical assignment (e.g. strongly agree = 2, agree = 1, neither agree nor disagree = 0, ...).

Second, the survey data and interview transcripts were analysed using the thematic analysis technique described by Braun and Clarke [38]. Their qualitative techniques are widely used in relevant computer science literature, particularly when applied to the results of surveys and interviews [9, 210, 224, 127, 71]. As such, their approach was deemed to be appropriate. The process is outlined as follows. First, all relevant data is centralised and read over, and initial observations are noted. Then, interesting features within the data are systematically coded and the relevant data is extracted and grouped. Care is taken to preserve the context of the data by including surrounding sentences in the extract [39]. The codes (and associated data) are then iteratively grouped and refined to ensure that the identified themes are consistent with a

Type	Survey	Interview
BBP	12	3
Both	10	2
VDP	16	3
Other	1	0
All	39	8

Table 2.1: Type of programme operated by each survey and interview participant.

given group of codes, and across the entire corpus. The themes resulting from the application of this process are used to help address the research questions presented in Section 2.2.

2.4 Background statistics

Following the qualitative data collection period, as described in Section 2.3, the responses for 39 organisations were collected in conjunction with a further 8 follow-up interviews. The background information provided by each participant is presented within this section.

2.4.1 Type of programme

Table 2.1 shows the type of programme operated by the participants of the survey and interviews. Respondents operated Bug Bounty Programmes (BBP), multiple programmes consisting of at least a bug bounty and vulnerability disclosure programme (Both) and Vulnerability Disclosure Programmes (VDP), with one respondent operating a very informal programme that they, at that stage, classified as belonging to neither sub-type of CVD (Other).

2.4.2 Platform usage

For those operating a bug bounty programme (either BBP or Both as per Table 2.1), respondents were asked to provide details of whether they made use of a third-party bug bounty platform, and, if so, to what extent they made use of the available services.

Although some platforms allow for VDP programmes to be hosted, VDPs are not considered within this section due to the high variability between platforms as to the functionality provided to those listing VDP programmes. As shown in Table 2.2, approximately 55% of respondents made use of a bug bounty platform, with Bugcrowd (8) being the most popular platform,

Type	Used platform	Didn't use platform
BBP	7	5
Both	5	5
All	12	10

Table 2.2: Usage of third-party bug bounty platforms for participants operating a bug bounty programme.

Type	Fully-managed	Semi-managed	Self-managed
BBP	1	2	4
Both	2	2	1
All	3	4	5

Table 2.3: Bug bounty programme management style for those hosted on third-party platform.

followed by HackerOne (5) and Synack (1). Two respondents used two different platforms simultaneously.

Those operating a bug bounty programme on a platform have the option to let the platform operators manage part (semi-managed) or all (fully-managed) of the vulnerability reporting process, often for a substantial fee. Common features and services that may be included as part of the management agreement include: assistance during programme creation (e.g. setting scopes, rewards, and disclosure policies), tools to invite specific hackers to a programme (either background checked or specific background skills), front-line report triage and validation, remediation advice, and patch verification [104, 43]. Alternatively, a programme can be hosted on a platform, but run independently without the use of the available services, although there may still be a hosting fee. Shown in Table 2.3 is a breakdown of the management style for those operating a bug bounty programme that is hosted on a platform.

2.4.3 Rewards

In many cases, hackers are rewarded by organisations following the submission of a valid vulnerability report. This may be in the form of monetary compensation (often linked to the severity of the vulnerability identified) or company branded merchandise ('swag'). Shown in Table 2.4 is a summary of whether the respondents offered any form of reward to successful hackers. Those respondents operating a bug bounty programme (BBP and Both) always offer some form of reward, but just over half of those exclusively operating a VDP offer rewards.

Type	Offer rewards	No rewards
BBP	12	0
Both	10	0
VDP	9	7
Other	0	1
All	31	8

Table 2.4: Propensity to offer any form of reward to hackers, by programme type.

Reward	BBP	Both	VDP
Monetary payouts	11	10	3
Swag	2	2	4
Reputation points	4	3	2
Public disclosure	6	3	3
Other reward	0	1	3

Table 2.5: Type of rewards offered by respondents, by programme type.

For those organisations offering rewards to hackers, the range of rewards offered is summarised in Table 2.5. The other rewards offered include: product licenses, contracts for private penetration testing, and further public recognition of the hacker. Of the three VDPs offering monetary rewards to hackers, none mentioned the usage of bounty payouts publicly (often to avoid increased programme activity), and two of them only awarded payouts in exceptional circumstances (e.g. particularly impressed at the novelty or quality of the discovery by a hacker).

2.4.4 Safe Harbor

Safe Harbor clauses aim to provide hackers with some reassurance that they will not face legal repercussions following an attempt to discover (successfully or unsuccessfully), vulnerabilities in an organisations assets, subject to conditions (see [63] for examples where greater protection is needed). These clauses can include 'good-faith' statements [65]:

> "We will consider your security research to be authorized if you make a good faith effort to comply with this policy during your security research.
>
> ⋮
>
> For inadvertent, good-faith violations of this policy, we will not take civil action or file a report with law enforcement. Before doing anything that may be inconsistent

with or unaddressed by this policy, please contact us by submitting a report."

Alternatively, they can include more extensive 'authorisation' statements [64]:

> "When conducting vulnerability research, according to this policy, we consider this research conducted under this policy to be:
>
> - Authorized concerning any applicable anti-hacking laws, and we will not initiate or support legal action against you for accidental, good-faith violations of this policy;
> - Authorized concerning any relevant anti-circumvention laws, and we will not bring a claim against you for circumvention of technology controls;
> - Exempt from restrictions in our Terms of Service (TOS) and/or Acceptable Usage Policy (AUP) that would interfere with conducting security research, and we waive those restrictions on a limited basis; and
> - Lawful, helpful to the overall security of the Internet, and conducted in good faith. . . . "

Despite the inclusion of partial or full Safe Harbor clauses in a programme's policy, such a policy does not legally protect hackers from third-party organisations:

> "note that the Safe Harbor applies only to legal claims under the control of the organization participating in this policy, and that the policy does not bind independent third parties." [64]"

In the survey, one respondent commented negatively on the usage of a Safe Harbor in relation to its failure to protect hackers from third-party organisations: "use of the term Safe Harbor is misleading and dangerous to hackers and should be abandoned" [P20].

Within the survey, respondents were asked if they provided hackers with a Safe Harbor when conducting security research — either in the form of 'good-faith' statements (partial Safe Harbor) or in terms of explicit authorisation (full Safe Harbor). The results are shown in Table 2.6. As can be seen, a significant proportion of operators of bug bounty programmes

Type	Full	Partial	None	Unsure	No answer
BBP	5	1	1	3	1
Both	5	1	0	2	1
VDP	4	2	6	3	0
Other	0	0	1	0	0
All	14	4	8	8	2

Table 2.6: Extent to which Safe Harbor was offered by organisations.

Asset	BBP	Both	VDP	Other	All
Web	12	10	15	1	38
Desktop application	5	0	6	1	12
Mobile application	6	3	3	0	12
Source code	5	2	7	1	15
Hardware	1	1	2	0	4
IoT	0	0	1	0	1

Table 2.7: Types of in-scope assets included in respondents CVD programmes.

(BBP and Both) offer full Safe Harbor. In contrast, many of those exclusively operating a VDP provide no Safe Harbor clauses to hackers. Interestingly, 8 respondents were unsure as to the details of their organisation's Safe Harbor clause. As the first responders to incoming vulnerability reports, it may be prudent for the programme operators and triaging teams to be aware of organisational policy in order to best assess whether a hacker has violated the policy during their exploration.

2.4.5 Asset coverage

Within a programme's policy, an organisation may specify which of the assets under their ownership are defined as being in-scope for hackers to explore. Respondents were asked whether they included assets in any of six broad categories and the results are shown in Table 2.7. Somewhat unsurprisingly, web assets were most commonly covered, with all but one respondent having web assets in-scope. Section 3.10 in the following chapter further discusses the assets typically included in a CVD programme.

2.5 Issues and Fears Surrounding CVD Operation

We consider the first of the chapter subsidiary research questions (see Section 2.2) in this section. Fears expressed by participants prior to the launch of a programme are presented in Section 2.5.1. This is followed by a discussion of the issues faced by operators after the launch in Section 2.5.2. The countermeasures enacted by participants to combat the aforementioned issues are displayed in Section 2.5.3. The section concludes with an analysis of the findings in Section 2.5.4, and comparisons are drawn with reference to the earlier work by Al-Banna et al. [9].

2.5.1 Pre-launch fears

From the results of the survey and interviews, five themes relating to organisational pre-launch fears are identified: usage of hackers, distrust of hackers, communicating with hackers, volume of reports, and internal limitations.

Fear: Usage of hackers

Unlike the typical employee recruitment process that seeks to reduce the information asymmetry between an individual and an organisation concerning the trustworthiness, skill, and motivation of the individual, most CVD programmes open their gates to a global community of unknown and untrusted hackers. For many organisations these uncertainties manifest themselves in pre-launch fears. There are three sub-themes identified around the usage of hackers.

First, there is concern about the poor quality of reports submitted by hackers. Understandably, duplicate and invalid reports are of little-to-no value to an organisation. Furthermore, a lack of skill or a lack of incentive may result in the submission of shallow or low quality reports. One participant mentioned that they had a fear of a "high volume of low value and duplicate reports (this turned out to be a very valid fear)" [P3]. This sentiment is explicitly echoed by many of the participants.

The cause of poor quality reports may be, in part, due to a lack of skill or understanding by

the hacker. The second sub-theme surrounds a lack of knowledge surrounding the competency of many hackers, with [P10] stating that they had a fear about the "experience of researchers". Poor competency may cause hackers to find "stupid things that are publicly viewable that one thinks is worthy of reporting and receiving recognition" [P5].

Third, there is uncertainty as to whether an organisation "would be considered attractive to researchers" [P9]. If hackers are unmotivated to search for vulnerabilities, an organisation will see little benefit in starting a CVD programme. This sub-theme is expressed in a budgetary context in the *internal limitations* theme (Section 2.5.1).

Fear: Distrust of hackers

Despite the growing adoption of CVD programmes and widespread stories of the successes of hackers in finding vulnerabilities, there exists a distrust of hackers within some organisations. First, there is a fear that, following a negative interaction with a hacker, retribution may be taken in the form of full disclosure or through reputation damage on social media. Both [P26] ("we'd get something wrong and be dragged on social media") and [P11] ("there is also the fear that a reporter will talk badly about you on Twitter or similar") expressed this fear. As discussed in Section 2.5.2, reputation damage in the eyes of the hacker community can have adverse effects on the performance of a programme.

Second, from the perspective of some departments, the unorthodoxy of allowing external and unknown hackers to explore an organisation's assets induces fear. Three of the participants note that their legal teams raised early concerns and issued guidance prior to launch: "Our legal team raised concerns about giving access to external people" [P30]. Legal complications surrounding the usage of hackers from unknown locations, particularly as there may be a risk of communicating with individuals from sanctioned regions, led to [P30] noting that "if my legal team had to choose they wanted to restrict the geo-location of the researchers". The legal complexities associated with using hackers are further explored in Section 2.7.2.

Fear: Communicating with hackers

For organisations opening up lines of communication with hackers, there are two areas of concern: disagreement and harassment.

[P19] expressed an early fear that hackers may not agree with the re-assessment of the severity of a vulnerability: "they would find a problem that we would not feel is not severe and the hacker would disagree that the issue is severe and there would be a conflict of interests". In conjunction with the potential for reputation damage in the eyes of the hacker community, it is understandable that disagreements between programme operators and a hacker are undesirable.

Additionally, [P20] noted a "fear of dealing with harassment from impatient researchers as we got our processes in order". Within the early stages of programme operation, an organisation may not have the processes in place to respond to incoming reports in a time-frame that is preferable to all hackers. In such a scenario, "chasing by reporters" [P32] may only impede the timely operation of a programme.

Fear: Volume of reports

The uncertainty around the number of incoming reports is another fear held by organisations. Primarily, organisations fear a volume of reports that is too large to effectively deal with: "could our systems and processes cope with an influx of disclosures at the beginning?" [P9].

Fear: Internal limitations

A number of internal limitations could impede the successful operation of a new programme and induce anxiety in those involved. There are three sub-themes that relate to the fears participants held in relation to internal limitations.

First, an organisation's employees may not posses the skills required to operate a programme correctly or fix the identified vulnerabilities. Amongst the community of hackers there exists a broad range of technical expertise and proficiencies, which in some circumstances may exceed the breadth of knowledge possessed by the security engineers within an organisation: "I feared that they would find a problem that we could not fix" [P19]. The inability to fix an exposed vulnerability not only puts an organisation at further risk to unknown hackers,

but may sour communications with the original reporter, potentially leading to full disclosure. In summary, "you launch a programme but you don't have the expertise to evaluate the submission that you receive, you are going to have a bad time!" [P30].

Second, a failure to address incoming reports in a timely manner is identified as a fear. Not only do longer response times displease hackers, but the burden of responding to reports has the potential to overwhelm a team: "we also had to discuss how much time triaging reports would take" [P15].

Third, programme operators fear that their budget, or lack thereof, may inhibit the success of an upcoming programme. As noted in the *usage of hackers* theme, some organisations are uncertain whether their programme will motivate hackers to participate. Many attribute this (prior to launch) to low or no bounty payouts as a reward for successfully finding a vulnerability, with the major constraint being internal budget allocation. "Our fear, as a not-for-profit organisation, was the limited impact of a CVD programme (e.g. low quality vulnerability reports) if you do not award large monetary rewards. Which proved to be true" [P34]. Others note that their budgets would prevent them from using (often prohibitively expensive) bug bounty platforms, thus potentially limiting visibility to hackers [P30] or that a series of successful reports may result in the depletion of the budget, leaving no money available for continued operation [P9].

2.5.2 Post-launch issues

Three themes are identified when considering the issues faced by organisations post-launch: issues with reports, issues concerning the behaviour of hackers, and internal issues.

Issue: Reports

Four sub-themes are identified in connection with the vulnerability reports submitted by hackers: quality, volume, scope, and severity.

Respondents commonly cited the poor quality of incoming reports as a major issue facing their CVD programme, with many reports being: false positives (invalid), inconsequential (valid, but not useful), duplicates, incomplete, or relevant to another organisation. Aside from

the validity of the information contained within, many reports are written in bad English, fail to demonstrate the issue, or are in an unreadable file format. The use of automated scanning tools, particularly by low skilled hackers [P9], often leads to a significant volume of low quality results: "it seems like everyone who can run Burp cut-n-pastes their scanner output into our vulnerability report form" [P39]. The problem is exacerbated by numerous hackers submitting almost identical reports on duplicate issues, who "report the same reports over and over" [P19], leading to further decline in the average quality of incoming reports.

Interestingly, respondents face differing issues when it comes to the volume of reports they receive. For some organisations, high volume goes hand-in-hand with poor quality: "we get a massive amount of unhelpful reports" [P18]. Many attribute the use of scanners and other automated tools as the cause of a high volume of incoming reports and placing an undesirable load on an organisation's live environments [P9]. Conversely, organisations may experience a drought of reports, perhaps due to unmotivated hackers, diminishing the returns from the programme.

Typically, an organisation will define limits as to the types of vulnerabilities they will accept, as well as the areas that they will allow (either implicitly or explicitly) hackers to explore for valid issues. However, a commonly observed problem is hackers ignoring (or in some cases arguing against [P28]) the programme scope set out by an organisation in their CVD policy. It is believed that some individuals fail to read the policies outlined by an organisation, instead choosing to test in both in- and out-of-scope assets. "Many researchers do not read (or understand) our disclosure policy, particularly with regard to in-scope systems" [P9]. Aside from the impact on the organisation in the form of invalid reports being submitted, hackers, by exploring unauthorised areas, may be in violation of anti-hacking laws. This aspect is further explored in the *issues concerning the behaviour of hackers* theme. Significant volumes of poor quality, out-of-scope vulnerability reports are commonplace in CVD programmes, causing undue burden on operators and limiting the extent to which a programme can benefit the security of an organisation.

When submitting a vulnerability report, hackers are often asked to indicate the severity of the vulnerability in question, often through the provision of a CVSS score. The final sub-theme

concerning *issues with reports* pertains to the over-inflation of the report severity. The cause of the issue may be twofold: potential monetary rewards are directly linked to the severity of a vulnerability (misaligned incentives), or as an outsider, a hacker is unable to appropriately quantify the severity of a vulnerability in the context of an organisation's assets (information asymmetry or lack of hacker skill). As such, severity over-inflation is common, requiring reassessment by a skilled programme operator [P19].

Issue: Behaviour of hackers

There are four sub-themes identified in connection with the behaviour of hackers: communication, motivation, rewards, and undesirable behaviour.

For many organisations, the fear of poor communication with hackers has manifested as a prevalent issue during operation. Respondents reported that frequently the most 'vocal' hackers were those that had submitted an invalid report: "communication is generally poor, a lot of them refuse, don't understand that we are communicating, don't understand why what they have found is not a real issue" [P26]. As put by one operator, a "sense of entitlement" [P6] exists amongst a subset of individuals, with another stating "over and over [they] say 'are you going to pay me?', no you didn't find anything" [P26].

The success of a programme is, in part, dependent on attracting high quality researchers. Once the 'low-hanging fruit' have been reported, subsequent discoveries require the continued effort of motivated hackers. For operators, enticing hackers back to a programme is problematic: "many researchers they're really focused on low-hanging fruits" [P1]. An operator of a mature programme expressed a difficulty in motivating hackers to search for vulnerabilities in their assets as they are not trivial to test [P1]. Another respondent noted that previous reputation damage in the hacker community has dissuaded previous participants from returning [P20].

Following the submission of a valid vulnerability, it is usual for an organisation to publicly acknowledge the efforts of the hacker, often through the use of a 'Hall of Fame'. However, listing a hacker's username and link to social media or personal website is not without its problems. It may be important for the organisation to ensure that the individual they are promoting has

not been linked to, and is not associated with, any entity that is considered undesirable: "we do due diligence just to make sure that it is not, we aren't promoting someone that, you know, that is part of a community is counter to company policy or culture" [P5]. Although this issue does not overly impact the operation of a CVD programme, two participants reference this as a difficulty that is faced in their independent CVD programmes [P30, P33].

Although some of the aforementioned behaviours may also be undesirable to programme operators, particularly those relating to poor communication, several issues raised by respondents suggest that a small number of hackers participating in CVD programmes behave in a manner that violates programme and platform policies, or in some instances they carelessly perform illegal activities. Such behaviour can negatively affect those operating CVD programmes, and other hackers that participate in them. An organisation may carefully construct their policy documents to explicitly define client (customer) systems as being out-of-scope, especially if they not have ownership of the systems. One respondent ([P26]) noted that hackers have, in the past, tested against and incidentally exploited the vulnerabilities in client systems when searching for and then later reporting issues. Such behaviour is clearly extremely undesirable, and often cannot easily be prevented. Another respondent stated that, "as we do not offer bounties, we are unable to make some of them reveal their true identity and hence there is no legal recourse we may pursue in such cases" [P32]. While this serves as an example of very serious undesirable behaviour, respondents identified several other examples of potentially less serious behaviours in some hackers. Two respondents ([P20] and [P32]) had previously experienced full disclosure of vulnerabilities by hackers without their consent: "some researchers have gone public, especially when we used to use [platform] and [platform]" [20] (platform names redacted). For hackers on a bug bounty platform, such behaviour may violate platform policies, and can result in account termination [101].

A final benign example of undesirable behaviour seen by operators arises due to the intangible rewards that are offered on many bug bounty platforms. Hackers may be rewarded with non-monetary 'reputation' or 'kudos' points following the submission of a valid vulnerability report via a platform. In some instances, the submission of a duplicate report may still constitute the hacker being eligible for being awarded points. One operator ([P26]) observed a

particularly blatant approach by one hacker to submit the same vulnerability through several different accounts, with near identical usernames, in the hope of being awarded points for each duplicate submission. Although there is no direct monetary cost associated with this 'reputation farming', its existence undermines the 'value' associated with these intangible points and could lower the effectiveness of the gamification elements commonly used by platforms to motivate hackers. If organisations use such metrics as an indicator of the expertise of a hacker (say when inviting hackers to a private programme), they may be misled as to their true skill level.

Issue: Internal to organisations

Four sub-themes are identified surrounding the internal issues faced by organisations post-launch: management, burden, push-back, and lack of skilled staff.

Separation between the team responsible for managing the CVD programme and the teams responsible for developing or securing various company assets (while often necessary) causes management issues within some organisations. Several ongoing issues surrounding project management are highlighted by respondents. First, for large organisations, the process of identifying the responsible product owner, assigning the security issue, and managing communication can be a time-consuming and complex process [P3, P11, P21]. Second, while many organisations make use of ticketing systems to resolve the aforementioned issues, management and tracking can still be challenging [P34, P3].

A consequence of a high volume of incoming reports is that significant effort may be expended by operators to process and investigate issues. Furthermore, developers may be burdened by the increased number of security issues to resolve [P13]. Respondents highlighted the issue of spending large amount of time processing reports, with one stating "we spend more time weeding out the garbage from our security reports than we do fixing anything that's actually reported" [P35] and another commenting on their programme being "a massive time sink for senior engineering time" [P37].

Despite the successful discovery, triage, and internal communication of a vulnerability, several organisations report push-back from the teams responsible for a fix; a reluctance to fix

valid security issues undermines the work of the operators and diminishes the benefits gained from running a CVD programme. In particular, this appears to be an issue for vulnerabilities that may not demand the urgency of a critical severity bug. Instead, respondents noted a "lack of willingness to prioritise remediation by some product teams (substantial issue for easily discoverable issues that are then repeatedly reported)" [P3]. However, when constrained by the choice of spending time on product development versus fixing low-severity vulnerabilities [P21], it is understandable that some overstretched teams may de-prioritise security: "We have not been internally diligent about following up on low-impact reports. While this is in many cases an issue of engineering resource allocation, multi-month turnarounds lead to researcher dissatisfaction and disillusionment" [P39]. Ultimately, for any organisation with security as a top priority, resisting push-back is necessary: "we had to use diplomacy to make them understand security wasn't optional and that it was a mandatory requirement" [P13].

A common theme for many respondents concerned their admiration for the skilled hackers that possess the ability to discover complex vulnerabilities within their assets: "some researchers on these platforms that are really making being a security researcher an art" [P1]. However, the discovery of a highly technical, obscure or otherwise complex vulnerability sometimes necessitates a corresponding member within the organisation with the prerequisite knowledge or skill to triage and fix the issue.

An issue within some organisations is the absence of employees capable of tackling the demanding reports submitted by hackers. Furthermore, a lack of experience may fail to address broader issues surrounding the existence of a vulnerability: "junior engineers often can't properly assess the depth of issues or may patch a bug without addressing a systematic issue that could re-occur in other areas of code" [P37]. In one organisation, the triaging team for the CVD programme contained employees without any cybersecurity skills, potentially inhibiting the process of raising the alarm upon the submission of serious issues [P13].

2.5.3 Countermeasures

To counteract many of the fears and issues outlined earlier, a range of countermeasures may be employed. From these, there are four themes identified: internal changes, use of platforms,

policy alterations, and use of additional tools.

Countermeasure: Internal changes

As highlighted in Section 2.5.2, myriad internal issues hamper efforts to operate a CVD programme following launch. For some organisations, a change in attitude and refinement of internal processes eased the burden on the operators and developers: "[we] obtained senior management support for prioritizing resolving vulnerabilities over developing new features" [P21]. In order to stem the tide of low-severity reports, one respondent altered internal policies by "treating certain lower-severity / theoretical issues as proper vulnerabilities, mostly to avoid getting repeated reports about them" [P26]. The lack of actionable countermeasures within this theme may highlight the continued difficulty in instituting internal change to counteract the issues previously raised.

Countermeasure: Use of platforms

Bug bounty platforms offer many additional, often paid-for, services to organisations aside from hosting of programmes. Services such as the use of private invite-only programmes and the provision of dedicated front-line external triaging teams are designed to alleviate some of the fears and issues faced by organisations, but these often come with significant added cost (see Section 2.7.2).

To mitigate the drain on internal resources due to triaging and communicating with hackers, several organisations chose to make use of the 'managed programme' service offered by platforms such as HackerOne and Bugcrowd: "We have launched the program on [platform] so that this is no longer our internal ticketing triaging team responsible for triaging bug reports" [P34]. Furthermore, alongside the use of a managed programme, one respondent noted that private programmes, when operated on a platform, allow the operator to carefully invite hackers with verified identities, helping to partially resolve some of the fears of distrust and issues concerning hacker behaviour: "we accept only vetted users into our program and have full control on their accounts ... so, not only are we accepting verified researchers but we also control their access, because if we want to revoke a credential we need the power to do

it" [P30].

Countermeasure: Policy alterations

It is typical for an organisation to outline the 'rules of engagement' for hackers in a policy statement [145], which is often viewable on a programme page. Although there is much variation as to the content of the policy, it is common to include information on: in- and out-of-scope assets, conditions for report eligibility, safe-harbor or 'good faith' statements, and a description of the reporting process. Furthermore, operators may outline the potential rewards available to successful hackers.

Alterations to programme policy were the most common type of countermeasure employed by respondents, resulting in two sub-themes: reward modification and scope alteration. For the successful discovery and disclosure of a vulnerability, some organisations offer tangible rewards to hackers, with monetary payouts and 'swag' (e.g. company t-shirts) being common. Altering the programme reward policy was used by organisations to both disincentivise the submission of undesirable reports (e.g. those of low severity vulnerabilities) and provide a greater incentive for hackers to search for vulnerabilities. In an attempt to reduce the number of lower severity reports, several respondents changed their bounty structures to offer no payouts for low severity reports: "we ultimately decided to only award bounties for medium-severity issues and above" [P15]. For those not in a position to offer monetary payouts, other tangible rewards proved successful: "we are now offering some 'swag' as rewards, which has led to a higher number of useful reports" [P34].

Using a well-defined scope allowed some respondents great control of the types of reports that could be ignored or rejected during triage, thus helping to reduce the burden on the operators, with one respondent having "amended [the] scope several times to exclude low value classes of vulnerability" [P3]. By only considering worthwhile vulnerabilities in higher priority assets, some operators were able to more carefully control the running costs of the programme by limiting the number of incoming reports eligible for a bounty payout: "[we] had a pretty thorough program scope, so that cut down on payouts where we didn't think it was appropriate to reward" [P26]. However, as noted by one respondent, reducing the scope

did not markedly reduce the incoming volume of reports. Instead, the usage of standardised email responses for out-of-scope reports allowed for quicker rejections: "it allows you to be quite clear in an email, standardised email response template. You just say, it is out-of-scope, go away" [P4].

Countermeasure: Use of additional tools

Adopting new tools and using new services allowed some respondents to overcome certain operating issues. Several noted that they had developed in-house tools to better automate the triage process to help reduce the time spent processing incoming reports. One operator of a private programme described that the requirement for hackers to be authenticated while accessing assets allowed for the monitoring of hacker activity to be automated, giving an indication as to the effort being put into their bug bounty programme by hackers: "I set up a way to monitor how many researchers, more or less, are working on a weekly basis on our programme. You definitely see a trend that is going down over time" [P30].

2.5.4 Analysis

Inspired by relational visualisation between fears, issues and countermeasures by Al-Banna et al. [9], a diagram similar in concept is shown in Figure 2.1 using the themes discussed in this section. In contrast to the visualisation of Al-Banna et al. [9], it is more logical to show how fears evolve into operation issues, and how they are subsequently affected by the enacted countermeasure.

Analysis of pre-launch fears

As shown in Figure 2.1, 11 predominant fears were reported by respondents, covering the four themes described in Section 2.5.1. Comparing these to the fears previously identified by Al-Banna et al. [9] reveals that many still persist within organisations pre-launch.

The only fear that is significantly different is that of programme 'cost escalation' [9]. In their paper, Al-Banna et al. attribute the uncertainty in, and inflation of, operating costs to a requirement by the organisation to pay out to a hacker after the submission of a vulnerability.

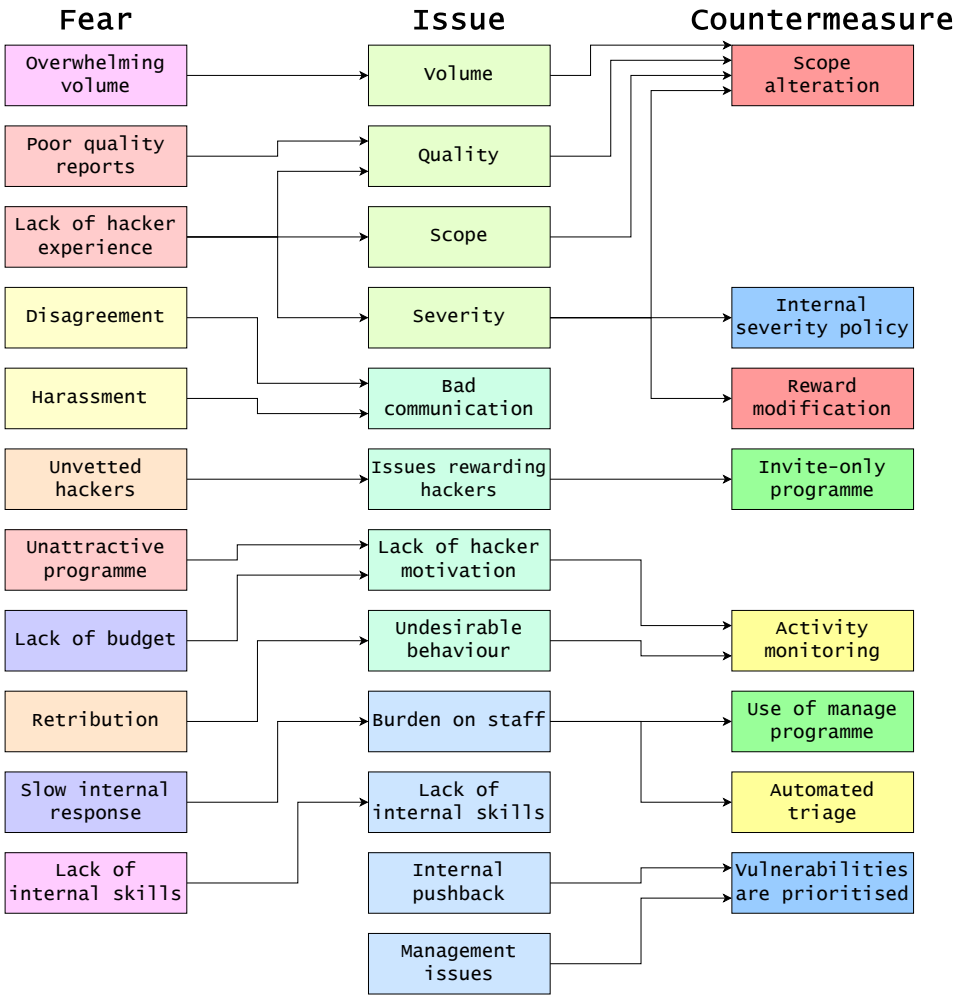

Figure 2.1: Relationship between identified fears, issues, and countermeasures. Some connections have been hidden for clarity. Colour groups within a column represent items within the same theme (no relationship across columns).

However, as demonstrated by the legal threats against hackers in [63], there are public examples of organisations failing to pay out for, or in some cases acknowledge, the reports submitted by hackers. Furthermore, should a disgruntled hacker then choose to publicly disclose the details of the vulnerability, they may be in violation of the Digital Millennium Copyright Act (DMCA) [63, 85, 241]. These examples contravene the claim that organisations have to "abide by the rules for payments" [9]. Additionally, as noted by [P18] (see Section 2.7.2), reports that could contain valid information can be rejected on the basis of minor errors by the operators on a discretionary basis. Given that operators are not legally obliged to pay out for all valid

vulnerabilities, it is perhaps why respondents focused on a lack of budget (leading to budget depletion), over unsustainable cost escalation. However, as noted by Al-Banna et al. [9], the difficulty in determining the number of vulnerabilities present (and then later discovered) in an organisation's assets before launch makes budget allocation extremely difficult.

Overall, it appears that the fears felt by operators has remained somewhat consistent over time. And, as demonstrated by the issues faced by operators (see Section 2.5.2), the fears are often valid.

Analysis of post-launch issues

In common with Al-Banna et al. [9], issues surrounding the quality of reports, lack of hacker motivation, and burden on staff are still present to those currently operating CVD programmes. Despite continued researched aimed at addressing the prevalence of low quality reports [144, 183, 5, 158], it would appear that many organisations have yet to make use of the findings. Further to the aforementioned issues, respondents also identified issues pertaining to the themes of hacker behaviour and internal issues.

Aside from the issue of motivation, the behavioural issues raised by respondents may stem from the (almost) total anonymity possessed by many hackers, leading to a debasement of the severity of potential repercussions. Within Psychology, there is a surfeit of research that considers the changes in behaviours of individuals when acting anonymously [215, 143, 203]. On bug bounty platforms, such behaviour can be punished with 'warnings', 'temporary bans', and eventually (in some cases after several offences), the 'permanent ban' of a hacker's account on the platform [101]. However, given the ease of creating a new and unblemished account, these punishments appear somewhat redundant in the face of users that place little to no value in their account (e.g. new users). As such, it may be of little surprise that issues surrounding bad behaviour persist, both on- and off-platform.

In regards to internal issues faced during operation, it is concerning, yet not surprising, to see a disconnect in the priorities of security teams, product teams, and management: the "sell first, fix later" mentality persists within some organisations [22].

Analysis of suggested countermeasures

For most of the issues raised in Section 2.5.2, respondents offered suggestions of countermeasures that helped alleviate these ongoing issues. This is shown by the connections between issues and countermeasures in Figure 2.1.

While a lack of internal skills may be counteracted by the hiring of skilled staff or selective training (see Section 2.6.2), respondents did not come to a consensus as to the best countermeasure for this issue, particularly in a resource-constrained environment.

Comparing the suggested countermeasures with those of Al-Banna et al. [9] reveals many similar approaches used by organisations. Notably, 'third-party support' and 'limiting participants' can be realised through the adoption of managed and invite-only programmes. However, as we shall discuss in Section 2.7.2, these countermeasures often come at great monetary expense, so are not applicable to many smaller organisations. The absence of other countermeasures that address the need for verifiable hacker identities, or actions that can be taken to reduce the burden on operators, exhibits a serious problem for the operators of many CVD programmes.

There is, perhaps, a need for a low-cost (in comparison to platform fees) decentralised identity verification service that allows operators a sense of safety when interacting with hackers, and in such a way that is platform agnostic. Understandably, as perhaps demonstrated by the minority of users on bug bounty platforms with verified identities, the lack of legal protections offered to hackers (discussed in depth in Chapter 5) inhibits the propensity for an individual to undergo background check or identity verification (see the Safe Harbor discussion in Section 2.4.4). Additionally, in respect to the burden on operators, although some respondents make use of tools to help automate the triage of reports, many solutions are in-house, commercial, or linked to a platform. For those without the resources to adopt such measures, there is perhaps motivation for further open-source solutions.

Although presented as countermeasures, there are several measures that may be best suited for use pre-launch, such as scope alterations, reward modifications, and prioritisation of vulnerabilities fixes. More thoroughly discussed in Section 2.7.3, it is recommended that operators

start off with a narrow scope and policies that reward only 'useful' vulnerabilities. By addressing these points prior to launch, the impact of certain issues may be curtailed. In addition, it may be prudent for operators to seek as much internal support as is reasonable before launch in order to avoid later push-back or conflicts in objectives (e.g. security versus new features).

2.5.5 Summary of fears, issues, and countermeasures

Re-investigating the fears, issues, and countermeasures used by the operators of CVD programmes reveals that little has changed since the study of Al-Banna et al. [9]. Furthermore, the use of expensive third-party services demonstrates an inflexibility in current approaches to alleviating issues surround trust and internal operational burden. Despite the recommendations of Al-Banna et al. [9] as to directions for future work, the persistence of common issues may highlight a need to better integrate the results of academic research into solutions that are not purely commercial.

2.6 Impact on the SDLC

Inspired by the extent to which Verizon Media aims to integrate the results from their CVD programme into changes in the wider SDLC framework (aptly named the BBLC [166]), respondents (N = 39) were asked to describe how they utilised report information in the context of organisational change. Results pertaining to the closed-form survey answers are presented in Section 2.6.1, and the open-ended survey and interview answers are presented in Section 2.6.2. A summary of the findings, and a discussion of the extent to which SDLCs are supported by CVD programmes, is found in Section 2.6.3.

2.6.1 Closed-form results

Impact on security practices

As shown in Table 2.8, a majority of organisations (22) have been prompted to make changes to security activities due to the results from the CVD programme. For 13 of surveyed organisations, the results of the programme did not elicit a change in practices, and a further 3 were

Programme Type	Yes	No	Unsure
BBP	9	3	0
Both	7	2	1
VDP	6	8	2
All	22	13	3

Table 2.8: Responses to "Have the results from the CVD programme prompted change to security practices?" (N = 38). The response from one organisation has been excluded as no answer was provided.

Security activity	BBP	Both	VDP	All
Training for engineers	3	2	3	8
Focus or use of attack models	2	2	0	4
Specification of security requirements	6	3	1	10
Security standards and compliance measures	3	0	2	5
Communication between internal teams	6	4	2	12
Code review methods	1	2	1	4
Security testing tools or methods	7	2	4	13
Use of internal or professional penetration testing	2	3	1	6

Table 2.9: Number of respondents that made changes to particular security activities, given that their CVD programme results prompted change (N = 38). The response from one organisation has been excluded as no answer was provided.

unsure (one participant did not provide an answer). There is a significant disparity between those exclusively operating a VDP versus those operating a BBP or both types of programme, with a half of those exclusively operating a VDP seeing little change in security practices owing to the results from the programme.

Affected security activities

Respondents provided an indication of the security activities affected by the results of the CVD programme via a closed-form question, and the option to provide further information via an open-ended question. The results from the closed-form question are shown in Table 2.9. For those organisations that made changes, the average number of activities changed is somewhat similar across programme type: BBP - 3.3 changes; Both - 2.6 changes; VDP - 2.3 changes.

Elimination of vulnerabilities

Organisations were also asked to comment on the effectiveness of their current approach to SDLCs and usage of CVD programmes in the context of vulnerability reappearance. If an effective process is in place, certain vulnerabilities may be less likely to reoccur once found and patched. Answers on a Likert scale from 'Strongly agree' to 'Strongly disagree' (see Section 2.3) are given ordinal representations (see Table 2.10). Those operating a BBP or both a BBP and a VDP expressed, on average, a slightly positive sentiment in relation to their ability to prevent vulnerabilities from reappearing (0.5 and 0.8 respectively). Conversely, the operators of VDP programmes expressed, on average, slightly negative sentiment (-0.5). The sole respondent in the 'Other' category denoted a strong negative sentiment (-2). Across all respondents, the average sentiment was very slightly positive (0.10).

Response	BBP	Both	VDP	Other	All
Strongly agree (2)	2	1	0	0	3
Agree (1)	4	6	1	0	11
Neither agree or disagree (0)	4	3	8	0	15
Disagree (-1)	2	0	5	0	7
Strongly disagree (-2)	0	0	2	1	3

Table 2.10: Responses to "Certain categories of bug once found do not reappear due to changes in development and testing." (N = 39)

2.6.2 Open-ended results

Four relevant themes are identified from the open-ended answers: improved employee training and education, usage of internal security champions, testing techniques, and organisational strategy changes.

Training and education

The ongoing arms race between increasingly sophisticated hackers and security professionals can leave developers without the requisite knowledge to address new and complex vulnerabilities that may affect their organisation (see Section 2.5.2). Furthermore, labour supply shortages of well-trained security professionals [55, 87], and a general under-confidence in the

cybersecurity-related proficiencies of new graduate developers [47, 54], result in many organisations around the world having development teams that lack the technical expertise to secure new and existing assets. It is therefore not surprising that both the closed-form (Section 2.6.1) and open-ended answers highlight the role of additional employee training and education following feedback from the CVD programme.

For some respondents, training serves as a reactive measure to mitigate the risks of repeated known attacks: "education sessions are held in response to patterns of reports or novel attacks" [P9] and "if we are starting to see a lot of vulnerabilities in one area, we may increase training on a selected training on a selected portion of the population" [P5]. One respondent described the role of internal workshops and presentations thus: "there has been some level of awareness work out of the back of the responsible disclosure programme which may help people think more about certain considerations". They went on to say: "we have training paths now, focusing on certain commonly occurring vulnerability reports, so that more of the SOC team are able to triage reports when they come in" [P3].

Security champions

Security champions are individuals within an organisation, typically in development or engineering roles, that have a strong working knowledge of security topics, and may help guide or manage security activities [220]. Respondents noted that an apparent shortfall of skills in certain teams could be partially mitigated by leveraging an internal network of security champions. This in some cases necessitated an expansion of the "security champions forums" [P4]. Strengthening and using the champions network in one organisation allowed more experienced security professionals to "jump [in] and and help validate more difficult to verify queries" during triage [P3].

Testing techniques

Automated testing techniques, if configured correctly, may help developers catch mistakes during development or configuration issues during deployment. Some respondents noted that the results of the CVD programme prompted the organisation to revise or expand the testing

methodologies currently employed: "we built some new detection for subdomain takeover problems directly as a result of reports" [P3].

Organisational strategy

Without periodic reviews, programme managers may be unable to see the prevalence of certain patterns of reports. Furthermore, if programme managers are not able to communicate the patterns they discover, systemic issues may persist, leading to a drain of resources as the CVD programme sees repeated reports on previously identified issues. These concerns prompted two respondents to better integrate the CVD programme outputs into a periodic and systematic review process: "we've got meetings that we have weekly where we discuss some of the disclosures that we are seeing to see if there is some strategy that needs to be changed. And, in some cases budget allocated for programme work (training, etc.)" [P5]. Another respondent noted that a formal review process was introduced with development teams to help effectively communicate current issues: "It's an involuntary meeting once a quarter, to look at the most recent security alerts. Here are the things that happened this quarter, it is just an educational process. Here is what is going on in the industry, bring your knowledge up to speed." [P33]

2.6.3 Summary of the usage of CVD findings in the SDLC

As shown by the answers to the closed-form (Section 2.6.1) and open-ended (Section 2.6.2) questions, a majority of the respondents noted that they had used the results from their CVD programme to help guide changes in security activities within their organisation. The use of testing tools and methodologies, alterations to internal communication processes, and specification of security requirements were the activities most often affected across all respondents.

Despite this, it is apparent from the results, particularly the interviews, that none of the respondents integrated the results of their CVD programmes to the extent of Verizon Media's BBLC. Instead, the utilisation is on a shallower level that suggests a more reactive approach to altering security activities based on the incoming reports. It is perhaps due to a lack of resources or a lack of internal integration that greater institutional change is not made. As noted in Section 2.6.2, few respondents shared details of the internal periodic information flows

of report details that were used to help guide security strategies.

In summary, although organisations use the results from their CVD programmes to direct changes in security activities, at present this appears to be somewhat shallow within the organisations included in this chapter.

2.7 Recommendations to organisations

In order to aid those looking to set up a programme, respondents were asked about their set-up process and provided advice. Results for eight closed-form questions are presented in Section 2.7.1, and the results for the open-ended questions and survey responses in Section 2.7.2. Finally, a summary of pre- and post-deployment recommendations is discussed in Section 2.7.3.

2.7.1 Closed-form results

In the survey, Questions 7–11 aimed to uncover the areas of difficulty encountered by programme operators prior to or at the start of operation. Questions 20–22 asked respondents to reflect upon their overall experience operating a CVD programme. The average sentiments, expressed across programme types, for the following closed-form questions are presented in Table 2.11. For each question, significance testing of the average sentiment is performed at the 5% level with Welch's T-test (unequal variance assumed), using the Bonferroni correction where appropriate, to compare the effects of: programme type, usage of a platform (for BBP and Both), and management style (fully- or semi-managed programmes compared to independent management on a platform). Unless significant results are obtained and presented, it can be assumed that no significant difference in the average sentiment is present at the 5% level.

Understanding the resources

Being a relatively new security concept, organisations may not have the prerequisite knowledge or skills to easily set up a CVD programme. Respondents were asked whether the process of launching a programme and the resources required were well understood. The slightly

Question or statement	Average sentiment for type BBP	Both	VDP	All
The process of launching a programme and the resources required were well understood	0.84	0.9	0.44	0.67
A well defined process was in place allowing for reports form hackers to be validated	0.49	1.4	0.44	0.67
A well defined process was in place to notify the responsible team about the presence of a vulnerability	1	1.3	0.75	0.96
Systems allowed for clear and easy communication between the hackers, the CVD team and the fixers of the bugs	0.83	1.3	0.63	0.83
There was a clear method used to set bounty levels or rewards that were appropriate for the organisation	0.34	0.9	0.31	0.43
The use of a CVD programme has benefited the security of assets	1.49	1.6	1.2	1.35
The operation of a CVD programme is a good use of resources	1.26	1.5	0.7	1.05
Would you recommend the use of a CVD programme to other organisations?	1.51	1.6	1.32	1.41

Table 2.11: Average sentiment for the closed-form survey questions, expressed across programme type (N = 38). "Other" programme type omitted due to non-response.

positive average sentiment across all respondents suggests that organisations understand (to some degree) the steps leading up to the launch of a programme (see Table 2.11).

Validation processes

Following the submission of a report, the existence and severity of a vulnerability must be validated by an individual with sufficient skill to make a suitable judgement. For organisations, this can be done either internally or using the optional services provided by a bug bounty platform. Table 2.11 presents the (overall slightly positive) sentiment of the respondents when asked if a well defined report validation process was in place prior to launch.

Although no significant differences are found at the 5% level, testing at the 10% level reveals some significant differences between the different types of programme and for those using a bug bounty platform. Using the Bonferroni correction and comparing the average sentiment between Both (1.40), BBP (0.55) and VDP (0.44) demonstrates a significant difference between those operating Both as compared to a BBP ($p = 0.0280$) and VDP ($p = 0.0168$). This may suggest that programme operators that currently operate more than one programme

are able to carry forward processes honed during earlier launches. Controlling for platform usage (considering only BBP and Both) also reveals significant differences at the 10% level. Respondents making use of a platform reported, on average, a higher positive sentiment (1.25) compared to those that did not (0.56), $p = 0.090$.

Internal notification

As discussed in Section 2.5.2, one issue faced concerns the possible difficulty in identifying and communicating with the internal (or sometimes external) party responsible for fixing the vulnerability. Despite the occurrence of this issue, respondents, on average, noted a positive sentiment when asked about the processes in place (see Table 2.11).

Communication

Further to the previous question, respondents were also asked to provide feedback on broader communication systems in use around the time of launch. Again, the average sentiment is positive (see Table 2.11).

Setting bounty levels

For all programme operators, the decision to offer any rewards can have a significant impact on the success of the CVD programme [13]. For those offering monetary rewards, bounty levels must be set in such a way to balance total expenditure with the potential to attract increased volumes of higher skilled hackers. The need to offer competitive bounties is mentioned by respondents in Section 2.5.3 and in Section 2.7.2.

As shown in Table 2.11, although the average sentiment for setting bounty levels and rewards is positive (0.51), many respondents expressed uncertainty. Comparing the sentiment of those using a managed programme (1.42) versus an independent programme (0.38) on a platform reveals a significant difference ($p = 0.0496$) at 5%, suggesting that platform managers are able to clearly convey their rationale behind setting initial reward structures. As explained by one respondent, this typically involves comparing factors such as company size, industry,

level of security maturity and budget to the reward structures of existing public and private programmes on a platform [P30].

Impact of CVD on security

Through the operation of a CVD programme, it is hoped that the discovery of security vulnerabilities will allow organisations to further secure their assets. Respondents commented on the overall effectiveness of their CVD programmes, expressing, on average, a positive sentiment (1.39), as shown in Table 2.11. Although not significant at the 5% level, comparing the average sentiment between those using managed programmes (1.86) and those running independent programmes (1.36) on a bug bounty platform shows significance above $p = 0.0555$.

Effectiveness of resource allocation

As noted in Section 2.5.2, the operation of a CVD programme in some instances can be particularly burdensome on those processing incoming reports, and frequently results in low-value information being disclosed by hackers. Aside from the resource consumption issues discussed earlier, respondents on average expressed positive sentiment (1.08) towards the benefits conferred to an organisation that allocates resources to a CVD programme. This is presented in Table 2.11. It is interesting to note the difference in sentiment from the operators of BBPs (1.25) and Both (1.5) when compared to those exclusively operating a VDP programme (0.69). The p-values when tested against a null hypothesis of equal means are $p = 0.103$ and $p = 0.0163$ respectively.

Overarching opinions

The final question in the survey asks organisations to provide an indication as to their sentiment when recommending the use of CVD programme to other organisations (see Table 2.11). The overall sentiment is largely positive towards the use of CVD programmes (1.45). When tested at the 5% level, there is a significant difference in the average sentiment expressed by those using managed programmes (2.0) and those running independent programmes (1.29) on a bug bounty platform ($p = 0.0186$).

2.7.2 Open-ended results

Within the open-ended answers provided by respondents in the survey and interviews, there are several relevant themes identified. These themes are further separated into concepts pertaining to the operation of CVD programmes in general, and concepts involving the use of third-party platforms. The themes are: general recommendations and advice, overarching benefits, general economic considerations, and the use of platforms.

General recommendations

Respondents provided recommendations and advice that may benefit those looking to launch a CVD programme. Within the theme of general recommendations and advice, four sub-themes emerged: internal recommendations and considerations, setting the programme scope, defining programme policy, and usage of rewards.

As outlined in the BSIMM framework, CVD programmes represent a security activity that may be best suited to organisations with a mature security posture — and, as highlighted in Section 2.5.2, one that can at times place significant burden on security teams. Respondents highlighted the need for maturity prior to launch: "nobody should start a CVD program before they've invested deeply internally on security" [P20]. Furthermore, small companies may face additional difficulties due to the limited number of personnel available to operate a programme: "I think you probably need a security team of a certain size so that you have resource. If you are a tiny company you may struggle to operate, but then at the same time you may not get many reports because you only have a small web estate." [P3] Naturally, once a programme is launched, organisations may need to actively refine internal processes and external communications: "there's a bunch of maturity to achieve after establishing some sort of point of contact" [P1].

It is important to have a proactive view of security throughout an organisation: "in order for a CVD program to be effective, there needs to be high-level institutional support for actually resolving security issues. Otherwise, it will be very difficult to assign and track the work" [P21].

Not only can the operation of CVD programme open-up an organisation's assets to the eyes of the unknown, but difficulties can exist when interacting with, and rewarding, individuals of

unknown origin or background. Further consideration of the legal complexities can be found in Chapter 5. One respondent expressed concerns when paying out bounties to unvetted hackers: "if you pay hackers you have to be really careful that you don't go sideways with the rules about support terrorism ... you really got to know who they are. And even inadvertent support is against the law in our country." [P33] Furthermore, particularly for US companies, this applies to awarding payouts (including virtual currencies) to any individual currently located in, or ordinarily a resident of, any sanctioned country.

A further concern for for US companies is that trade or transactions must not be made with individuals on the US Department of the Treasury's Specially Designated Nationals and Blocked Persons List (SDN) [246]. Furthermore, legal complexities surrounding vulnerability disclosure are not limited to organisations. Hackers may face confusion due to the ambiguity of export controls and multilateral treaties (such as the non-binding Wassenaar Arrangement) when disclosing the details of certain vulnerabilities, such as zero-day vulnerabilities, to entities in certain countries (see Export Administration Regulations (EAR) cyber rules [245, 244]) [200, 196]. This lack of clarity may inhibit the success of programmes operated from certain geographies as hackers may be dissuaded from participating. Given the restrictions placed on organisations, it is not surprising that one respondent noted that "the legal team will need to approve all the risk in having the programme and such. Be sure to contact the legal team as soon as you can. Because, it can be a really, it can be a blocker" [P30]. As noted in Section 2.5.3, the use of third-party bug bounty platforms helps mitigate some of these risks by providing vetting services and payments only to those with verified identities.

For an organisation about to embark on the journey of setting up and launching a programme, it is recommended that responsibilities are clearly delegated within the CVD team: "I think you need to understand who is going to triage and what, and how they identify who is responsible for fixing on the things you've put in scope" [P3]. This may help to alleviate the issues surrounding project and programme management, as highlighted in Section 2.5.2. Several respondents recommended that the CVD team should be led by an individual with a background in security and preferably with prior CVD experience, either within an organisation or as a hacker: "if you want to launch a BBP you need a person that knows about the

topic because a lot of times this [lack of experience] is bad side of the programme" [P30]. One respondent noted that they "hired a former security researcher to manage this for us" [P7]. A lack of both managerial and CVD specific experience hindered the success of programmes for several respondents.

Typically, a programme launch involves opening up a well-defined point-of-contact, enabling hackers to easily communicate reports and questions to the programme operators. Policy documents outline the terms and conditions of participation, and often include details of how to securely disclose the details of a vulnerability. Even in the absence of a formal CVD programme, respondents highlighted the need for organisations to provide a vulnerability disclosure point-of-contact. Not only does this open up a communication channel for hackers that may have found glaring issues ("there will always be people who will find something that you've overlooked but have no way of contacting you, and that's really the most important part" [P1]), but it may also benefit organisations that are impacted by the exploitation of another companies' assets: "there is nothing more defeating than, we've had the same problem when we've tried to go to companies who's websites have been compromised and have been used in phishing etc. to attack us. Often we can't even reach the people who are responsible" [P3]. As such, it is recommended that organisations, include those not yet wishing to set up a formal CVD programme, include at minimum a contact email address that is publicly viewable on the organisation's website.

In the absence of individuals with a great deal of prior experience in running a CVD programme, respondents often chose to launch their programmes with limited scope: "if you are unfamiliar with this process, and if you don't have a reasonable scope in your program this can be a massive waste of your time and resources at first" [P19]. It is recommended that the initial scope focus on "high value components (container escape, prod RCE, etc.) and not minor misconfigurations (missing HSTS header)" [P18]. Opening up all assets to hackers while having immature and untested triage, communication, and fixing processes may overwhelm the responsible teams, and infuriate both internal and external stakeholders. In summary, make sure that the scope is "very well defined" [P26], and "start small" [P33].

Further, to defining a suitable scope for a CVD programme, operators must also publish

the "rules of engagement" [145] — a set of externally-facing clauses explaining the many facets of a programme. Typically, this includes: a boilerplate company statement, details of any reward structures (e.g. $ payout amounts per CVSS score), assets deemed to be in- and out-of-scope, and Safe Harbor statements. Many respondents opted to survey the content currently used across existing programmes, seeking inspiration as to the important points that should be included: "[we] were looking at people like Google and what they had in their policies and what they were allowing to be submitted, we nicked a bunch of language from other people's disclosure policies it's fair to say" [P3].

Two respondents chose to use open-source templates to help form their initial policies. One highlighted the need for renewed attention in updating the existing templates to reflect up-to-date norms: "I also looked to a few open source initiatives, I think they are probably a bit stale now" [P9]. Others noted that hackers complained about the initial Safe Harbor clauses used in a programme: "they said — we are not happy with your Safe Harbor clauses, we don't think they are safe" [P3].

A programme policy may contain information pertaining to the rewards that may be offered to hackers, subject to the aforementioned legal requirements, upon the disclosure of a valid vulnerability report. For the users of third-party platforms, advice is sometimes given by the platform operators as to sensible initial bounty levels: "we talked to [platform] way back when, and sort of said, let's start so we don't blow through our budget immediately" [P26]. Those that set bounty levels independently offered up similar advice: "define what you are capable of supporting based on your current staffing and current funding and then grow it based on what you are seeing. Don't jump off the mountain right at the start without a parachute. Take it step by step." [P5] Over time, the rewards that may have initially been considered competitive can often appear unattractive to hackers, as often other organisations raise payouts to further motivate hackers to search for vulnerabilities within their programmes.

The need to increase payouts in order to stay competitive was mentioned by respondents: "so we noticed that although the rewards we set for our programme, they were quite high relative to other programmes a couple of years a go, we are not as unique in that sense any more" [P1]. Unfortunately, there was no consensus as to how the payout levels should be

adjusted, with the the most concise advice being "we are discussing upping the rewards again. And the other only way up, is to double the rewards." [P1]

Although the initial bounty levels are often tailored to the budgetary constraints of an organisation, external market forces can quickly render these rewards unattractive, leaving those that cannot match the desirable price levels in a disadvantageous position. Non-monetary rewards, such as Halls of Fame and company merchandise ('swag'), offer an alternative for those unable or unwilling to offer monetary rewards: "definitely do acknowledgement pages as they are free and doesn't cost you anything ... [be warned the] logistics for reward shipping is an absolute nightmare" [P3].

Overarching benefits

As indicated in Section 2.7.1, the general perception of CVD programmes was positive amongst the respondents. Several respondents provided further evidence for this in their closing remarks. In particular, it was highlighted that the operation of a CVD programme can allow "access to some of the global top 100 researchers" [P3], some of whom "take their time getting really familiar with your products, and they are not just looking for quick results" [P1], resulting in a "breadth and depth [of vulnerabilities being discovered] which simply wouldn't otherwise be achievable" [P4]. Many emphasised the uniqueness of the issues identified, including "classes of bugs that pen testers don't look for" [P3] and resulting in "paying far less for much better data" [P26].

Economic considerations

In addition to the discussion around setting bounty levels, there are several key economic themes identified that relate to leveraging the work of hackers, including cost effectiveness, influence on the market for vulnerabilities, and issues concerning the hacker 'labour' market.

A prominent theme in the responses was that of cost effectiveness when compared to alternative security investments, such as the expansion of the security team or the usage of external security services. Particularly for smaller businesses in developed countries, CVD programmes can offer security benefits at a cost that is less than the marginal cost of internal

expansion [255]. One respondent noted that "[I had] a positive view of the ROI on running one of these programmes, so its not purely a commercial decision but as a small team we have quite limited resources and the necessary the deep expertise so opening it up to the world and encouraging people to support us helped me grow my virtual team as it were" [P9]. For other respondents, their CVD operating costs were less than the cost of external audits and penetration testing services, and in many cases seemingly offered greater returns: "I've heard people say that a bug bounty programme is not a replacement for pen test. I get where they are coming from but the reality in our experience, and I would venture for many companies, a bug bounty programme is better than a pen test." [P26]. Furthermore, some respondents used their CVD programmes as a means to recruit hackers to private, and somewhat informal, penetration testing sessions: "[the programme is a] recruitment mechanism for contracting independent penetration testers and possible full time in house red teams" [P20], allowing the operator to gain the benefits of an individual already familiar with their systems for a fraction of the cost of a profession service.

Aside from the measurable costs and benefits that can be achieved through the operation of a CVD programme, there are two additional benefits that respondents thought demonstrated their cost effectiveness. First, as is commonly touted when discussing security investment, the prevention of the exploitation of vulnerabilities is often far less costly than the intangible (e.g. reputation damage) and tangible (e.g. fines) costs bore by an organisation following successful exploitation: "in retrospect, the cost of operating the programme (engineering time, finance time, bounties, etc.) is significantly less than the potential cost of exploited issues had they not been patched" [P9]. In the case of CVD programmes, this 'value' may be more perceptible as the severity and wider impact of a discovered vulnerability is analysed throughout remediation. Second, the operation of a CVD programme may demonstrate an organisation's commitment to security investment, or confidence in the security of their systems: "often a bounty program is security theatre because it's some public statement of *we're good enough, try to hack us*" [P18]. This public-facing approach to security may lead to greater intangible benefits, particularly in customer relations, when compared to some less obvious security investments: "CVD programmes shows our customers we care about security and contribute to

reassure them their assets are safe with us" [P13].

The exchange of vulnerability information via a CVD programme represents a subset of the potential avenues within the regulated vulnerability market [11]. Within this 'white' market, itself a subset of the complete vulnerability market, transactions are formally recorded and the information is disclosed to the legitimate owners [8, 164, 170]. Outside of regulated markets, grey and black markets that serve as the information exchanges between hackers and buyers, in which the receiving party need not be the legitimate owner. For example, aside from the obvious use of black markets by malicious entities, government organisations may procure vulnerabilities for defensive and offensive purposes [214]. In the latter case, the role of government agencies constitutes an ethically complex situation, as many are not automatically obliged to disclose the details of a vulnerability to the responsible parties (see the UK Governments' Equities Process for an example [232]). For hackers, these grey and black markets can offer substantial rewards for the sale of a vulnerability, in many cases matching or exceeding the rewards offered by organisations in regulated markets [8]. Given the existing complications associated with prosecuting hackers for issues surrounding cybercrime [26, 28, 49], it may be that the perceived or actual risk in using illegal or legally questionable markets is too low to act as a deterrent. Thus, although CVD programmes incentivise some hackers to only engage in regulated markets [262], they alone have not eliminated the use of alternative markets.

Respondents commented on the positive effect that their CVD programme has had in mitigating the perceived threat of a hacker selling vulnerability information to other entities: "it is a strong way to provide an incentive to have security vulnerabilities to be reported to your company rather than being sold to or used by bad actors" [P14]. This "alignment of incentives" [P37] gives social benefits to hackers and organisations, but also benefits the (external) consumers, albeit with the potential risk of "incentivizing your product as a target" [P35]. The growing role of bug bounty platforms as a third-party commercial intermediary in regulated markets complicates the previously (somewhat idealised) aligned incentives between market participants. As a for-profit entity, often with significant venture capital funding [57], the incentives for the platforms may not be completely aligned with the needs of the organisations operating CVD programmes on the platform. (The negative effect on business sustainability

resulting from venture capital investment is explored in [126] and [190].) Respondents commented on this issue, particularly in reference to managed programmes: "I think the problem is very much with their business model to begin with, it is in their best interest to make money obviously, and it doesn't align a lot of what the programmes want to do" [P15].

In addition to discussions surrounding the markets for vulnerabilities, respondents also commented on the existing structure and incentives of the hacker 'labour' market. Reports published by platform owners highlight the geographic distribution of their registered user-base (of hackers), demonstrating the global community they are able to leverage [100, 99]. Significant numbers of hackers are based in the United States of America, Russia, and India, with smaller numbers coming from almost 170 other countries [100]. Having a large user-base located outside of developed, high-wage, countries brings with it great comparative advantage: hackers in lower-income countries will be willing to work for the prospect of far lower rewards than those in higher-income countries: "$500 isn't that much money ... if it takes somebody 10 hours, maybe 15 hours, it is really good for where they are in the world, a lot of these bug testers are in, it seems, Asia" [P26]. In comparison, with the average wage in the US being $70,000 (US in 2020)[5], CVD rewards may appear less appealing for US based hackers, but more enticing for US organisations: "for us, working in mostly US and Canadian wages this calculation really works out for us" [P1].

Some respondents spoke negatively of the current labour market structure involving the use of hackers: "bug bounty programs are 'gig economy' and somewhat exploitative especially if they allow focus [to reject reports based] on minor errors" [P18]. Certain aspects of the CVD process not only exemplify the 'gig' economy, but go one step further. It is typical for only the first hacker to disclose a vulnerability to be rewarded, subsequent reports are marked as duplicate. If hackers work in isolation and without coordination, it is reasonable to expect that multiple individuals will independently (and perhaps simultaneously) discover and disclose the same vulnerability. For those other than the first discloser, any effort will have been done in vain. One respondent that had worked as a hacker noted that "I draw

[5] " ... average wages are obtained by dividing the national-accounts-based total wage bill by the average number of employees in the total economy, which is then multiplied by the ratio of the average usual weekly hours per full-time employee to the average usually weekly hours for all employees" [238]

parallels between bug bounty hunting and gambling ... it is not really deterministic, it is very random. And, especially if you depend on that money, it is not the best way to go about living, in my opinion" [P15]. Schechter previously discussed the facets that make for a 'good' market (Böhme notes that imperfections exist due to demand-side price setting [33]) for the sale vulnerabilities [205, 206]. He suggests that organisations only reward the first instance of a vulnerability report in order to maximise value. However, he notes that complications may arise from vulnerability reports not being disclosed immediately by organisations, as other hackers will remain unaware that their work will lead to no reward due to duplication. This added risk may cause hackers to demand higher rewards for their work [205]. Another respondent highlighted the use of alternative business models to reward the efforts of hackers: "[there is a business] model where they give money upfront so they will give you, not necessarily a bounty, but a bit of money to say, 'can you go and focus on this?', so you are more willing to invest the time. You are basically being paid for your time, and that is not typical in bug bounty hunting" [P15]. Such a paradigm shift may benefit those hackers that are already established and have demonstrated their potential value to an organisation. For an organisation looking to leverage hackers, it is perhaps useful to be mindful of these alternative approaches to incentivising the disclosure of vulnerability information.

Platforms and their services

Respondents offered a plethora of advice regarding the use of platforms and managed programmes, covering: platform advantages and disadvantages, platform pricing, managed programme advantages and disadvantages. As stated previously, hosting a CVD programme on a third-party platform often comes with many built-in services that ease the operation of a programme. Especially when setting up a programme, respondents noted the usefulness of these services: "they provide you some guidelines, like okay 'according to your industry, this is like the standard', when not the standard, this is the average from other companies that are in your industry and maturity level" [P30]. Furthermore, integrated payment and reward systems shift the burden of legal checks and payment processing away from the operator: "I like that [platform] just does it and its, they do the points and if we do money, great, its easier

that way than doing it ourselves. I wouldn't want to go back to doing it ourselves just because it is annoying to deal with that stuff." [P26]

Aside from any other advantages mentioned elsewhere, the use of a platform can give organisations a significant boost in visibility within the hacker community. One respondent feared moving off-platform as "you lose all the visibility from the community" [P30]. On the contrary, respondents noted several key drawbacks to hosting on a platform. First, although it may be advantageous for many to have great programme visibility, this can come with the possibility of a large number of low quality reports being submitted: "commercial bug bounty platforms often bring more harm and noise than good" [P20]. Second, there is a great deal of opacity when it comes to the cost of utilising their services: "some part of the license or pricing were not so clear ... it is not like you go on the website and they say 'this is the price you will pay', you have to go with their sales teams" [P30].

Respondents were most vocal about the prohibitively high costs associated with hosting on a platform and making use of their services: "They are not too cheap, and I'm thinking if you are a small company and you want a bug bounty programme, it is not really viable because it is not cheap. Just the fee for [hosting on] the platform is not cheap" [P30]. For responding in smaller organisations, the sentiment was echoed as it was not always justifiable to pay these expenses: "we also expect our cost of using a third-party tool to be greater than our cost of running the programme (even with the high volume of invalid reports)" [P9].

Comments on the positives of managed programmes echoed those expressed within earlier sections. Respondents value the ability of the external triaging team to filter out low quality or duplicate reports: "I find them very useful in filtering out this low-hanging fruits. So my response to things, when I get an email from the programme, I can tell by the initial description that this is the low-handing fruit, and I can leave it to them to clear this out" [P1].

As discussed in Section 2.5.3, managed programmes are an effective countermeasure to the noise received by a programme. On the other hand, respondents criticised the lack of contextual understanding on the part of the external triaging and fix validation team in respect to the systems and assets of an organisations: "they provide the triage feature, but they of course don't have our context" [P30]. This can lead to a misunderstanding as to what type of

reports an organisation values, and a mis-characterisation of a report's severity: "we've had breach reports closed as out of scope by triage personnel on both platforms" [P20]. A lack of contextual understanding and product knowledge was noted by Ozment as a difficulty faced by TTPs when verifying vulnerability reports [181].

2.7.3 Summary of recommendations

The overarching consensus amongst respondents was that CVD programmes are beneficial to security, a good use of resources, and are highly recommended security activity that organisations can adopt to help secure their assets in a cost effective manner. Using the results in Section 2.5.5, Section 2.6.3, as well as results elsewhere in Section 2.7, a list of recommendations is made for those looking to set up a CVD programme.

Pre-deployment recommendations

- Open up a point of contact. As discussed in Section 2.7.2, it is highly recommended that a point of contact between hackers and a responsible internal team is established. This may be as simple as a secure form on an organisation's website or provision of an email address. Obviously, this is required for any organisation looking to set up a CVD programme. It is also recommended for those not yet willing to launch a programme as a means to establish a method by which vulnerabilities can be communicated.

- Require managerial experience. As discussed in Section 2.7.2, it is highly beneficial to have an individual with prior experience in running a CVD programme lead the creation of a new programme.

- Deploy experienced staff. One of the issues reported by respondents in Section 2.5.2 and by Al-Banna et al. [9] is that of inexperienced staff within the CVD or triaging team. Two respondents had a background that involved participating in CVD programmes, and this undoubtedly gave them significant insight into the practice as a whole. It should certainly not be the case that staff have no cybersecurity background (see Section 2.5.2).

- Restrict initial scope. A common countermeasure to a high volume of low-quality, low-value, reports involves restricting the scope of a programme to only cover those assets deemed to be of higher importance (Section 2.5.3). However, as recommended by respondents, organisations should start off with a narrow scope to prevent issues from arising after launch (Section 2.7.2).

- Reward only valuable reports. In conjunction with the previous recommendation, respondents often chose to exclude certain classes of vulnerabilities from reward structures, instead opting to only pay out for higher value reports. Commonly this is achieved by excluding low (and sometimes medium) severity vulnerabilities. Although no quantitative advice is provided in order to help set bounty levels, the prevailing comments suggest setting the initial bounty levels to an amount that is sustainable for a given budget. This is clearly an area that would benefit from further quantitative analysis. Furthermore, given the complexities around payout bounties to hackers (Section 2.7.2), it should be decided whether monetary, non-monetary, or no rewards will be offered to hackers.

- Define internal information pathways. To ease the operation of the programme once launched, it is sensible to have a well-defined system in place for tracking the status of report vulnerabilities (Section 2.5.2), finding the responsible (internal or external) product owners (Section 2.5.2), and communicate patterns of reports (as part of a rudimentary BBLC) to relevant parties (Section 2.6.3).

- Seek support from upper management, legal, and product teams. As noted in Section 2.5.2, if internal teams are unwilling to fix vulnerabilities, the benefits to operating a CVD programme are restricted. Further, without authorisation from the legal team, a programme may be entirely unable to operate (Section 2.7.2).

Post-deployment recommendations

- Modify reward structures to incentivise hackers. An issue faced by respondents was that of low hacker motivation (particularly after the discovery of 'low-hanging fruit' (Section 2.5.2)), which limits the number of reports that operators receive. Some re-

spondents suggested increasing current payouts as a solution to this issue (Section 2.7.2). The anecdotal evidence used by organisations to increase bounty levels prompts further quantitative analysis into the effect raising bounty levels has on the returns of the programme.

- Use incoming report information for reactive security. A majority of respondents used the results from their CVD programme to include (shallow) changes within their organisation (Section 2.6.3). For an organisation not currently utilising their reports, it may be beneficial to use them in a reactive manner to influence the security activities outlined in Section 2.6.1.

- Use incoming report information for proactive security. For organisations with a mature CVD programme, the deeper integration of report information into organisational security strategies may provide further benefit. As detailed in Section 2.6.2, only a few respondents reported to using such an approach.

2.8 Summary of findings

Motivated by the question *How effectively are organisations utilising CVD programmes?*, we have considered topics surrounding the use of CVD programmes within organisations via 39 survey responses and 8 semi-structured interviews. A summary of the findings to the subsidiary research questions presented in Section 2.2 are as follows:

1. Building on the work of Al Banna et al. [9], we found that many of the previously identified problems associated with the operation of a CVD programme persist: organisations still struggle with high volumes of poor quality reports that burden the operators and provide little benefit to security. Respondents suggest that alterations to the definitions of in-scope and out-of-scope assets can help to mitigate these issues.

2. Although a majority of organisations report making changes to security activities based upon the content of vulnerabilities reports, there is little evidence suggesting that organisations deeply integrate the results into a form of BBLC. Respondents typically did not

institute change throughout their SDLC to the extent of organisations such as Verizon Media.

3. Overall, respondents highly recommended the use of CVD programmes as a cost-effective security activity. Both pre- and post-deployment recommendations were offered to those looking to set up a CVD programme. Designing reward structures to incentivise hackers and avoid rewarding low-value reports, restricting programme scope, and having support from experienced staff and management were all reported to be important considerations.

It is important to recognise the limitations of the work presented in this chapter. First, the large non-response rate may limit the generalisability of the findings as only a small proportion of the population could be sampled. Furthermore, results may be biased towards certain sub-populations. Notably, all organisations that participated in the interviews were predominantly based in the US, UK, and Canada. As such, certain considerations, particularly those surrounding economic and legal issues may be less applicable to organisations based in other regions. The interpretation of the qualitative results, and the answers provided by respondents, may be subject to additional cognitive biases. Confirmation bias, hindsight bias, and perhaps a tendency to portray an organisation's activities in an overly positive light, may impact the conclusions that are drawn from the results. Finally, while the work presented in this chapter inadvertently captured insights from several respondents that had participated previously as hackers, it does not attempt to broadly survey the opinion of hackers in general. Nevertheless, it is hoped that what is presented is beneficial to those interested in the theory and practice of CVD programmes.

In the next chapter we use empirical data to further explore the characteristics of bug bounty programmes, with the ambition to quantify some of the costs and benefits discussed by respondents throughout this chapter.

Chapter 3

Current State of Bug Bounty Programmes and Platforms

Following on from the investigation of organisational perspectives of CVD programmes in Chapter 2, in which bug bounties were defined as a sub-type of CVD programme, the current state of bug bounty programme operation is further explored and discussed within this chapter. This is carried out with the ambition to answer to research question *Is it beneficial for a company to make use of bug bounty programmes, instead of hiring additional software engineers?*

Further background information on bug bounty programmes, and the use of third-party bug bounty platforms is presented in Section 3.1. Despite previous empirical works exploring the state of programmes, in an ever growing market there is need for renewed investigation in order to better understand the current state of bug bounty programmes. The motivation behind the work presented in this chapter is described in Section 3.2. Using empirical data collected from bug bounty platforms, quantitative insights can be produced. The process of data collection and analysis is outlined in Section 3.3. From Section 3.4 to Section 3.11, the insights are presented and discussed with the ambition to not only answer the research questions presented in Section 3.2, but with the aim to also set the scene for the remaining chapters. A concise summary of the findings is found in Section 3.12, and the chapter ends with a discussion of the limitations of these insights.

The work presented in this chapter is based on [255]. It is important to note that the costs described throughout the chapter were correct when the analysis was conducted in late 2019.

3.1 Background to bug bounties

In an increasingly digital economy, cyber crime has far-reaching effects on both businesses and individuals. As a response to a vast increase in the number of cyber attacks targeting organisations, security budgets have soared to record levels, in the hope that the risk of an attack can be mitigated [2].

Following a security breach, loss of revenue and fines can result in an immense burden on an organisation. In recent years we have seen British Airways fined $230 million, a $124 million fine for Marriot, and $575 million for the 2017 Equifax breach [218]. Exposing the digital data of an individual through a data breach may reveal far more confidential information than can gained from a physical breach (e.g., due to a burglary), and may have many real-world implications on the individual, such as increased risk of financial or identity fraud [171]. An early survey by IBM highlighted growing consumer distrust in organisations that hold their personal information in a digital form [120]. As of 2017, approximately 64% of American adults reported falling victim to a data breach and having their personal information exposed [178]. It is well understood that, in order to reverse the trend of increasing cyber crime and consumer distrust, organisations need to further improve their efforts at addressing the vulnerabilities being currently being exploited.

A study by Bendovschi [29] identified the three root causes of a security breach: criminal intended attacks (malicious actors), human error, and system vulnerabilities. Bendovschi found that malicious actors were the root cause in fewer than 50% of attacks, noting that an attacker's skills and knowledge are complemented by the abundance of both social and technical vulnerabilities that can be exploited.

There are many contributing factors making up each of the root causes. Malicious actors benefit from lagging legal frameworks that fail to cover the latest technological developments [142] and the ineffectiveness of prosecutors to successfully take legal action against

them [253]. Human error manifests itself in the actions of consumers and employees. A lack of security awareness among individuals and improper training of employees may leave both groups susceptible to social engineering attacks [125, 129]. System vulnerabilities may be present due to, among other factors, misaligned incentives in the software industry in relation to the production of bug-free software [19], or the presence of design and implementation flaws [162].

Several approaches have been proposed to address security concerns within software development. For example, the adoption of a Secure Software Development Lifecycle (SSDL) is one method to address this problem. Microsoft's Security Development Lifecycle (MS SDL) was one of the first industry-leading frameworks to encourage thinking about security at all stages of development [173]. As part of the framework, 12 practices provide guidance, ranging from pre-development security training to post-deployment standardised incident response procedures [169].

As introduced in the previous Chapter (see Section 2.1), the Building Security In Maturity Model (BSIMM) is an alternative approach to advising security managers on the subject of security; it reflects on the current security practices employed by 120 organisations [223]. Four domains (Governance, Intelligence, SSDL Touchpoints and Deployment) are broken down into 12 practices, each covering different aspects of security. For each practice, activities are categorised into three levels of maturity. Security managers are able to identify activities that will improve their organisation's security and gain insight into the high-level deployment details.

The act of operating a bug bounty programme is the most popular mature activity within the Configuration Management & Vulnerability Management (CMVM) category, and one of the most popular mature activities listed in the BSIMM (the 6^{th} most popular Level 3 activity, with 10.8% participation). This activity describes the operation of a bug bounty programme as a mature method of vulnerability management, used to improve the security of a system through the identification, re-mediation and mitigation of software vulnerabilities [82] in the release and maintenance phases [42]. As of 2018, 13 of the 120 organisations considered in the development of BSIMM operate a bug bounty programme [223].

Many large organisations (including Google, Facebook and Microsoft) host their own bug bounty and vulnerability rewards programmes [94, 73, 168]. Unlike smaller organisations, these companies have enough visibility to hackers due to their reputation and market prevalence. Thus, the need to advertise a programme through a third party is removed. One problem faced by large technology companies is the sale of their vulnerabilities on the black and grey markets. For example, zero-day vulnerabilities found in Apple's iOS have been known to fetch up to $1 million on grey markets [165] — far higher than any bounty payout. Furthermore, with few or no legal repercussions, hackers would avoid the legitimate approach to submitting vulnerabilities to Apple in favour of the more lucrative opportunities available on other markets [185]. A relaunch of Apple's programme saw the maximum rewards match the grey and black markets prices, with the hopes of incentivising hackers to choose the legitimate option [172]. Sutton and Nagle discussed the impact of these alternative markets, noting that the existence of other markets may force vendors to pay hackers for vulnerability information [217].

Many smaller organisations choose to use a third party to host, advertise and often run their bug bounty programme. Some popular examples of these bug bounty platforms include HackerOne [103], Bugcrowd [41], Cobalt [51], Synack [219], Intigriti [121], and Yogosha [265].

The platforms HackerOne, Bugcrowd and Intigriti offer publicly available programmes that allow anyone to sign up to and start participating in the search for vulnerabilities. Free hosting allows organisations to advertise their programme to hackers, with only a 5% surcharge on any bounty payouts. Monetisation of the platform comes in the form of live support and the operation of a fully-managed programme — allowing for organisations without experience to benefit from the operation of a bug bounty programme. In addition to the public programmes, each of HackerOne, BugCrowd and Intigriti offers private programmes that enable organisations to invite trusted hackers to search for vulnerabilities in domains not available to the public (e.g., in beta versions of software, or in the source code).

Cobalt, Synack and Yogosha operate using a different user-base model. All hackers must submit an application and undergo vetting and testing before they are allowed to participate in the search for vulnerabilities. This process typically involves: third party government ID verification, a review of their social media accounts, skills assessments, and a video interview.

As a result, organisations may be able to place greater trust in both the skills and ethics of these hackers. In Section 2.5, it is noted that the trustworthiness of hackers continues to be an issue discussed by operators. One downside to the strict entry requirements set by these platforms is their smaller user-base in comparison to platforms that offer public participation. For example, as of the 21^{st} of March 2020, Cobalt had a total of 282 certified hackers, of which 254 have participated in the search for vulnerabilities [52]. In comparison, both HackerOne and Bugcrowd had over 300,000 registered users [99, 45]. However, the reliability of the user metrics is debated (as we shall see in Chapter 4). As noted in the previous chapter, one benefit offered by platforms is the increased visibility of programmes, allowing for a large number of security researchers to search for vulnerabilities [140, 195].

White-hat hackers are independent security researchers that are authorised by an organisation to identify security vulnerabilities in hardware, software or networks [197]. They must comply with the 'rules of engagement' set out by an organisation to prevent misuse or intrusion into areas that are out of scope [145]. Kranenbarg et al. [137] recommend that new hackers should be taught about the rules around vulnerability disclosure so as to avoid accident misuse.

In 2018 the International Organization for Standardization (ISO) published the latest version of ISO/IEC 29147 concerning requirements and recommendations to vendors on the disclosure of vulnerabilities and services [233]. However as the standard is not freely available to the public, it is of limited value to hackers.

HackerOne's yearly hacker report presents the results of a survey of the users; additionally, it summarises the recent successes and milestones reached by the platform. The 2020 edition details the results from over 3,150 respondents, giving insight into the background of hackers [100]. It is found that hackers from 146 countries participated in the search for vulnerabilities on HackerOne. Countries with the most number of users are: India (12%), the US (11%), and Russia (11%). Given this, it is unsurprising that hackers from India (18%) and the US (11%) submit the highest percentage of total reports. Hackers from the following countries receive the highest share of the total bounties awarded each year on HackerOne (almost $40 million in 2019): the US (19%), India (10%), Russia (8%), and China (7%).

Although not yet released, it may be expected that the geographic distribution of partic-

ipating hackers presented in HackerOne's 2022 report will be significantly altered due to US sanctions on Russia. As a result of the sanctions, US based platforms, such as HackerOne, no longer issue payments to any Russia-based hackers[1].

On the Bugcrowd and Cobalt platforms, a similar group of countries hosts the most hackers. The top three countries on Cobalt are: India (15%), the US (15%), and the UK (8%) [52]. Bugcrowd has the same three top countries; however, the number of users from each is not published.

The survey results also show the education levels of the hackers. Of those surveyed, 41% report studying computer science or programming at an undergraduate level, and 34% studied these at graduate level. Additionally, 27% of respondents are currently students and participate in bug bounty programmes to learn skills and techniques that will help with a career in security engineering. This crowdsourced approach to security brings together hackers from many backgrounds and disciplines, resulting in a wide range of approaches to finding vulnerabilities that might not otherwise be achieved by an organisation's security team [252, 146].

As of 2020, six users of the HackerOne platform had surpassed $1 million in lifetime earnings. Furthermore, 146 users had earned over $100,000 in total, with 50 users earning this in 2019 alone. These figures are indicative of the lucrative rewards that a hacker can earn if they are the first to discover and document a critical vulnerability, with the highest single payout (at the time of writing) standing at $100,000 [136]. This also legitimises the work of many hackers who might otherwise sell the vulnerability reports on cyber crime markets for a similar price [14].

Participating in bug bounty programmes is a viable method of full-time employment for 22% of hackers. However, as a majority of users make less than $20,000 per year, the feasibility of using bounty payouts as a sole source of income is often limited to hackers based in countries with relatively low average salaries (and costs of living) in comparison to Western nations.

HackerOne's 2020 report also highlights some of the issues faced by hackers that prevent them from reporting vulnerabilities. It is found that 38% of hackers did not report a vulnerability after finding it due to "threatening legal language" used by organisations in relation

[1]See `https://www.hackerone.com/press-release/hackerone-releases-faq-regarding-sanctions`

to their vulnerability reporting and disclosure policies (explored further in Chapter 5). A further 21% were unable to report a vulnerability due to a lack of an obvious communication channel with an organisation. The prevalence of these two issues highlights the importance of initiatives such as the aforementioned Disclose.io project (see Section 2.1).

3.2 Motivating factors

3.2.1 Empirical studies surrounding BBPs

In order to give appropriate consideration to both related work and our question of interest (presented in Section 1.2), we first need to establish appropriate criteria. One such criterion is the cost of employment of a software engineer, which we calculate as follows. The average salary for a London-based software engineer is £41,700 [91], which is equivalent to $51,708 (exchange rate £1 to $1.24, as of 17^{th} of July 2019). Of course, the hiring process has additional initial costs when new staff are employed — Blatter et al. [31] estimated the cost of hiring new skilled workers to be equivalent to 10–17 weeks salary. This puts the total cost of hiring an additional software engineer at between $61,652 and $68,613 for the first year. Consequently, an average value of $65,133 will be used to represent the cost of hiring an additional software engineer throughout this chapter.

Finifter et al. [75] considered the vulnerability reward programmes offered by Google and Mozilla for the Chrome and Firefox web browsers. Vulnerability reports over three years (2010–2013) were analysed to find the payouts, severity and frequency at which reports are submitted. Comparing the two programmes showed that Google fixed three times as many vulnerabilities identified by hackers than Mozilla. According to Finifter et al. [75] this is due to a larger number of white-hat hackers (when compared to the Google programme) taking part in the programme, as well as a broader reward structure.

The daily cost to operate each programme is reported as $485 for Google and $658 for Mozilla; over the course of a year, the total cost is given as $177,025 ($485 × 365 days) and $240,170 ($658 × 365 days) respectively. This is broadly comparable to the salary of three or four additional software engineers (assuming our estimated current average salary of

a software engineer of $65,133). The authors argue that it is economically viable to run bug bounty programmes instead of hiring additional researchers. Further work is recommended by the authors to include economic models to identify phases during the operation of a programme which, in part, motivates the present contribution.

In Section 2.7.2, the operators of CVD programmes put forward a positive view of the Return On Investment (ROI) that may come with the operation of a programme. One operator also noted that the use of hackers could be thought of as the creation of a 'virtual team', helping address the need for cost-effective vulnerability discovery in conjunction with their smaller internal security team.

A study by Zhao et al. [271] analysed web vulnerability reports on the Wooyun and HackerOne bug bounty platforms. Wooyun served as the predominant platform in China from 2010 until being shut down in 2016 [263]. Analysis of vulnerability reports is used to determine the number of white-hat hackers participating in the search for vulnerabilities. HackerOne is shown to have a consistent growth of approximately 75 new hackers each month from 2014 to 2015. Wooyun received almost 200 hackers each month; however, as most of the programmes do not give rewards, the incentives to participate may be different from comparable sites. An interesting observation is that over half of all hackers have only submitted one report; the top 100 have an average of 147 submissions per person. Basic economic analysis by the authors shows that offering a monetary reward attracts more hackers than non-monetary rewards or acknowledgements in a "Hall of Fame".

Unfortunately, the authors of [271] do not look into the use of the delayed full disclosure policy used by Wooyun. This controversial policy can be used to force an organisation to fix vulnerabilities more quickly. However, the policy creates conflict between organisations and hackers, which is one of the contributing factors that led to the shutdown of Wooyun in 2016, with several members being arrested [184].

3.2.2 Objectives

Since the 2015 study by Zhao et al. [271], the number of programmes available on popular bug bounty platforms (such as HackerOne and Bugcrowd) has continued to increase. And,

for the oldest programmes on HackerOne, over six years of disclosure data are available. This significant increase in publicly available data gives strong motivation to re-verify the hypotheses presented in the previous studies by Zhao et al. [271] and Finifter et al. [75]. Furthermore, a temporal study can be conducted investigating the frequency and severity of reports over time, allowing for the long-term impact of operating a bug bounty programme to be assessed and compared to the impact of hiring additional security researchers. In conjunction with the recommendations put forth in Chapter 2, the creation of new economic models for bug bounty programmes has the potential to be useful to organisations wanting to set up a new programme or maintain an existing one.

This chapter seeks to address the research question *Is it beneficial for a company to make use of bug bounty programmes, instead of hiring additional software engineers?* through the following set of subsidiary questions:

1. What is the expected cost per year of operating a bug bounty programme, and what results can be expected?
2. What is the expected benefit per year?
3. How does the programme activity vary across its lifetime?
4. Does the promise of higher bounty payouts result in an increased number of vulnerabilities being reported

Question 1 aims to quantify the cost to an organisation of running a programme. This running cost can be directly compared to the average annual salary and cost of hiring a software engineer (which, as discussed, we have estimated at $65,133). One argument in favour of making use of bug bounty programmes is an economic one: the cost of operating a programme should, ideally, be less than the cost of hiring security researchers, while yielding a similar number of vulnerabilities found per year. The benefit of operating a bug bounty programme is explored in Question 2. Question 3 considers the long-term effect of operating a programme. Understanding how the rate of vulnerability discovery changes over time allows for the long-term costs of operating a programme to be estimated. Finally, Question 4 considers the

programme rewards structure and the effect this has on hacker productivity. A company will want to keep the operating costs low and the rate of vulnerability discovery high. Investigating the link between bounty payouts and rate of vulnerability discovery allows for an optimal rewards structure to be proposed.

3.3 Empirical data

Statistics were collected from programmes hosted on bug bounty platforms. HackerOne and Bugcrowd were chosen because of the large quantities of publicly available data on their platforms. They are also the two most searched for bug bounty platforms [93]. Synack also provides a platform that is growing in popularity; however, there is currently no public data provided [219]. The lack of publicly available data and poor transparency in many programmes is an issue that inevitably limits current research [34, 50]. Increased transparency may benefit both hackers and organisations wishing to learn about uncommon vulnerabilities that might be found [199].

For HackerOne, all of the programme data available on the directory was collected. The data for each programme was: programme launch date, number of reports resolved, minimum bounty, and the average bounty. The Hacktivity section provides information on the latest reports submitted. Each entry contains the username of the hacker that submitted the report and the time of submission. Additional information is sometimes displayed, which includes: a description of the vulnerability, severity level, and the bounty awarded. Every entry in this section was collected. The leaderboard provides information on the top 100 hackers: reputation is used as a means to quantify of number of valid reports submitted, signal is the average reputation per report, and impact is the average reputation per bounty. This information was collected for user analysis, and remains distinct from the user data collect for the behavioural analysis presented in the following chapter. A summary of all data collected from HackerOne is shown in Table 3.1.

Programme data was also collected from Bugcrowd; this information was not as extensive as the comparable data available on HackerOne. For each programme, the number of

Programme statistics	Response efficiency
Total bounties paid	Average time to first response
Average bounty range	Average time to triage
Top bounty range	Average time to bounty
Bounties paid in the last 90 days	Average time to resolution
Reports received in the last 90 days	Meet response standards
Last report received	
Reports resolved	
Hackers thanked	

Table 3.1: Summary of types of data collected from HackerOne programme pages.

Bugcrowd user data
All-time points
Current rank
Total vulnerabilities
Accuracy
Average severity

Table 3.2: Summary of types of data collected from Bugcrowd programme pages.

vulnerabilities discovered and the average payout was collected. Bugcrowd provides detailed user performance statistics for each hacker, this is summarised in Table 3.2. This data was collected for users with the highest number of points in a programme's Hall of Fame. All data was collected from December 2018 to January 2019. A summary of all data collected is shown in Table 3.3.

3.4 Cost of programme operation

3.4.1 Yearly expenditure

The total cost to run each programme is given on HackerOne's directory. Taking this figure and dividing it by the total number of days since the programme was launched gives the average daily running cost for each programme. The average daily cost of operating a bug bounty programme is \$230, with a 95% Confidence Interval (CI) of [\$126, \$334]. This gives an average annual cost of \$83,950 (\$230 $\times$ 365 days). This value is greater than the \$65,133 required to hire each additional software engineer.

The daily cost of operating each programme is displayed in Figure 3.1. This shows that

	Platform	
	HackerOne	Bugcrowd
Programmes	212	99
Reports	5,832	Not available
User data	100	92

Table 3.3: Summary of quantity of data collected from HackerOne and Bugcrowd.

a large proportion of programmes have an operating cost of $250 per day or less. There are several outliers found on HackerOne; we consider these below.

- PayPal launched a programme in August of 2018 and paid out over $700,000 within the first six months. The high operating cost of $3,766 per day is due to the large number of critical vulnerabilities reported; these pay out up to $23,000 each.
- At the time of data collection, the newest available programme was Postmates, and within one week after launch they had already spent $26,124 to find 75 vulnerabilities. The daily cost was $3,732; this high operating cost is expected to fall as the obvious vulnerabilities are reported.
- Oath's programme, launched in 2014, has cost almost $1,000,000 per year to run. The daily cost was $2,730. This is the most active programme on HackerOne with over 5,000 vulnerabilities reported.

Severity ratings are either set by the organisation for each type of vulnerability (e.g., remote code execution designated as a critical vulnerability) or calculated by the hacker using a Common Vulnerability Scoring System (CVSS) score [78]. CVSS scores are calculated using exploitability, impact, temporal and environmental metrics, and represent the overall impact of a given vulnerability. A typical conversion from a CVSS 3.0 score to severity rating is shown in Table 3.4.

Table 3.5 shows the average payout for a vulnerability of a given severity, with Figure 3.2 showing the payout distribution. The distribution is visualised using a violin plot that combines elements from traditional box plot with a density plot (generated via a kernel density function), and aims to provide more detailed insights into the shape and variability of the underlying data.

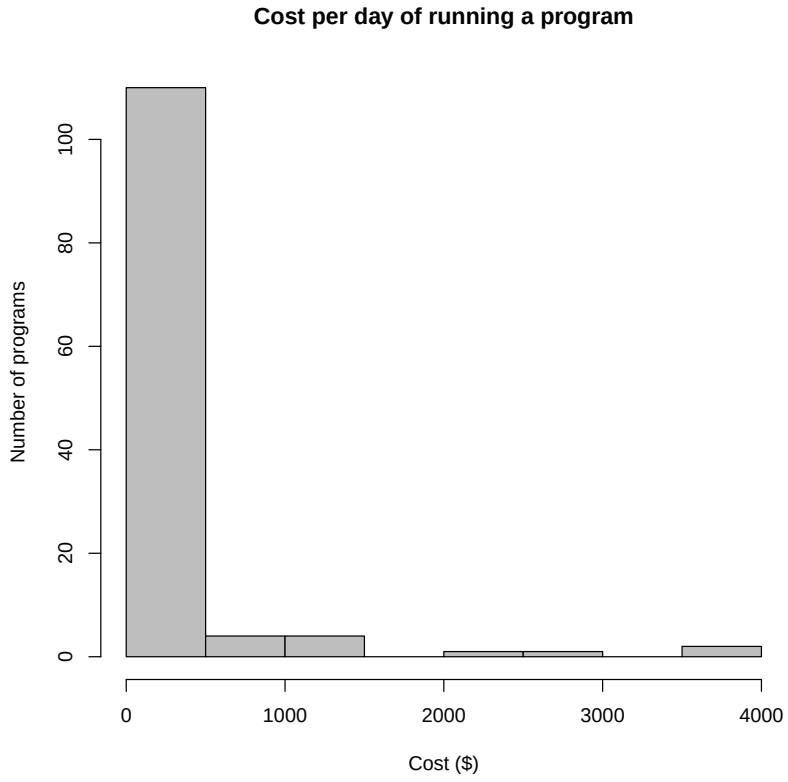

Figure 3.1: Histogram of the cost per day to operate a bug bounty programme.

Severity	Critical	High	Medium	Low	None
Score	10.0–9.0	8.9–7.0	6.9–4.0	3.9–0.1	0.0

Table 3.4: Conversion from CVSS 3.0 score to HackerOne severity rating.

Munaiah et al. [174] found that there is a weak correlation between CVSS score and the bounty payout, with the CVSS score being more likely to underestimate the actual bounty payout. This is also found in the skewed distribution shown in the figure. As discussed by Finifter et al. [75], certain architectural decisions can lower the proportion of critical vulnerabilities present in software — which has the effect of lowering the operating cost. For example, this is seen when comparing the privilege-separated architecture of the Chrome web browser to the Firefox web browser: the former has a lower proportion of critical vulnerabilities [75]. The payout distribution for each level of severity is shown in Figure 3.2. Critical vulnerabilities have the largest variation in payouts, ranging from $200 to $200,000 per report.

Severity	Proportion (%)	Average cost ($)	95% CI
Critical	8.5	7,924	[$3,324, $12,524]
High	19.6	2,610	[$2,046, $3,174]
Medium	33.5	786	[$618, $954]
Low	28.8	225	[$177, $272]
None	9.6	0	[$0, $0]

Table 3.5: Severity of the most recent (as of the time of data collection) 3,000 disclosed reports on HackerOne, along with average payouts.

3.4.2 Analysis

With an average annual cost of $83,815, the cost of running a full-time programme is greater than the $65,133 required to hire an additional software engineer. The running cost is highly dependent on the success of hackers at finding vulnerabilities, as well as the rewards structure in place. Oath's programme costs almost $1,000,000 per year to run, which is, in part, due to the high payout for critical vulnerabilities found in over 31 web domains and within 21 Android and iOS applications.

Finifter et al. [75] found the daily operating cost to be $658 and $485 for Firefox and Chrome. While these are above the current operating average of $230, they are not outliers in the current distribution of operating costs.

As the severity of vulnerabilities present in the software cannot be predicted before the programme is launched, the relative proportion of critical vulnerabilities is unknown. Once a programme has been launched, the rewards structure can be altered if large numbers of high-severity vulnerabilities are reported. This can be used to reduce the operating cost.

Zhao et al. [271] gave consideration to the severity distribution of vulnerabilities on Wooyun. Only 10% of submitted vulnerabilities were classified as of low severity, compared to 38.4% of vulnerabilities on HackerOne being classified as of low or no severity. The general decrease in high severity vulnerabilities could be in part responsible for lowering the average operating costs, as the payouts are far less for low severity vulnerabilities.

Another trend is the increase in payouts for critical and high severity vulnerabilities. Firefox and Google typically paid out $3000 and $1000 for vulnerabilities of this severity [75]. The maximum bounty on HackerOne in 2015 was $7,560 by Twitter [271]. The current average

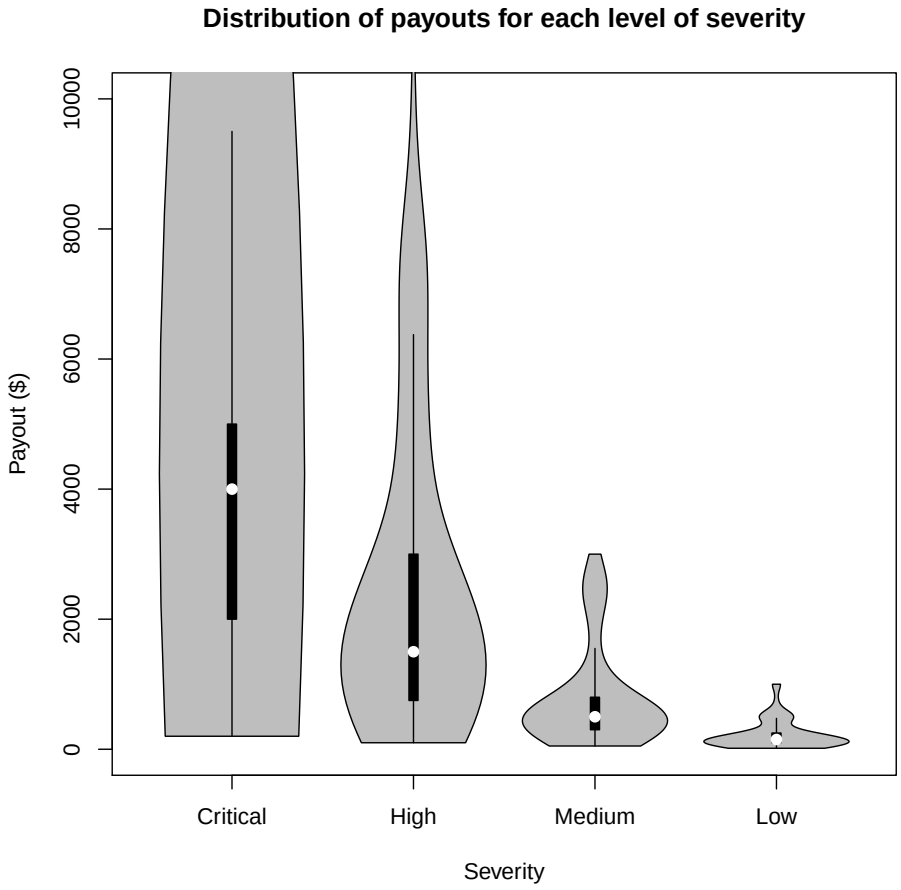

Figure 3.2: Distribution of the payouts for each level of vulnerability severity. The maximum value for critical is $200,000 and for high is $15,000.

payout for a critical vulnerability is $7,924 with a maximum of $200,000. Having an above average proportion of critical vulnerabilities can increase the operating cost significantly.

3.5 Expected benefits to operation

3.5.1 Average reporting statistics

On average, a programme can expect 0.429 new valid report submissions each day and 156 valid reports over the course of a year. Newly-released programmes can expect to see as many as 20 new valid reports per day. This may be due to hackers switching to new programmes, hoping to find the easy initial vulnerabilities.

With an average of 156 reports per year, an organisation can expect 13 critical vulnerabilities to be discovered. The relative proportion of each severity is shown in Table 3.5. Organisations are most severely impacted by critical vulnerabilities, which include unauthorized access, remote code execution, and privilege escalation [118].

3.5.2 Analysis

Finifter et al. [75] found that Firefox and Google received 63 and 167 vulnerability reports respectively per year. These rates are comparable to current rates seen on HackerOne and Bugcrowd.

Zhao et al. [271] reported that HackerOne saw 400 vulnerability reports per month in 2015. This has now increased to over 2,500 vulnerability reports per month across all programmes. This increase could be due to an increase in the number of hackers participating in programmes. HackerOne now boasts over 200,000 hackers [97].

Newly-released programmes can expect an initial spike in the number of reports submitted as hackers use automated testing to find vulnerabilities caused by cross-site scripting or SQL injection. The initial influx of vulnerability reports was raised as a concern by programme operators in Section 2.5, as the high number of low-quality reports, particularly from automated testing tools, can overwhelm the response team.

As the number of programmes available on bug bounty platforms increases, the number of hackers may continue to grow. This may result in the average number of vulnerabilities found per year increasing. The network externalities that exist surround bug bounty platforms are discussed further in Chapter 4.

3.6 Programme activity

3.6.1 Lifetime activity

The total number of reports submitted each day after launch, for all programmes over 100 days old, is shown in Figure 3.3. As shown on the graph, the first day of programme operation sees the greatest number of report submissions. The aggregate volume of reports submitted in the

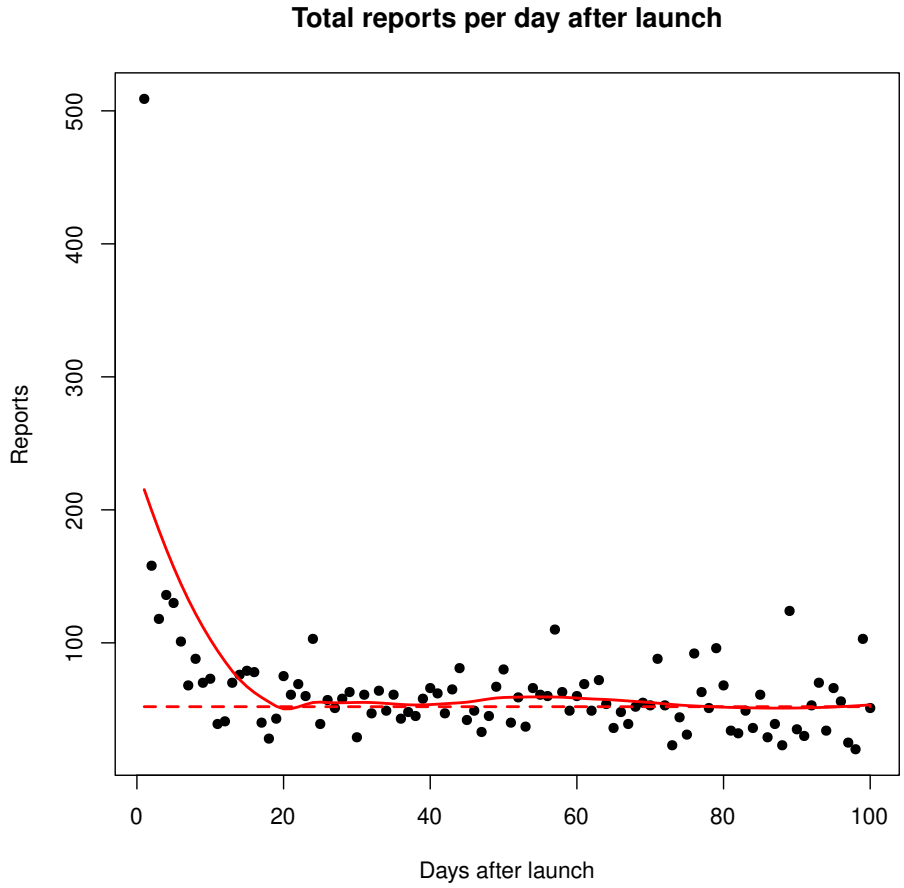

Figure 3.3: The total number of reports submitted for all programmes over 100 days old, with local regression line (50% smoothing) and line at $y = 52$.

following days decreases from the initial spike until around 20 days after programme launch, after which a fairly constant volume of reports is submitted to all programmes at the rate of approximately 52 reports per day.

The graph shown in Figure 3.4 displays the trend in the rate of report submissions for the first year after launch of a programme, for all programmes on HackerOne. A 5-day rate of submission moving average is calculated for each programme and normalised against the long-term submission rates (the total number of reports divided by the age of a programme). This allows for a comparison to be made between programmes that have achieved varying levels of success on the bug bounty platform, and as a result have different average rates of submission. The 5-day rate of 183 programmes (those at least one year old) is then averaged. The first

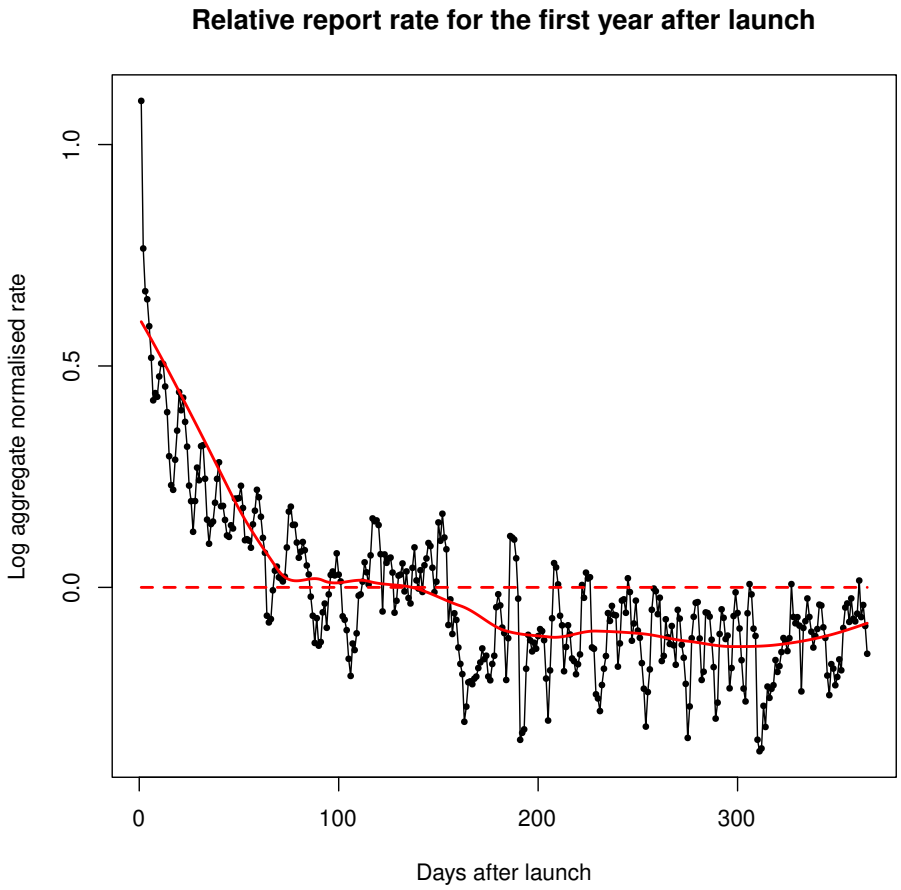

Figure 3.4: The log aggregate normalised reporting rate for all programmes on HackerOne over one year old, with local regression line (50% smoothing) and line at $y = 0$.

data point on the graph represents the increase in submission rate averaged over the first five days after launch. The initial rate is approximately 12.55 times greater than the average rate of submission over the lifetime of a programme (represented by the dashed line).

The normalised aggregate reporting rate for all programmes over 3 years old (93 programmes) is shown in Figure 3.5. In order to investigate the long-term rate, data from the first 100 days has been removed, as this period sees above average numbers of reports being submitted. Linear regression is performed in order to see how the rate of reporting changes after the initial launch period. A best fit line of $y = 3.28 \times 10^{-5}x + 0.882$ is fitted to the data. While a linear model is a very poor fit for the data ($R^2 < 0.01$), it does allow the average rate over time to be investigated. In the case of long-term programme reporting rates, a gradient

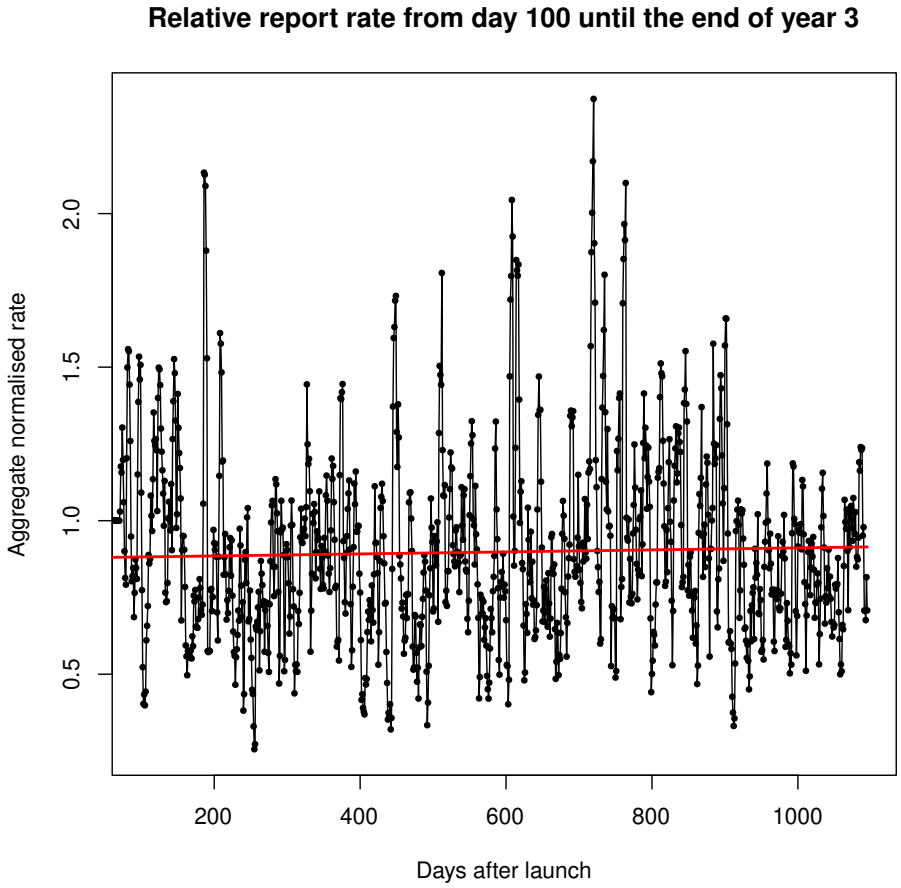

Figure 3.5: The aggregate normalised reporting rate for all programmes on HackerOne over 3 years old, with linear regression line at $y = 3.28 \times 10^{-5}x + 0.882$.

of 3.28×10^{-5} suggests that, on average, the rate is nearly constant. The intercept of 0.882 represents the expected rate relative to the long-term rate of reporting.

Investigating activity over the lifetime of a programme shows that, after an initial influx of vulnerability reports, the average rate of submissions decreases to an almost constant level.

3.6.2 Analysis

The results show that a programme sees the highest rate of report submissions in a short period after launch, with rates up to 12.55 times greater than the long-term average rate of submission. The high rate of vulnerability discovery and reporting after the launch of a programme, particularly on the day after launch, may be attributed to hackers switching their

focus to finding easy vulnerabilities in the programme [157]. This is consistent with the results presented in Chapter 2.

From Figure 3.4, the period of elevated activity is approximately 120 days, after which the average rate drops below the long-term average. After this initial period, the rate of submissions remains, on average, almost constant over the lifetime of a programme; such a trend was also observed by Finifter et al. [75].

When preparing to launch a bug bounty programme, operators should account for the high volume of submissions just after launch. Provisions should be made to ensure that all reports are addressed in a timely manner, so as to avoid irritating hackers awaiting a response and potentially discouraging from further interaction with the programme. Such considerations were also recommended by programme operators — as discussed in Section 2.8.

Future research efforts should endeavour to further examine bug bounty programme operating characteristics using the wealth of academic literature surrounding the topics of vulnerability discovery rates and software reliability growth. Early empirical work by Adams explored the discovery patterns in IBM software [4]. Ozment and Schechter studied the vulnerabilities present in OpenBSD, finding the median lifetime of foundational vulnerabilities (those present from the start of the project) to be 2.6 years [182]. Brady et al. further explore reliability growth models in complex adaptive systems from a theoretic perspective [37], and Anderson gives consideration for the added difficulty faced by testers that operate without access to the underlying source code, a factor particularly applicable when considering bug bounty programmes that only permit black-box testing [17].

3.7 Effect of bounty payouts

3.7.1 Relationship between total programme payouts and received reports

Figure 3.6 shows the relationship between the average programme bounty and the number of reports submitted. Linear regression is used to plot a trendline with an R^2 value of 0.0527. This suggests that linearly increasing the average bounty will not result in proportionally more reports. There is a weak relationship between the average bounty of a programme and the

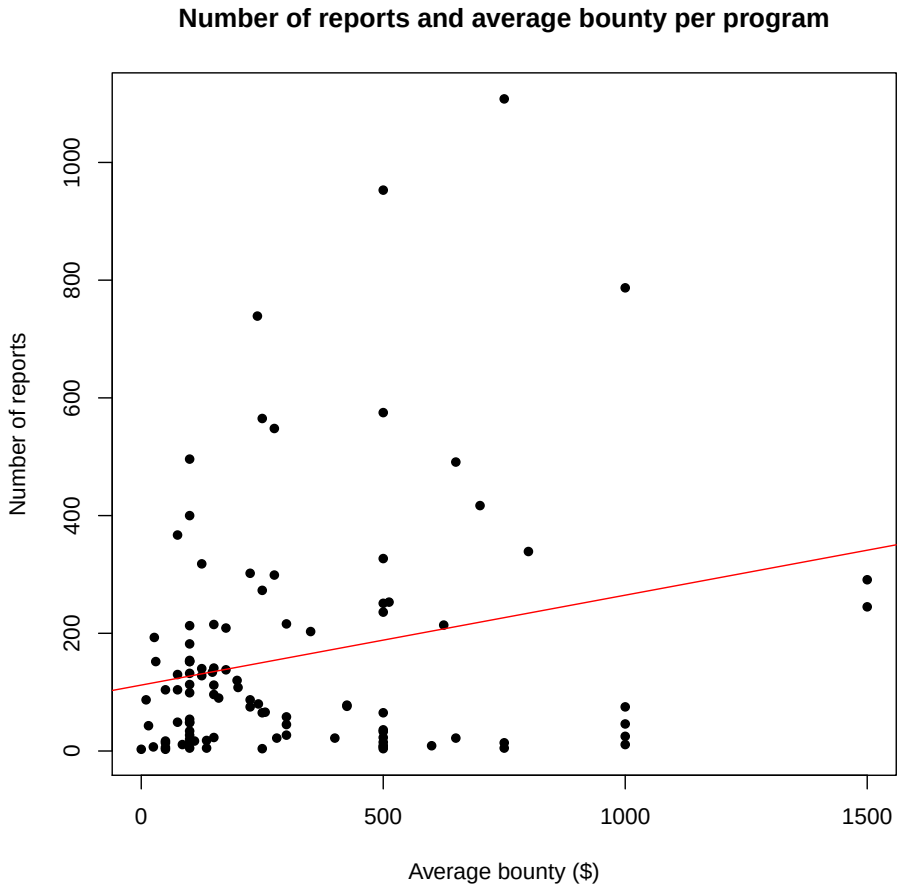

Figure 3.6: The number of reports for each programme plotted against the average bounty payout.

number of reports submitted, which is evidenced by the low correlation coefficient of 0.02295.

An interesting observation is that many hackers are willing to work without the promise of monetary rewards. They are instead motivated by reputation on the website, which is needed to be invited to private programmes. Table 3.5 shows that 9.6% of all reports submitted do not result in a bounty payout. Algarni et at. [13] found that, among the top users on the site, monetary rewards are still the primary motivation.

3.7.2 Analysis

There is a weak correlation between the success of a programme and the average bounty payout. Hackers value the challenge or opportunity to learn almost as much as the bounty payouts when

selecting programmes in which to participate [98]. This may explain the success of programmes with low or no bounty payouts. Finifter et al. [75] also found similar results. Mozilla had a far greater average bounty payouts than Google, but found much fewer vulnerabilities [271].

The non-linear relationship between number of reports and average bounty makes it difficult to model the effect of increasing the average bounty of a programme; however, the weak correlation suggests there may be some benefit to increasing the average bounty.

The average bounty for all programmes is $318. This can be used as a guideline for new programmes when setting a rewards structure.

3.8 Variation in programme availability and hacker participation

3.8.1 Programme availability

Figure 3.7 shows the number of programmes launched each month on HackerOne from late 2013 until the beginning of 2019. The yearly total for the number of programmes launched is shown in Figure 3.8. Growth from 2013 to 2017 steadily increased with approximately five additional programmes launched per year on top of the previous year's total. 2018 saw the highest overall growth with 10 more programmes launched that year than 2017.

3.8.2 Hacker participation

HackerOne uses reputation to measure the validity of reports submitted. Users with high reputation submit a large number of valid reports. Figure 3.10 shows the distribution of the reputation for the top 100 users, with these users being responsible for 40,690 vulnerability submissions out of almost 72,000. The distribution of reputation and number of vulnerabilities reported shows that a small number of users submit significantly more reports than the average user. Similar results are found for hackers on Bugcrowd in Chapter 4. The top five users of HackerOne have submitted a combined total of 7,121 vulnerability reports, 17.5% of vulnerabilities submitted by the top 100, and almost 10% of all vulnerabilities submitted to HackerOne.

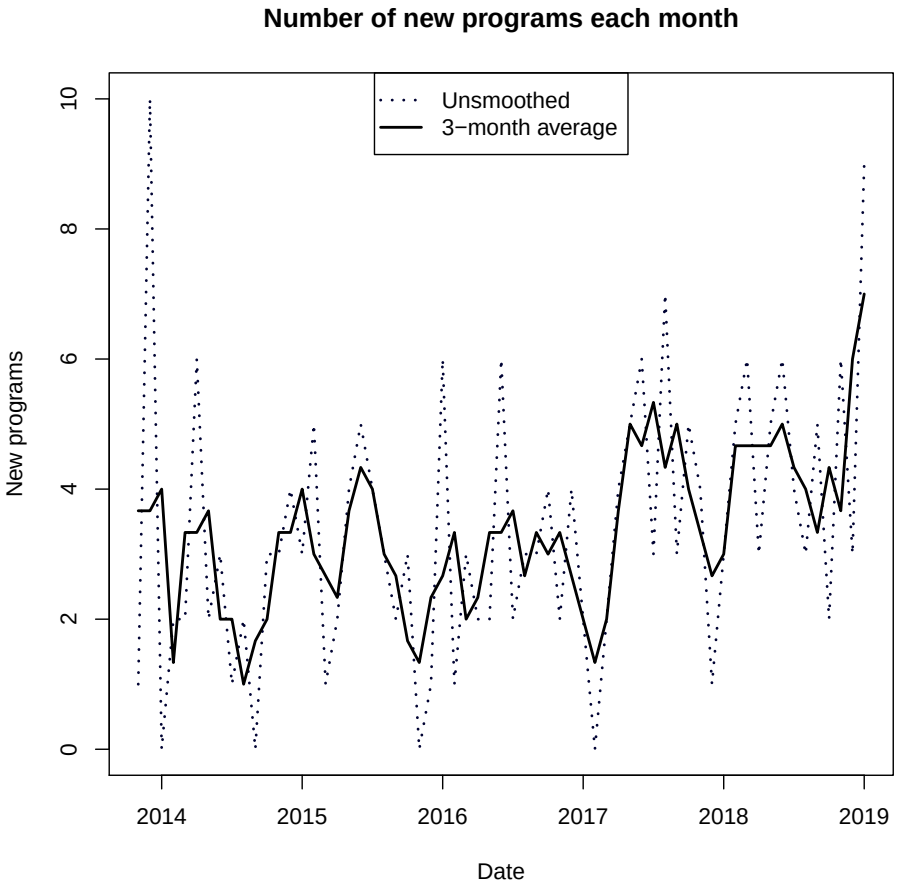

Figure 3.7: Number of new programmes on the site HackerOne from 2013 to 2018.

Bugcrowd provides detailed user activity information. It can be seen that the top 10 hackers have contributed to 118 programmes on average. The distribution of the number of hackers participating in each programme is shown in Figure 3.9.

3.8.3 Programme responses

HackerOne suggests a standard response for organisations to adhere to when responding to hacker-submitted reports. This includes a limit of five days to respond to a report and 10 days to triage [106]. The distribution of responses meeting the standards is shown in Figure 3.11. On average, an organisation will meet the standard in 97% of responses. However five organisations meet this target less than 20% of the time.

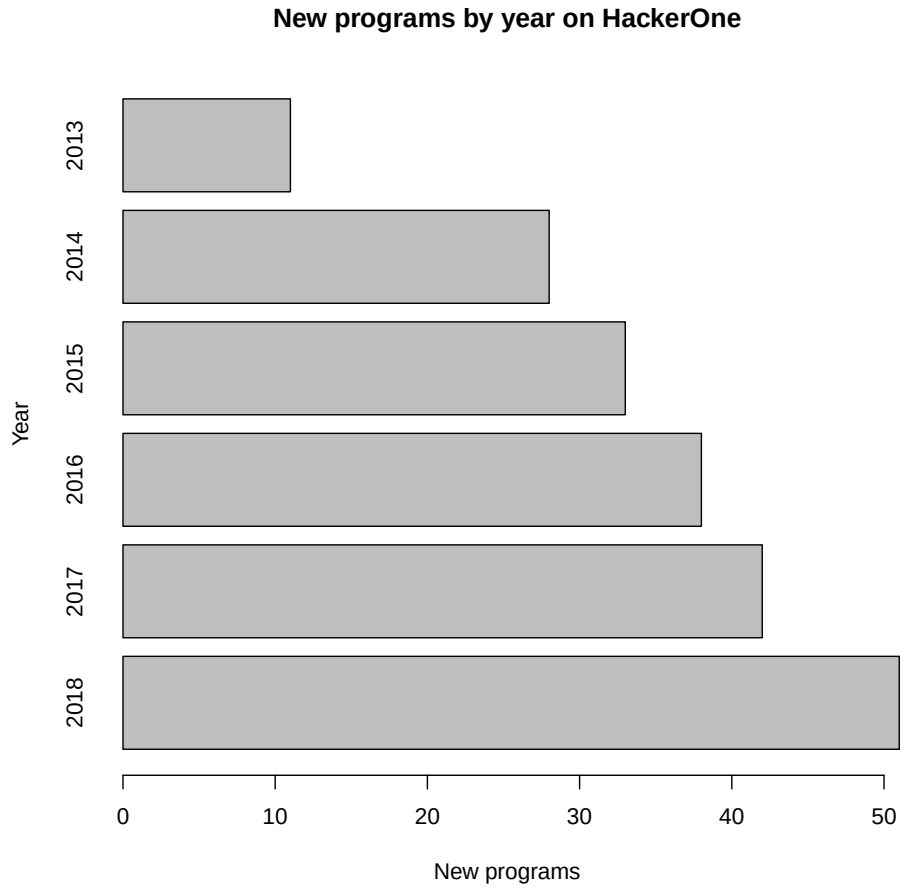

Figure 3.8: Number of new programmes launched each year on the site HackerOne from 2013 to 2018.

3.8.4 Analysis

Since the study by Zhao et al. [271], HackerOne has seen continual growth in the number of programmes available and the number of hackers participating. This demonstrates the viability of bug bounty programmes to provide benefit to both organisation and hackers. However, the participation of hackers on Bugcrowd and HackerOne is very skewed, with the top 100 users contributing a significant proportion of overall reports. This was also found by Zhao et al. [271], with each hacker in the top 100 submitting 147 reports on average. The top 100 have now submitted 406 reports on average. Despite the highly skewed nature of the participation metrics, as discussed in Section 2.7.2, operators in also highlighted the value that the top hackers can bring to programme.

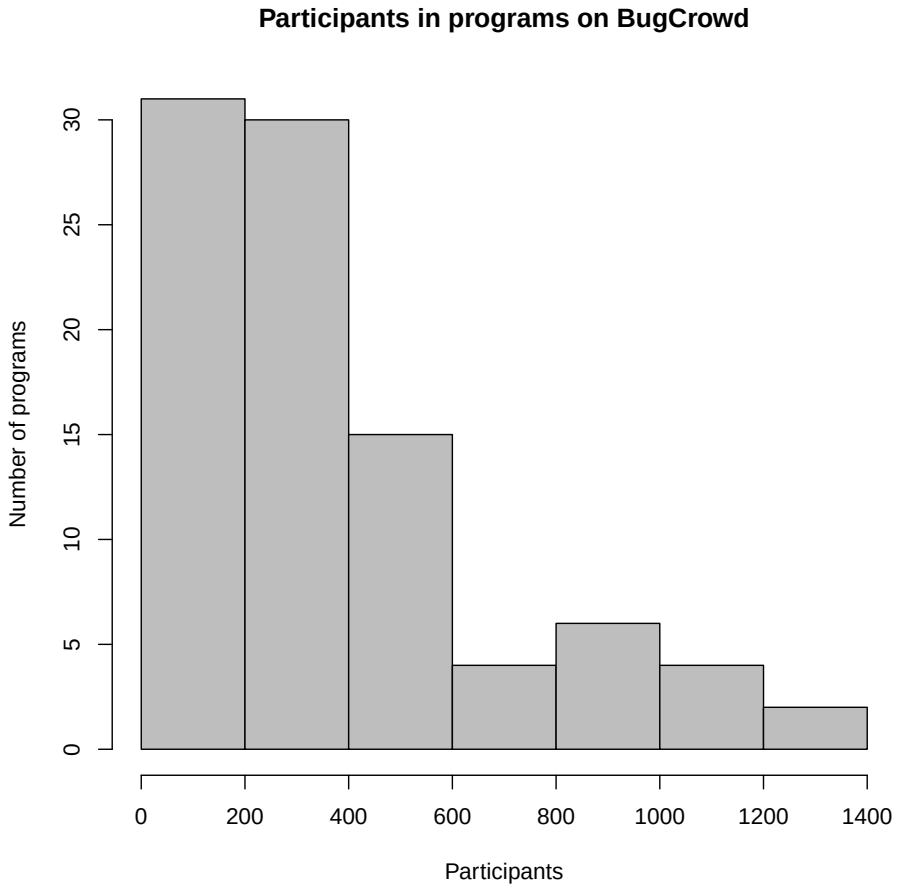

Figure 3.9: Number of participants for each programme on Bugcrowd.

Organisations meet the standard response guidelines set out by HackerOne in 97% of responses. Adhering to these guidelines ensures that hackers receive a reward promptly and organisations are able to mitigate any vulnerabilities identified. Large volumes of poor quality or invalid reports will reduce the ability of a organisation to meet the standards. A theoretical framework introduced by Laszka et al. [144] aims to reduce the numbers of these reports.

3.9 Background of organisations

3.9.1 SMEs and large enterprise

Analysing the range of businesses that use bug bounty platforms allows for any bias towards businesses of a certain size or sector to be identified.

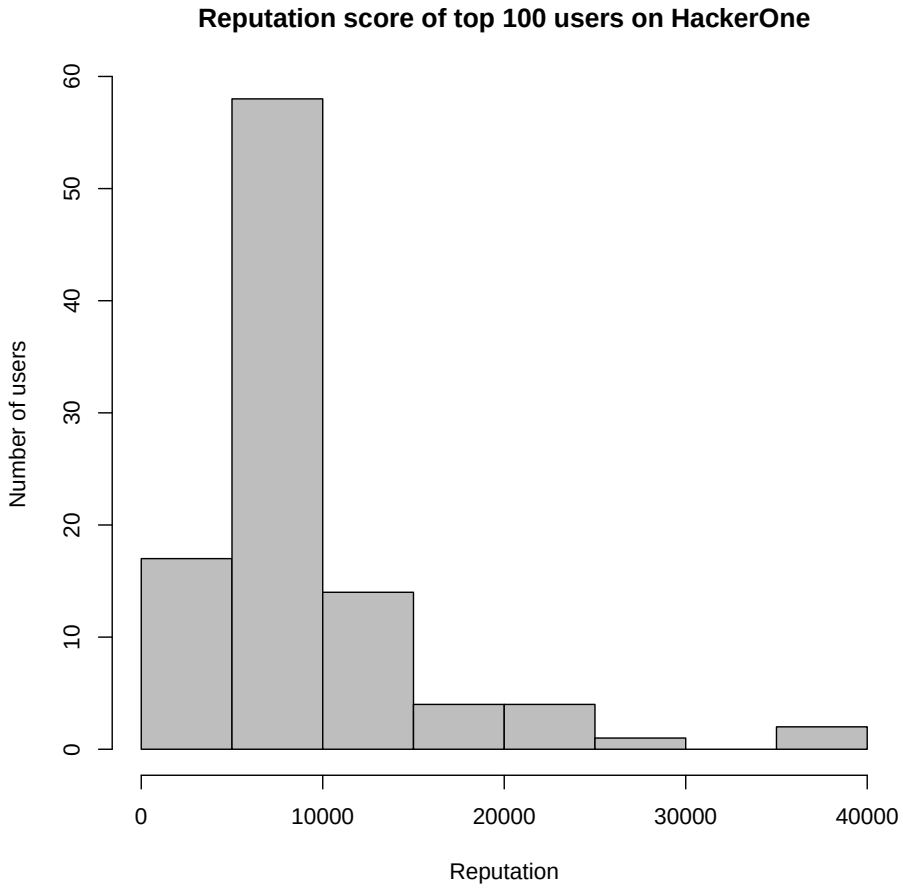

Figure 3.10: Reputation of users on HackerOne.

Publicly available data collected from Morningstar[2], Crunchbase[3], Companies House[4], and through information filed with the US Securities and Exchange Commission (SEC)[5], allows for each organisation to be categorised using the Small and Medium Enterprise (SME) and large business definitions of [230]. From the organisations listed on HackerOne and Bugcrowd, 142 can be categorised into the following sizes of business: micro, small, medium and large; the results of this are shown in Table 3.6.

[2] `https://www.morningstar.com`
[3] `https://www.crunchbase.com`
[4] `https://www.gov.uk/get-information-about-a-company`
[5] `https://www.sec.gov/edgar/search-and-access`

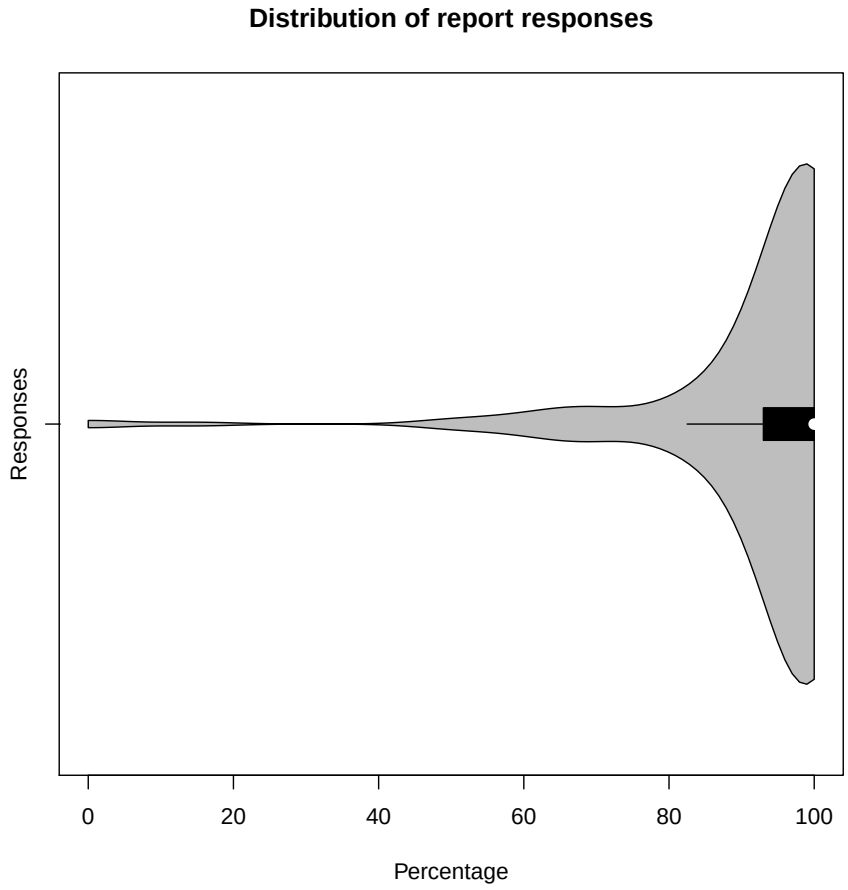

Figure 3.11: Distribution of the percentage of report responses that meet the standards set by HackerOne.

3.9.2 Primary business activity

From the list of organisations, 193 can be placed into 18 pre-defined categories. This is shown in Figure 3.12. The remaining businesses could not be categorised due to insufficient information. Software development companies and Open-source projects account for 41% of programmes listed on HackerOne and Bugcrowd.

3.9.3 Analysis

Although small and medium businesses do not typically have the same resources available as large organisations for the purposes of security, SMEs make up 50% of businesses using bug bounty platforms (compared to 99.3% of businesses in the UK [228]). This demonstrates that

Size	Number
Large	64
Medium	32
Small	39
Micro	7

Table 3.6: Sizes of businesses on bug bounty platforms.

		In-scope	
		Explicit	Implicit
Out-of-scope	Explicit	Exhaustive lists of all in- and out-of-scope assets	Exhaustive list of only the out-of-scope assets
	Implicit	Exhaustive list of only the in-scope assets	Non-exhaustive or no lists of assets defined as in- or out-of-scope

Table 3.7: Definitions for in- and out-of-scope assets.

operating a bug bounty programme may be viable for a business, regardless of size.

As shown in Figure 3.12, there is a wide range of sectors currently operating bug bounty programmes. Open-source projects make up 18.1% of listed programmes on bug bounty platforms, with many of these projects being funded by governments in order to improve the security of popular open-source software. The success of the European Commission's Directorate-General for Informatics (DIGIT) funded programmes has the potential to lead to an increase in the number of open-source programmes [229], and lead to positive externalities for organisations and individuals making use of the open-source software [122].

3.10 Programme in-scope assets

3.10.1 Programme assets

Programmes on HackerOne often explicitly state the assets that they wish hackers to explore in the search for vulnerabilities, these are listed on the scope section of a programme page. There exist several ways in which a programme may declare which assets are in- and out-of-scope for hackers. These are summarised in Table 3.7.

For the results presented in this section, only explicitly stated in-scope assets will be considered. From each programme on HackerOne, the assets were recorded from the scope section

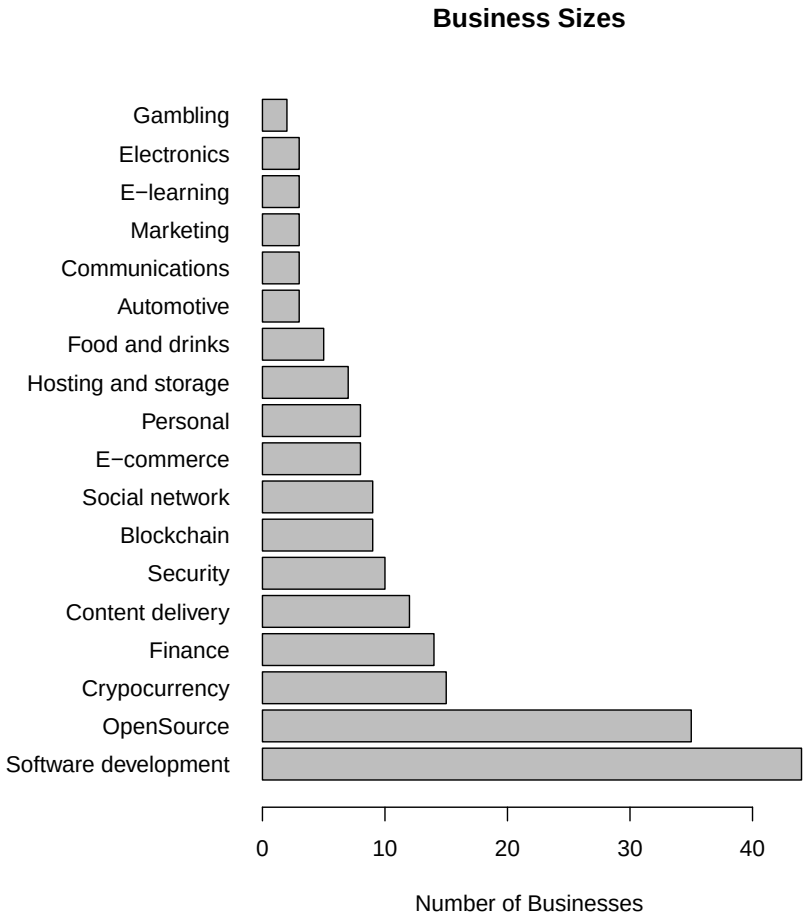

Figure 3.12: Types of businesses on bug bounty platforms.

and the body of the programme description. For the purpose of this research, the 12 types of asset listed on HackerOne are condensed into the following categories:

- Classless inter-domain routing (CIDR)
- Domains and APIs
- Mobile applications, consisting of 'iOS: App Store', 'iOS: Testflight', 'iOS: .ipa', 'Android:Play Store', and 'Android: .apk' assets. A total over 183 iOS assets (over 76 programmes) and 188 Android assets (over 81 programmes) were identified.
- Desktop applications, consisting of 'Windows: Microsoft Store' and 'Executable' assets, covering Windows, Mac, and Linux operating systems.

Type of asset	Total	Percentage of total assets
Domain/API	1,313	56.6%
Source code	441	19.0%
Mobile application	371	16.0%
Desktop application	141	6.0%
Hardware	25	1.1%
CIDR	19	0.8%
Other	11	0.5%

Table 3.8: Total of each asset type in HackerOne programmes.

	Programmes with asset	
Type of asset	#	%
Domain/API	154	75.1%
Source code	90	43.9%
Mobile application	81	39.5%
Desktop application	43	21.0%
Hardware	13	6.3%
CIDR	5	2.4%
Other	7	3.4%

Table 3.9: The number of programmes (from total of 205) that have at least one occurrence of an asset in scope.

- Source code
- Hardware/IoT
- Other, a category used when no other asset category applies, for example 'Only use this asset when nothing else can be reasonably selected'. These assets are ignored within this research.

A total of 2,321 assets were recorded over 205 programmes; this is summarised in Table 3.8. The coverage of programmes for each asset type is shown in Table 3.9.

The distribution of the total number of assets per programme is shown in Figure 3.13. Several outliers are not shown due to the very large number of assets: Nextcloud (192), Spotify (187), Node.js third-party modules (160), and Verizon (159). On average, a programme has 8.87 in scope assets listed on the programme page. Shown in Figure 3.14 is the distribution of each asset type per programme.

It is found that there is a weak correlation between the total number of reports a programme

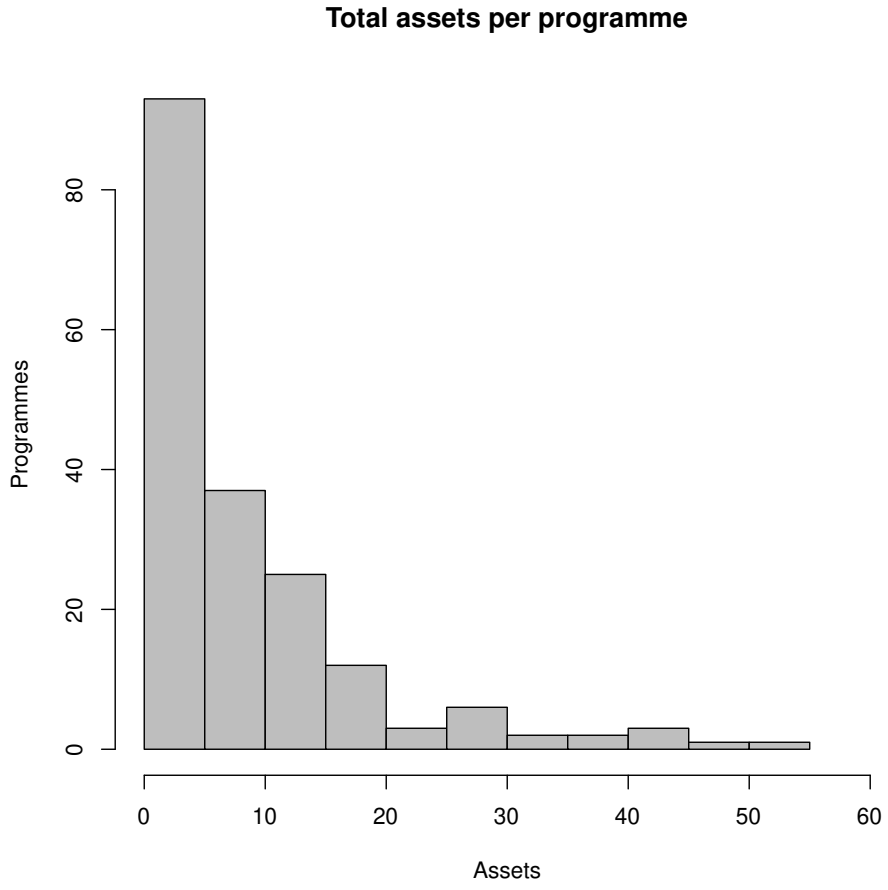

Figure 3.13: The number of assets included in scope for each programme on HackerOne.

receives, and the number of in scope assets (Kendall $\tau = 0.343$, Spearman $\rho = 0.470$; both significant at a 95% confidence level). Linear regression gives a model of $y = 15.8x + 35.5$, with y being the total number of reports and x the number of assets in scope. However the poor fit of this model ($R^2 = 0.226$) indicates a more complex relationship between the two variables.

Multiple linear regression is performed in order to assess the impact each type of asset has on the number of reports received for a programme. This is done in order to quantify the utilisation of each type of asset by hackers in order to find vulnerabilities. Data from 185 programmes is used to build the model, a summary of the assets used is shown in Table 3.10. The form of the regression model is shown in Equation 3.1.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 \tag{3.1}$$

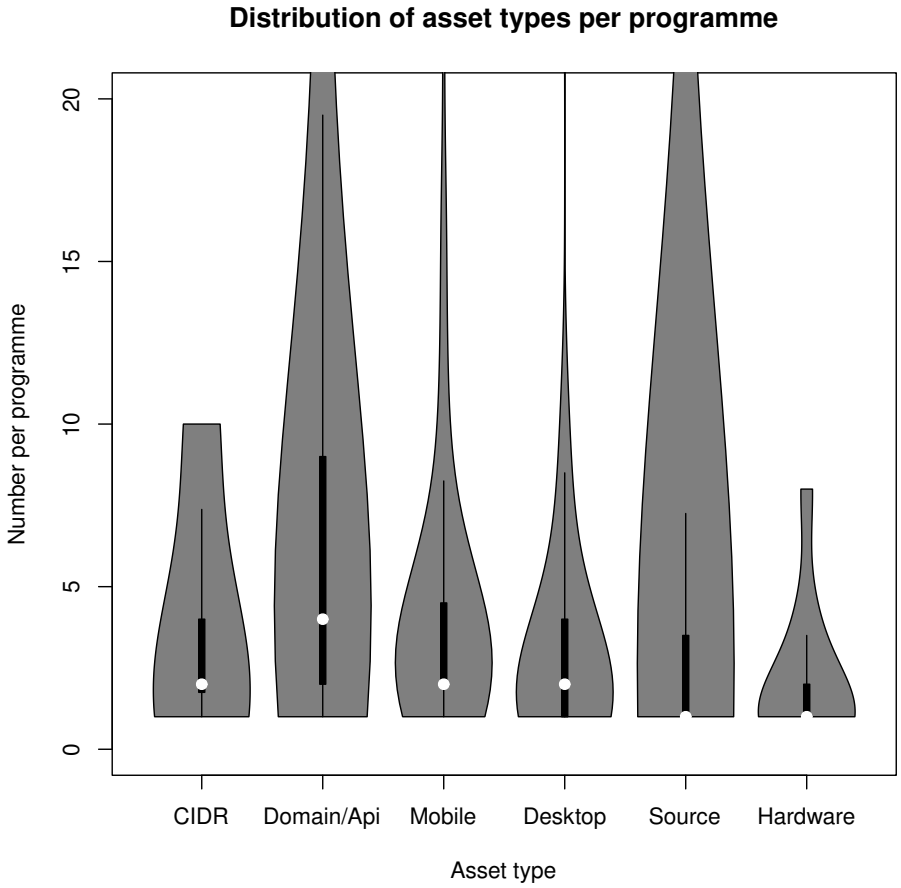

Figure 3.14: Number of each type of asset with at least one occurrence of the asset in scope for programmes on HackerOne.

The total number of reports is represented by y. The regression coefficients are β, with values shown in Table 3.11. The number of assets of each type, for a given programme, is represented by $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. These correspond to: CIDR, domains/APIs, mobile applications, desktop applications, source code assets, and hardware assets. The R^2 value for the regression is 0.360, this shows that the current linear model does not represent the data particularly well, however qualitative observations can still be made.

- CIDR assets make up only 15 assets (within four programmes) out of 1,641 assets used in the regression model. The large negative coefficient demonstrates that programmes that include CIDR assets in-scope see less total reports overall. However the small number of observations threaten the validity of this particular result.

Type of asset	Total
Domain/API	937
Mobile application	305
Source code	223
Desktop application	130
Hardware	22
CIDR	15

Table 3.10: Total of each asset type used in the multiple linear regression model.

Intercept	CIDR	Domain/API	Mobile	Desktop	Source code	Hardware
β_0	β_1	β_2	β_3	β_4	β_5	β_6
50.6	-21.9	8.80	47.9	6.25	-4.23	31.8

Table 3.11: Multiple linear regression coefficients, shown to 3 significant figures.

- Additional web domains and APIs contribute to the total number of vulnerabilities found in a programme, as shown by the positive coefficient.
- The large positive coefficient suggests that programmes that include mobile applications in-scope see a greater number of vulnerabilities reported.
- Similar to mobile applications, the inclusion of desktop applications contributes positively to the total number of vulnerabilities reported, although not to the same extent.
- Source code assets have a negative coefficient, suggesting that increasing the number of source code assets in-scope reduces the overall effectiveness of the programme.
- The large positive coefficient suggests that the inclusion of hardware assets is beneficial for a programme, however the small number of hardware assets used in the model threaten the validity.

3.10.2 Analysis

Investigating the number of assets included as part of a programme shows a weak correlation between the number of valid reports received and the total number of assets. The poor fit of the linear model suggests that dividing the attention of hackers over many assets does not always result in the returns one may expect. Creating a multiple linear regression model allows for the impact of each type of asset to be considered. Notably, the results show that

source code assets perform poorly. This may highlight the unwillingness of hackers to spend time manually searching through code to find vulnerability, especially as many programmes prevent the submission of vulnerabilities found through automated testing, making the search for vulnerabilities in source code an unappealing endeavour. It is important to note that the observations are purely correlative and do not imply causality.

3.11 Causes of programme closure

3.11.1 Former programmes

A list of public programmes that are inactive or defunct is available on HackerOne. As of April 2020 there are 103 programmes no longer operating, with the reasons for ceasing operation being shown in Figure 3.15.

For 36 of the former programmes, there was no publicly available information as to the reason behind the shut down of the bug bounty programme. Of these organisations formerly operating a programme, 30 are still active and a further three are active open source projects. The remaining three organisations appear to be inactive, however it is not possible to confirm whether the organisation has shut down.

It is found that 34 organisations have shut down, and as a result their bug bounty programme has been suspended indefinitely. Some organisations will notify the participants of their programme that both the organisation and the programme will be shutting down. However, a majority of organisations give no indication to hackers that further submissions will go unanswered.

Shown in Table 3.12 are the six programmes formerly operated on HackerOne as part of the EU-FOSSA project. These limited duration programmes resulted in the discovery of 89 vulnerabilities in three popular open source projects and awarded over $70,000 to hackers. In addition to the EU funded programmes, the other limited duration programmes operated were competitions and the Flash Internet Bug Bounty (IBB).

The acquisition of an organisation resulted in the closure of eight bug bounty programmes. A further eight programmes were merged into the programme of a parent organisation or a

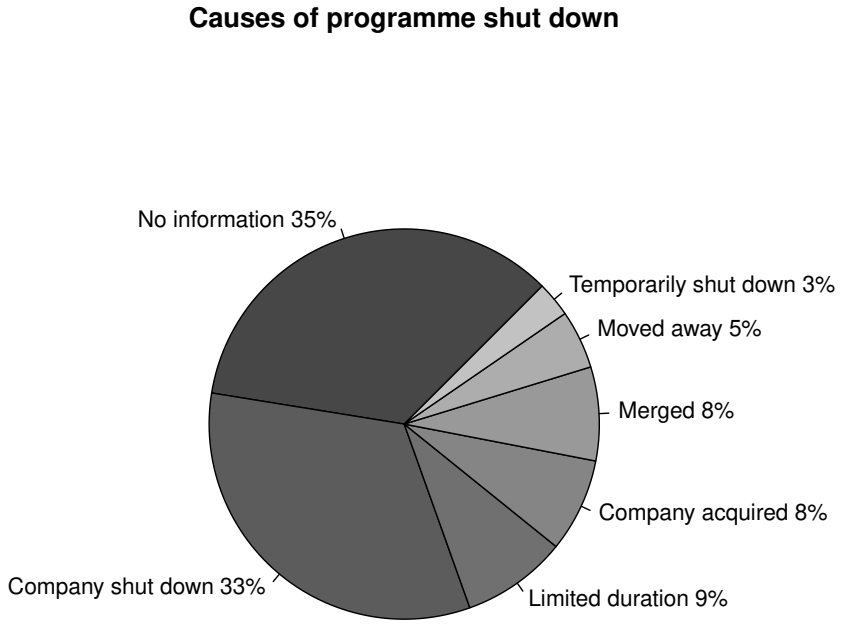

Figure 3.15: Causes of bug bounty programmes shutting down on HackerOne.

parent programme.

Public bug bounty programmes were not seen as a viable option for five organisations. Four organisations publicly announced that they would be moving to a responsible disclosure programme, and were not offering advertising any further bounty payouts. As of January 2020, the public programme operated by Alibaba closed, and moved to an invite only private programme.

The volume of submissions received has caused three programmes to temporarily shut down. However external factors, such as the COVID-19 pandemic, may be attributed to the shut down of these programmes. The effect of the pandemic has seen other programmes (e.g., Priceline) reduce their bounty payouts.

3.11.2 Analysis

The lack of information surrounding the 35% of programmes that shut down without any indication as to the reason behind the closure acts as a barrier to identifying the problems encountered during programme operation. The limited data available for these programmes

Programme	Reports resolved	Bounty payout
VLC	41	$34,435
PuTTY	34	$27,954
Midpoint	14	$8,330
Notepad++	0	$0
FileZilla	0	$0
Apache Kafka	NA	NA

Table 3.12: Programmes funded by European Commission's Directorate General for Informatics (DIGIT).

shows that each of the programmes successfully received reports from hackers, each receiving 42 valid reports on average. Suggesting it was not lack of activity that caused the programmes to close.

For a majority of former programmes, the factors causing the shut down of a programme can be identified, and the primary cause being attributed to an organisation becoming insolvent. A minority of programmes (8%) have closed due to the public operation of a programme being inappropriate for the current requirements of an organisation.

The date that the last report was resolved indicates the date that the programme shut down. From 2015 to 2019, an average of 15 programmes shut down per year. In the first four months of 2020, 18 programmes shut down. There is currently insufficient data to ascertain the cause of a sharp increase in the programme closures in 2020.

3.12 Summary of findings

Motivated by the question *Is it beneficial for a company to make use of bug bounty programmes, instead of hiring additional software engineers?*, we have considered the operational characteristics of bug bounty programmes hosted on the HackerOne and Bugcrowd platforms, and reconsidered the findings of earlier related work from Finifter et al. [75] and Zhao et al. [271]. A summary of the findings for each research question listed in Section 3.2 is as follows.

1. The average annual cost of running a bug bounty programme is now $83,815. Analysis of existing programmes has shown that the average cost of operating a programme for a year is now less than the cost of hiring two additional software engineers.

2. After an initial spike in vulnerability reports following the launch of a programme, an average of 156 vulnerabilities per year can be expected to be found by hackers. Collectively hackers are able to use their diverse skill set in order to find and report vulnerabilities of all severity levels.

3. A fairly constant rate of vulnerability discovery can be expected soon after the launch of a programme. However, further work is needed to determine the impact of changing bounty payouts on the rate of discovery.

4. Hackers are incentivised to participate in programmes with low or no bounty payouts in order to build up their reputation. As a result there is a weak correlation between the number of reports received and the average bounty payout.

As noted earlier, the costs described throughout this chapter were correct when the analysis was conducted in late 2019.

Without quantifying the productivity of software engineers, the usefulness of comparing the cost of employing additional software engineers to the cost of operating a bug bounty programme is limited. However, organisations will be able to compare the efficiency of their software engineers with the results detailed within this chapter. The productivity of software engineers can be measured by, for example, the number of vulnerabilities they find per year, the chance to detect vulnerabilities, or the average number of vulnerabilities they introduce per line of code written.

The previous experience and education of an employee will influence their effectiveness at finding or preventing the existence of certain types of vulnerability. As such, we have to be cautious when comparing the 'average' software engineer with the 'average' hacker. In comparison, bug bounty programmes are open to hackers with diverse technical backgrounds, who can find vulnerabilities beyond the scope of traditional software engineering skill sets. For organisations with technical skill gaps, there is an additional benefit to the search for vulnerabilities via bug bounty programmes; however, this is not quantified here.

It is also important to consider the applicability of the results in light of the additional costs that may come with hosting a programme on a bug bounty platform. Although the

payments made to hackers via bounties may be quantifiable, the fees charged by platforms are often opaque and precluded by non-disclosure agreements that exist within service agreements. As noted by one operator in Section 2.7.2, if fees are considered, the cost-effectiveness of a programme can be significantly diminished.

In the next chapter we look further into the behaviour of hackers on bug bounty platforms, and expand upon the insights concerning the high-skewed productivity of hackers. For programmes hosted on platforms to continue to be cost-effective, platforms must retain their highly productive hackers, and continue to attract inquisitive hackers to search for vulnerabilities.

Chapter 4

Understanding the Behaviour of Hackers

After the investigation of the current state of bug bounty programmes in Chapter 3, it is prudent to further consider aspects that impact programme operation. As such, the dynamics of platforms and the behaviour of their populations of hackers are considered in this chapter, helping to answer the research question *How effective are bug bounty platforms at retaining the interest of hackers?*

In Chapter 2 and in an earlier study by Al-Banna et al. [9], the operators of bug bounty programmes repeatedly expressed their concerns over the motivation (or lack thereof) exhibited by hackers based on platforms. As highlighted in Chapter 3, high-productivity hackers contribute disproportionately to the success of many programmes, with just the top 100 hackers on the HackerOne platform discovering a majority of the reported vulnerabilities. As discussed in Section 2.7.2, operators emphasised the benefits that top hackers can bring to a programme, and noted the value of information delivered in reports in contrast to many hackers that are simply looking for 'quick results'. The lack of motivation of top hackers has far-reaching consequences both for the success of bug bounty programmes and for the organisations operating bug bounty platforms.

A behavioural study of participants has the potential to indicate the platform-wide motivation of hackers when searching for vulnerabilities to organisations looking to operate bug

bounty programmes; it also has the potential to provide additional insight to organisations already operating such programmes.

Section 4.1 presents a survey of literature concerning the behaviour of hackers, with a particular focus on issues surrounding sources of motivation, the influence of market forces, and a general discussion of hacker productivity. This is followed by an outline of the motivation for further understanding the behaviour of hackers on bug bounty platforms in Section 4.2, in which pertinent research questions for the chapter are presented. While a portion of the data collected as part of the investigation into the current state of bug bounties (Chapter 3) may have been used to comment on the general characteristics of a sub-population of hackers, Section 4.3 presents a distinct source of behavioural data used throughout the modelling process introduced in Section 4.4. A consideration of the number of hackers actively engaged in the search for vulnerabilities is presented in Section 4.5. The archetypal behaviours exhibited by hackers are explored in Section 4.6. Broader issues concerning the viability of participation for hackers in the search for vulnerabilities are discussed in Section 4.7. Finally, the chapter concludes in Section 4.8 with a presentation of the key findings and a brief discussion of their limitations. The work presented in this chapter is based on [256].

4.1 Surveying views of hacker behaviour

4.1.1 Motivation

There are several theories as to the primary factors that motivate hackers to search for vulnerabilities. The relationship between monetary compensation and intangible rewards is discussed consistently within the literature.

Malladi and Subramanian [158] argue that "financial compensation is the sole motivation for researchers" and that intangible rewards such as reputation are ineffective. Algarni and Malaiya [13] note that, for top hackers (i.e. those with a high number of submissions), monetary rewards are a key incentive to hackers.

Results from the HackerOne 2019 *Hacker Report* [99] reveal that 14.3% of those surveyed reported that "to make money" was their primary motivation for participation. Conversely,

Metric	HackerOne	Bugcrowd
Number of programmes	326	133
Number of users	>600,000 [100]	>200,000
Total bugs found	>150,000 [100]	>26,000 (2019)
Est. total payout	>$80M [100]	>$6.5M (2018)
Total funding	$110.4M [58]	$78.7M [56]

Table 4.1: Comparison between the HackerOne and Bugcrowd platforms. Data current as of 10^{th} February 2021, or as otherwise specified.

56.7% of hackers stated that non-monetary motivating factors, such as the potential to learn, were their primary motivation. Approximately 28% reported altruistic reasons for hacking. However, it could be argued that this figure is over-inflated due to social-desirability biases.

The success of programmes and full-disclosure platforms without any monetary reward suggests that hackers often value public recognition and reputation above immediate financial gain. As discussed in Chapter 3, the defunct platform Wooyun was studied by Zhao et al. [270], who found that hackers found thousands of vulnerabilities in the absence of financial motivation. Prestige as a motivator may emerge from a high reputation score, featuring in a "Hall of Fame", or being named in a public disclosure — all of which are used as rewards [271].

Votipka et al. [252] identified three motivators for hackers when selecting programmes and areas on which to work: potential monetary payouts, value of target asset to an organisation (finding vulnerabilities in an important asset), and non-monetary rewards (enjoyment from the task, altruism and pride). With observations of the previous payouts awarded to hackers and with an estimation of the potential monetary payouts, geographic factors and local labour conditions may significantly impact a hacker's motivation to work on a programme. For example, it has been found that hackers from developing countries with weaker economies make up a significant proportion of those contributing to bug bounty programmes [13, 11]. This may be, in part, due to the increased salary that can be made from bounty payouts when compared to the local median annual wage [99]. In these regions, monetary rewards may become the primary motivator. In contrast, the uncertainty of rewards and relatively high median wage within a richer economy may cause hackers to favour conventional employment.

The use of a tiered payout structure has been found to encourage participation in a particular programme [75]. Organisations will offer a range of payouts for each severity category of

reported vulnerability, e.g., $1,000–$5,000 for critical bugs. The distribution of payout structures for bug bounty programmes on HackerOne is investigated in Section 3.7, where it is noted that the average payout increases significantly with severity (see Table 3.4). Fryer and Simperl [86] found that Bugcrowd has initial recommendations for the payout structure, dependent on the maturity of an organisation's security process maturity, however the rationale behind the suggestions is not transparent. As discussed in Chapter 2, operators noted that platforms often give recommendations to the operators of new programmes to assist in setting suitable payout structures.

One cause of rising operational costs is due to the perceived increase of product security over time [75]. Organisations may judge their own security based on the current payout level [181]. It is suggested that increasing payouts over time motivates hackers to search beyond the initial easy bugs [252, 157].

4.1.2 Behaviour

Organisations have a pre-adoption fear of untrustworthy hackers participating in the search for vulnerabilities (see Section 2.5.1). Instead of turning over the discovered bugs to the programme operator, hackers may choose to sell the bugs on the black or grey markets [95]. A study by Egelman et al. [68] compared arguments for the use of zero-day markets (ZDMs) and vulnerability reward programmes (VRPs), raising the question "can prices be regulated, e.g., by VRPs or other mechanisms, in such away as to avoid the need for ZDMs?" [68].

Meakins [165] notes that grey markets can be a big source of vulnerabilities, often attracting the attention of government intelligence agencies. Governments may decide to pass on details of purchased vulnerabilities to the relevant organisation; this is decided through a Vulnerabilities Equities Process (VEP), and is dependent on the strategic value of the vulnerability in question [247]. The economic factors surrounding the decision for a hacker to sell a vulnerability to a black or grey market are among many that influence hacker behaviour.

Kesan and Hayes [131] proposed an independent third party to regulate the sale of vulnerabilities. Another solution puts further emphasis on the use of attractive bug bounty programmes, providing hackers with an easy, legitimate and legal method to monetise their

work [11].

It seems somewhat inevitable that the proclivity of certain hackers to ignore white markets will continue to shape the operation of bug bounty programmes, potentially influencing the behaviour of all vulnerability hunters.

4.1.3 Productivity

Increasing the number of participants, rather than investing resources to maintain current high-productivity hackers, is often recommended as the most effective way to increase the rate of bug discovery [157, 271]. Discussed in Section 2.5, maintaining high-productivity hackers is a challenge faced by current programmes. It is recommended that organisations endeavour to have fast and effective communication between security teams and hackers, thus helping to maintain their good standing with participating hackers [158, 252].

Other suggestions to improve productivity and bolster motivation include: having experts liaise with hackers, 'game-ifying' the searching and report process, and decomposing search domains into small task areas [67, 138, 141]. So-called 'gamification' of platforms currently exists in the form of leaderboards, badges, and reward points; however, it has been recognised that the extent to which such techniques are employed is limited [183].

4.2 Motivating factors

With limited security budgets, it is difficult to allocate resources in such a way to cover all aspects of security. This is complicated as not all organisations conduct detailed cost–benefit analyses for security investments [198], thus making it increasingly difficult to choose appropriate security investments. These difficulties can be attributed, in part, to organisations possessing incomplete information.

In Chapter 3, we discussed the idea that bug bounty programmes can offer organisations quantifiable returns (e.g., $ per vulnerability found). However, when considering a new programme, a resource-allocation problem arises: How should one choose between competing platforms such that the highest rate of vulnerability discovery is achieved for the lowest cost?

To explore this question, we consider a manager that is looking either to launch a new programme (hosted on a platform) or to modify an existing programme. With many platforms available for hosting, a manager must consider how to allocate resources to utilise the products and services of one or more platforms.

Raymond suggested that it is beneficial to have as many individuals searching for vulnerabilities as possible [195]. Arising from this is a clear network externality that impacts the utility an organisation can gain from the use of a particular bug bounty platform: platforms should become more attractive as their hacker-base grows (this is recognised by operators in Section 2.7.2). However, although platforms claim to harbour hundreds-of-thousands of hackers (recall Table 4.1), organisations still worry about the inactivity (and low motivation) exhibited by hackers (see Section 2.5 and work by Al-Banna et al. [9]). As platforms (understandably) do not publish accurate figures representing the number of hackers actively participating at a particular point in time, there exists a significant degree of uncertainty with regards to the expected utility gained when using a specific platform [128]. The operators of CVD programmes in Section 2.7.2 also noted the economic misalignment of incentives between platforms and programmes.

It follows that further investigation into the true number of active hackers on a platform, as well as their behaviour, is needed to address the information asymmetry that exists between programme managers and bug bounty platforms. Resolving this asymmetry has the potential to aid managers in maximising the benefits an organisation can receive through the operation of a bug bounty programme.

This chapter seeks to address the research question *How effective are bug bounty platforms at retaining the interest of hackers?* through the following set of subsidiary questions:

1. How many hackers are actively engaged in the search for vulnerabilities?
2. What are the archetypal behaviours of hackers?
3. Is participation in bug bounty programmes currently viable for most hackers?

Bug bounty platforms claim to leverage large, diverse 'communities' of hackers to address global security concerns. However, it is necessary to distinguish between active participants

and hackers that are no longer involved in the search for vulnerabilities. Thus, Question 1 aims to quantify the number of active and inactive hackers (taking Bugcrowd as a representative example), to provide an insight into the long-term motivation of hackers as well as a more accurate representation of the size of the network.

The ambition of Question 2 is to help identify characteristic behavioural patterns that are common amongst subsets of the user-base. Additionally, we explore the behaviour of the top hackers found on Bugcrowd, who have been shown to contribute far more than other bug hunters (see Section 3.8 and [255]). Subsequently, we aim to categorise hackers into productivity groups and provide qualitative descriptions of the predominant working patterns observed in the data.

Question 3 seeks to combine the insights and results from the previous questions to comment on the viability of participation in the search for vulnerabilities. Commonly found unproductive and inconsistent behaviour may indicate that the current system is unsustainable for many hackers, who in turn choose not to continue to participate.

4.3 Data

4.3.1 Data collection

The bug bounty platform Bugcrowd serves as the source of data for the work presented in this chapter, as it allows for the output of hackers to be studied over time through time-series data presented on each user profile. Furthermore, as one of the major platforms, it attracts significant interest from large numbers of hackers and organisations.

A conceptual diagram of the utilised data collection, modelling and validation processes is shown in Figure 4.1. The remainder of this section will detail the data collection stages.

As of the 17^{th} of January 2021, Bugcrowd was host to 129 public bug bounty programmes that are open to participation from any individual (private programmes require a hacker to be invited before they are allowed to participate). Upon the submission of a valid vulnerability report, a hacker will be listed within the "Hall of Fame" for the corresponding programme. As such, the "Hall of Fame" for each programme contains a list of all hackers that have previously

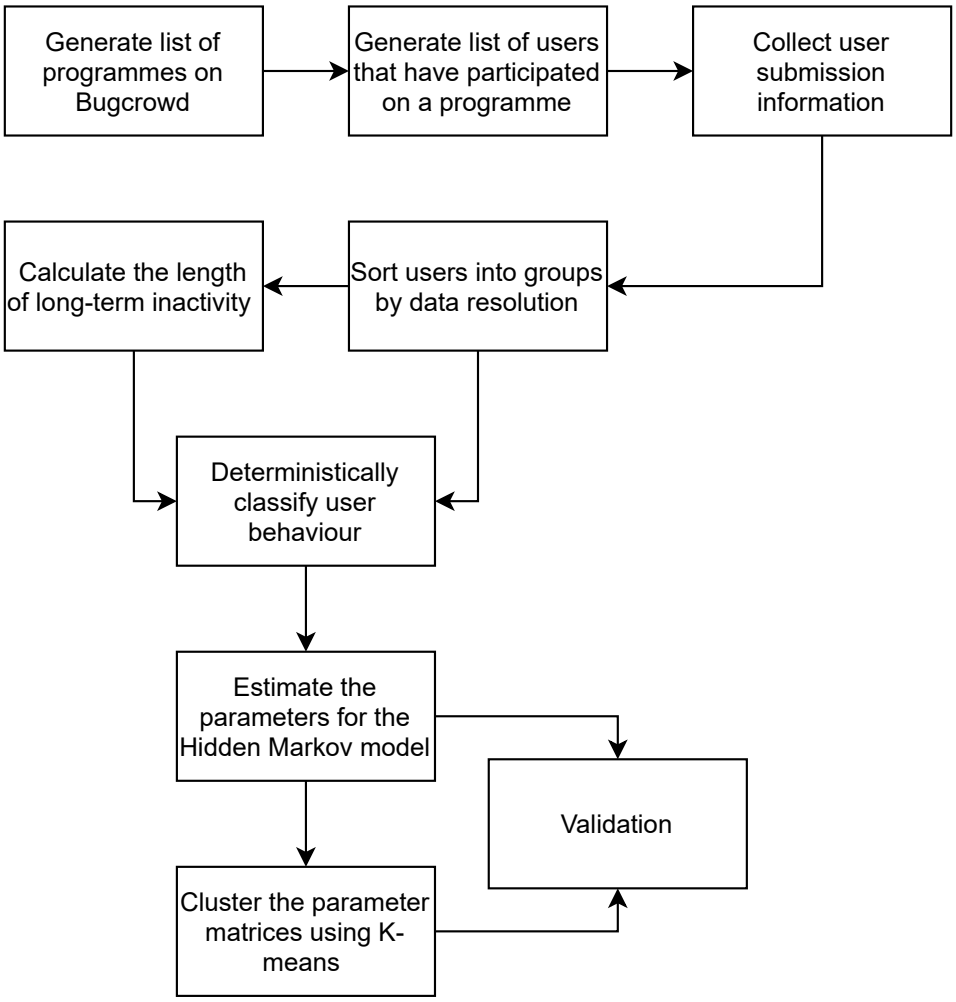

Figure 4.1: Conceptual diagram of the data pipeline used to model hacker behaviour.

participated. (Some hackers may choose to remain anonymous, in which case "Private User" is displayed instead of a username.)

Using all available public programmes on Bugcrowd, a list of 6,927 unique usernames was collected, with each corresponding to a hacker who has submitted at least one valid vulnerability report in the past (a valid report may not be unique, credit is sometimes given to those that submit a duplicate report).

Following the collection of all relevant user-names, time-series submission data was scraped from each user's profile page (114 profiles were unavailable or empty). An example of the time-series is shown in Figure 4.2.

Bugcrowd uses binning to group occurrences of vulnerability reports into fixed intervals of

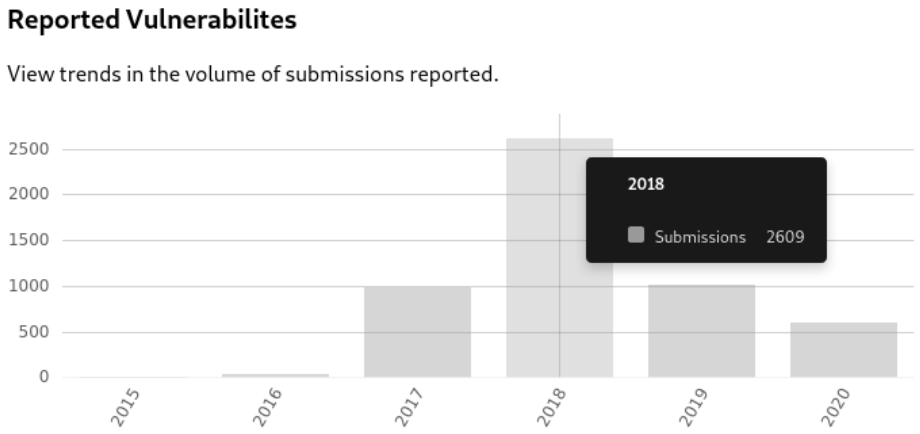

Figure 4.2: Example of a hacker submission time-series collected from Bugcrowd.

			Age (Intervals)	
Timebase	Users in group	Min	Avg.	Max
Day	74	6	35	63
Week	958	11	26	36
Month	2220	9	15	22
Quarter	2135	8	12	17
Year	1426	5	7	9

Table 4.2: Hacker profiles grouped by timebase.

time (e.g., number of submission made within a month). The granularity of the data (temporal resolution of the time-series) is determined by the age of a particular account. As accounts become older, the temporal resolution decreases. For example, an account created 63 days ago will have the submission data displayed by day, but an account created 77 days ago will use weekly resolution. The relationship between the age of the account and resolution of the data is set by Bugcrowd, and unfortunately cannot be changed.

We refer to the "timebase" as being the temporal resolution of a given time-series, and the "intervals" as the number of data points. An account with weekly data displayed on the time-series is described as being in the "week" timebase category. Table 4.2 shows the user data categorised by timebase, and summarises the minimum and maximum age (number of intervals) of user accounts. Data was collected from accounts ranging from 6 days to 9 years old.

4.3.2 Activity classification

To study the behaviour of hackers, it is necessary to classify a user's activity over time. A conceptual model of a hacker's behaviour is created from first principles, and contains three primary activities:

- *Submitting*: A hacker has submitted at least one report in a given interval of time.
- *Searching*: A theorised period of time in which a hacker is actively searching for vulnerabilities. A survey of hackers conducted by Hata et al. found that the average hacker takes up to a week in the search for a vulnerability [110]. As such, we define a fixed period of up to 7 days before the submission of a vulnerability report in which we believe the hacker is actively searching for vulnerabilities.
- *Inactive*: There are extended periods of time in which some hackers seem neither to be searching nor submitting. We further decompose inactive behaviour into three sub-types:
    1. Initial account inactivity: The period of time after account creation and before searching for, or submitting, the first report.
    2. Inter-report inactivity: The measurable period of time between submissions that is not classified as searching.
    3. Long-term inactivity: A state in which a hacker is no longer participating in the search for vulnerabilities.

For all user time-series, we deterministically label each interval of time with one of these activities.

4.3.3 Classifying long-term inactivity

As stated in Section 4.3.2, one of the inactive sub-types relates to the state in which a hacker is no longer participating on the platform. Furthermore, it is assumed that they will not return. To determine the number active hackers on a platform, it is necessary to know the probability of a user submitting a report in the future given that they are currently inactive.

From the time-series of all hackers that have submitted reports on at least two separate occasions, the lengths of inter-report inactivity between adjacent submissions are recorded. From this, distributions of the inter-report inactive durations are created for users within each timebase group (e.g. year, quarter, etc.). The distributions from higher resolution timebases are used as prior distributions to re-sample data in one timebase to a shorter one (e.g. year to quarter). Where a prior distribution can't be generated from existing data, a uniform prior is used. This allows for all inter-report intervals to be re-sampled into the day timebase.

The resulting distribution represents the probability of a report being submitted on or before day n, given $n-1$ days of inactivity. Through Monte Carlo simulation we find that, after submitting a report, there is a 95% probability of the next report being submitted within 289 days [285,293] (95% CI). This allows us to define a length of continuous inactivity greater than 289 days to be classified as long-term inactive, with a Type 1 error of 5%. If a hacker remains inactive for over 289 days following their last submission, the period is reclassified as 'long-term inactive', with the hacker labelled no longer active. In order to make an estimate of the number of hackers that have left the platform, it is necessary to assume that a period of inactivity greater than 289 days is due to a hacker being permanently inactive.

4.4 Modelling behaviour

4.4.1 Model assumptions

There are several important assumptions made about the behaviour of hackers. First, hackers that enter a state of long-term inactivity will not participate again in the future. As stated in Section 4.3.3, we take a time limit that is accurate for 95% of hackers. This implies that the long-term inactive state in the model is absorbing (once entered it cannot be left). Second, we do not attempt to model the number of reports submitted in one interval. Instead, we are concerned only if a hacker has submitted any report in a given interval. Third, we assume that all parameters governing the behaviour of hackers are temporally homogeneous. The limited resolution of the data results in time-series with few points. It is, therefore, not appropriate to explore temporal heterogeneity within the parameters.

	Initial state			Final state		
Timebase	0	1	3	2	3	4
Day	7	17	50	69	5	0
Week	281	136	541	898	60	0
Month	709	312	1199	1308	272	640
Quarter	1054	0	1081	638	440	1057
Year	554	0	872	0	513	913

Table 4.3: Distribution of initial and final states.

The model adheres to the two Markov assumptions [194]: the probability of moving to a particular state is dependent only on the current state (first-order Markov chain) and the transition probabilities exhibit temporal homogeneity (stationary process).

4.4.2 Hidden Markov Model

Previous work has highlighted the success of Hidden Markov Models (HMMs) when applied to modelling sequential data [109, 135], especially when used for state-based behavioural models [116].

The activities outlined in Section 4.3.2 are used as the hidden states in the behavioural model, as shown in the model topology in Figure 4.3. Note that 'Initial account inactivity' is always a transient state (once left, it is not returned to), and 'Long-term inactivity' is always an absorbing state (once entered, it cannot be left).

If a given hacker has a non-zero transition probability from the 'Submitting' state to the 'Long-term inactive' state, the model becomes an absorbing Markov chain (an absorbing state can always be reached), as all hackers reach the 'Submitting' state at least once. A distribution of the initial and final states is shown in Table 4.3.

The emissions from the model are limited to binary values, indicating if the hacker has submitted reports in a given time interval. Other than the 'Submitting' state, all hidden states emit a zero, representing the absence of new submitted reports.

For model creation and testing, 70% of the hacker time-series dataset is randomly selected, with the remaining 30% used as an unseen validation dataset.

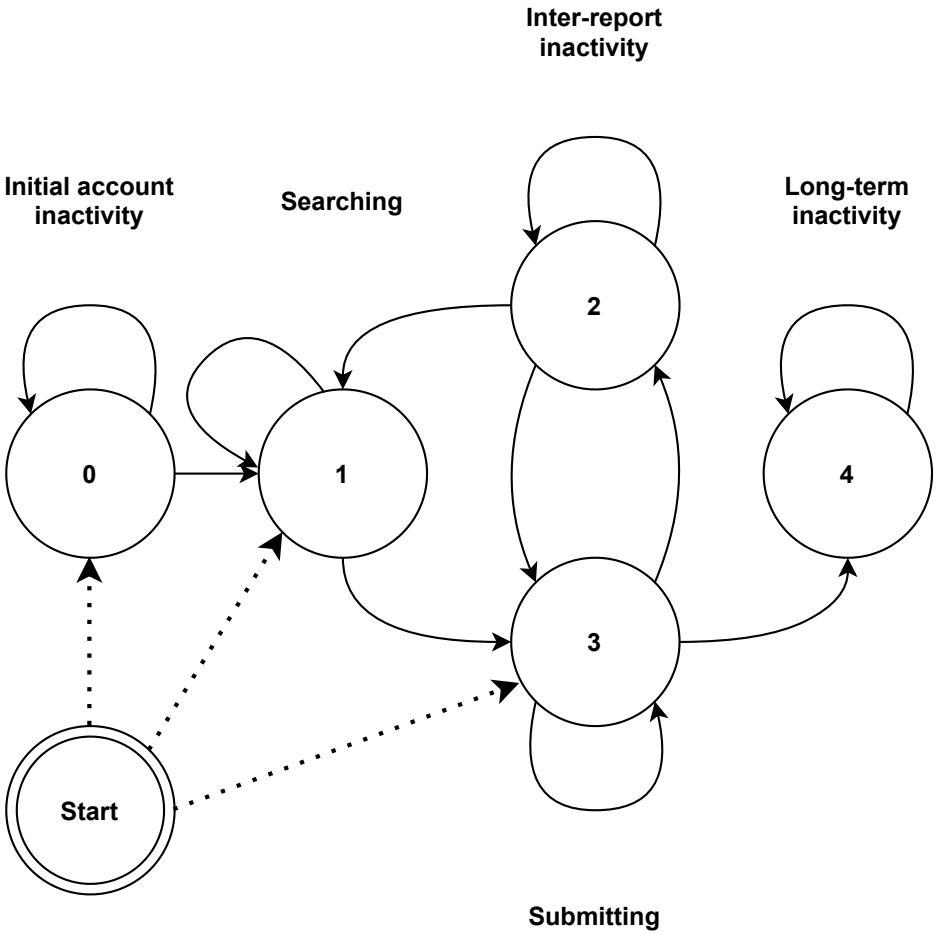

Figure 4.3: Topology of the Hidden Markov Model used to model hacker behaviour, showing commonly used transition pathways.

4.4.3 Generating the transition matrix

Within each group, the 5×5 state transition matrix is computed for each user and the initial state is recorded. Initially the model is assumed to be fully connected. As both the hidden state sequence and the observation sequence are known (due to the data being fully-labelled), the first-order transition probabilities can be calculated using [212]:

$$p_{ij} = \frac{n_{ij}}{\sum_{j=1}^{k} n_{ij}} \tag{4.1}$$

Here, n_{ij} represents the number of transitions from state i to j and k represents the total number of states.

Timebase	K
Day	23
Week	50
Month	100
Quarter	100
Year	52

Table 4.4: Number of clusters chosen for each group.

4.4.4 Clustering hacker behaviour

To group hackers with similar behavioural patterns, unsupervised clustering was performed within each group. First, simple dimensionality reduction was performed by removing any unused transition probabilities ($p_{ij} = 0$) within a group of hackers (e.g., p_{44} is unused in the day timebase). The remaining states were transformed into a feature vector for each hacker.

A K-means algorithm was used to form clusters using the feature vectors. The Elbow and Silhouette methods were both used to select the appropriate number of clusters within each group [161, 267]; the chosen number of clusters for each group are shown in Table 4.4.

4.4.5 Model validation and replication

We now consider results from the model validation process.

Individual

The model's ability to replicate individual user data was tested. For each hacker in the aforementioned timebase groups, the rate of submission and standard deviation were calculated using the collected time-series data, as shown in Table 4.6.

A time-series was generated using the transition matrix calculated from the labelled data for each hacker. Monte Carlo simulation was used to generate 10,000 synthetic time-series for each hacker. The observed values for the rate of submission and standard deviation were compared to the expected values generated from the simulation, and the Mean Absolute Errors (MAE) were recorded. The time spent in each state was recorded, and MAE across all users was calculated (Table 4.5).

As per Table 4.6, using individual model parameters to generate synthetic time-series leads

	Time-in-state MAE (intervals)				
Timebase	0	1	2	3	4
Day	0.73	1.95	2.45	0.64	NA
Week	0.91	0.47	2.0	1.18	NA
Month	0.73	0.36	1.34	1.08	0.30
Quarter	0.85	NA	0.87	1.04	0.75
Year	0.26	NA	0.25	0.71	0.60

Table 4.5: Mean absolute error calculated for the time in each state, using transition matrices for individual users.

Timebase	Individual Error (%)	Cluster Error (%)
Day	-10.3	-11.3
Week	-12.0	-7.5
Month	-17.8	-21.9
Quarter	-7.2	-7.3
Year	-13.7	-14.1

Table 4.6: Expected rate of submission error, using transition matrices for individual users and those generated from K-means cluster centres.

to an underestimate of the number of emissions when compared to the hacker data. Hackers in the year group exhibit the largest error, possibly due to the limited quantity of data available.

Table 4.5 shows the mean absolute error for the time spent in each state. With the exception of state 2 (Inactive) in the day group, all errors are less than 1 time interval.

Clusters

The cluster centroids generated by the clustering algorithm were used as transition matrices, and the ability to generalise user behaviour was similarly tested. The final number of clusters is shown in Table 4.4. The submission rate error is shown in Table 4.6; the time-in-state error is shown in Table 4.7.

Transforming the centroids for use as transition matrices leads to an underestimate in the number of emissions, with the month group having the largest error of -21.9%. All states see a small increase in the time-in-state MAE when compared with the individual time-in-state error.

	Time-in-state MAE (intervals)				
Timebase	0	1	2	3	4
Day	0.72	1.97	2.47	0.65	NA
Week	0.98	0.67	2.45	1.44	NA
Month	0.78	0.41	1.47	1.16	0.31
Quarter	0.90	NA	0.94	1.12	0.77
Year	0.27	NA	0.25	0.73	0.61

Table 4.7: Mean absolute error calculated for the time in each state, using transition matrices generated from K-means cluster centres.

Analysis

The results demonstrate that the approach to modelling hacker behaviour outlined in Sections 4.3 and 4.4 is effective.

Clustering the individual transition matrices and using the centroids to represent the behaviour of similar groups of hackers leads to a small decrease in performance. This is, in part, due to the cluster centres being an imperfect representation of each parameter matrix within the cluster as there are variances in intra-group hacker behaviour. This is shown by an increase in the (absolute) expected rate of submission error in most timebases, and an increase in the MAE for the time spent in each state (for all states and timebases). Performance may be improved by further optimising the number of clusters used in each timebase.

Aside from the ability of the centroids to capture behavioural characteristics of hackers, they also serve as a compact representation of the behaviour of hackers on Bugcrowd.

4.5 How many hackers are active?

4.5.1 Population of active hackers

We define an active hacker as someone whose behaviour is not classified as long-term inactive. (As stated in Section 4.3.3, any individual with a trailing period of inactivity longer than 289 days is defined as long-term inactive.)

In the model, long-term inactive hackers are represented as finishing in state 4 (long-term inactivity). Table 4.3 shows the final states of all hackers within each timebase. It was found that 2,610 hackers (37.7%) are estimated to no longer be active; for hackers with accounts

older than 9 months old the inactivity rate is 44.3%. A larger proportion of hackers in the Day (93.2%) and Week (93.7%) timebase were found to be in an inactive final state. However, due to the age of the accounts (all less than 289 days), it was not possible to classify any of these accounts as long-term inactive. It is therefore assumed that the true number of inactive hackers is higher than 37.7% due to inactive hackers within these groups with accounts too new to be classified accurately as such.

4.5.2 Analysis

From a user-base of over 200,000 hackers, it is only possible to identify 6,813 unique individuals that have (publicly) participated in at least one public bug bounty programme. A rough estimate of the number of unique anonymous hackers (listed as "Private user" on a Hall of Fame page) can be made using the average proportion of anonymous hackers (relative to public hackers) across all Hall of Fame lists. This leads to an estimate of 3,717 unique hackers that have participated and chosen to remain anonymous (assuming both groups have the same distribution of the number of unique programmes a hacker has participated in). As such, only 10,530 ($\approx$ 5.27%) of accounts created on Bugcrowd are predicted to have participated in at least one bug bounty programme.

Of the 6,813 hackers identified within this study, it is estimated that 2,610 (37.7%) are no longer active participants. As noted in Chapter 2 and by Al-Banna et al., one issue faced by programme operators is the difficulty in maintaining participants (within a specific programme) [9] — not least because common or obvious vulnerabilities are quickly found [148]. The high levels of hacker inactivity suggest that bug bounty platforms are also unable to sustain hacker motivation, even as the number of new programmes (and potential vulnerabilities) continues to grow (see Chapter 3).

Both programme and platform operators should endeavour to explore the impact of financial incentives (by, for example, adjusting bounty values) and ways to improve accessibility, to encourage hackers to continue to search for vulnerabilities. As previously alluded to, gamification of tasks is one such method that has been suggested as a means to increase and sustain motivation in software engineering and crowdsourced activities [138].

The operators of bug bounty programmes may see increasing benefits as the number of active hackers on a platform increases [195]. Calculating the magnitude of this positive network externality requires knowledge of the number of hackers searching for vulnerabilities — a figure not currently published by the operators of bug bounty platforms. It is hoped that the results of the study will help address this existing network externality, and that the method may be extended to platforms other than Bugcrowd.

4.6 What are the archetypal behaviours of hackers?

4.6.1 Archetypal behaviours of top hackers

As highlighted in Chapter 3, a small proportion of the overall user-base contributes a significant proportion of the submitted vulnerabilities; the top 100 hackers on HackerOne are responsible for over 50% of all vulnerability discoveries. As such, the work habits of the top 100 Bugcrowd hackers were studied. To be in the top 100, a hacker must discover and report more than 193 vulnerabilities. Hackers meeting this criterion are found in the month (3), quarter (28), and year (69) groups.

Using the inactivity limit of 289 days, the number of long-term inactive top 100 hackers is estimated to be 19 (1 in the quarter group, and 18 in the year group). Within each group, the proportion of hackers classified as long-term inactive is far less than the overall proportion of long-term inactive hackers. As detailed in Section 4.5, the estimated number of long-term inactive hackers within the groups is: 28.8% in months (0.0% in top 100), 49.5% in quarters (3.6% in top 100), and 64.0% for year (26.1% in top 100).

As defined in Section 4.3.2, initial account inactivity is the period of time before the first interval containing submissions. Table 4.3 shows the number of hackers that start in an inactive state. For the top 100 hackers, 14 in the month group and 15 in the year group start as inactive. The proportion of top 100 hackers that start as inactive is comparable or lower than the group average.

Analysing the distribution of parameters within the individual transition matrices allows for further investigation into the work habits of the top hackers. The transition parameter

	top 100			All		
Timebase	$\bar{p}_{00}$	$\bar{p}_{22}$	$\bar{p}_{33}$	$\bar{p}_{00}$	$\bar{p}_{22}$	$\bar{p}_{33}$
Day	NA	NA	NA	0.23	0.82	0.22
Week	NA	NA	NA	0.34	0.75	0.40
Month	0.00	0.00	0.98	0.29	0.48	0.39
Quarter	0.17	0.05	0.94	0.32	0.36	0.39
Year	0.05	0.02	0.87	0.19	0.15	0.39

Table 4.8: Mean parameter values of p_{00}, p_{22}, and p_{33} for all hackers and top 100 hackers.

p_{33} represents the probability of submitting vulnerability reports in the next available interval, given that reports were submitted in the current interval. High values of the parameter ($p_{33} \in [0, 1]$) indicate that a hacker is consistently submitting reports in each observable interval of time. Lower values represent more sporadic behaviour. As shown in Table 4.8, the top 100 have significantly higher average p_{33} values than other hackers within the same group.

To quantify consistent behaviour during the period of time in which a hacker is active, the following harmonic mean is defined:

$$\alpha = \frac{2}{\frac{1}{1-p_{22}} + \frac{1}{p_{33}}} \tag{4.2}$$

Here, α is the harmonic mean ($\alpha \in (0, 1]$).

Top hackers across all groups have an $\bar{\alpha} = 0.923$, far greater than the figure for all hackers ($\bar{\alpha} = 0.459$). This indicates that the top 100 hackers tend to discover and submit vulnerabilities far more consistently than other hackers on Bugcrowd. This also suggests that barriers, other than skill, may limit the majority of hackers in the search for vulnerabilities in a consistent manner, and that only a small proportion of hackers are able to search on a full-time basis.

4.6.2 Archetypal behaviours of all hackers

Clustering hackers allows for the identification of similar behavioural patterns (this process is detailed in Section 4.4.4).

For each cluster a value of α was calculated using Equation 4.2, allowing the clusters to be divided equally between four productivity groups: very low, low, medium, and high. Qualitative descriptions of hacker working patterns for each group are given below.

- Very low: Over the lifetime of an account, hackers tend to submit reports in one or two intervals, before going into an inactive state. It is common to see activity only in the first few intervals after account creation.

- Low: Hackers will submit sporadically over time, often with long gaps of inactivity between submissions.

- Medium: Groups of sequential submissions are observable, indicating longer periods of hacker activity.

- High: After a possible period of inactivity following account creation, hackers will submit reports in almost all intervals. All of the active top 100 hackers identified in Section 4.6 are part of this productivity group.

4.6.3 Analysis

From Bugcrowd, the top 100 hackers (by number of submissions) were identified as those with over 193 discovered vulnerabilities. Within this group far fewer hackers (19%) are classified as long-term inactive (as compared to the all hackers), suggesting that there is continual motivation to search for vulnerabilities.

It has previously been found that use of leaderboards, achievement points, and badges of recognition motivate some students to work harder and undertake increasingly challenging tasks [84]. The ambition to retain a top leaderboard ranking may serve as an additional source of motivation for these top hackers. Highly consistent submission patterns are a characteristic feature within the group of top hackers ($\bar{\alpha} = 0.923$), and all top 100 hackers are found within the high-productivity group (as defined in Section 4.6.2).

A typical (and commonly found) working pattern of hackers in the very low productivity group is submissions being made in one or two intervals following account creation, followed by inactivity. As noted in Section 3.2, a similar behaviour is reported by Zhao et al. [271]. Programme operators have expressed concerns over the low quality of submitted reports and the large amount of time and resources required to process them (see Section 2.5 and work by Al-Banna et al. [9]). The presence of large numbers of these 'one-off' hackers, in conjunction

with their unfamiliarity with the discovery and reporting process, may result in dissatisfaction from both parties. Further investigation is needed to understand why this type of behaviour is prevalent among hackers.

4.7 Is participation viable for most hackers?

4.7.1 Viability of long term participation

Of the 6,813 hackers identified within the study, it is predicted that 2,610 are no longer active. The proportion of hackers entering a state of long-term inactivity (Table 4.3) increases with account age: 29% of accounts with an age of 9–22 months are inactive, rising to 50% of account between 8–17 quarters, and 64% for older accounts. The behavioural analysis of Section 4.6.2 reveals that large numbers of hackers can be characterised by their unproductive behaviour. Particularly notable is the subset of hackers that only submit one or two reports after account creation and thereafter end their participation in the search for vulnerabilities.

4.7.2 Analysis

Over time, a greater proportion of hackers become inactive and no longer search for vulnerabilities. This suggests that, in their current configuration, long-term participation in bug bounty programmes is not viable for most hackers. Furthermore, a low proportion ($\approx 5.27\%$) of accounts on Bugcrowd have participated, suggesting that more needs to be done to encourage new users to find their first vulnerability. Encouragement for new users could come in the form of increased gamification of the discovery process, or continued investment in the development and publication of teaching resources. Free online resources such as Bugcrowd University [46] and HackerOne's Hacker101 [102] (acquired from Breaker101 in 2018 [57]) provide individuals with video guides and interactive challenges (e.g. capture the flag) with the aim to improve their understanding of the vulnerability discovery process.

4.8 Summary of findings

Motivated by the question *How effective are bug bounty platforms at retaining the interest of hackers?*, we have analysed the behaviour of 6,813 hackers that have participated in the search for vulnerabilities in bug bounty programmes on Bugcrowd. Additional behavioural analysis has enabled a prediction of the proportion of active users to be made: it is estimated that approximately 2,610 hackers are no longer active in the search for vulnerabilities and that 19% of the top 100 hackers on Bugcrowd are inactive, potentially raising concerns regarding the long-term motivation of hackers when participating in bug bounty programmes.

Providing an otherwise inaccessible figure of the number of active hackers to managers currently operating, or looking to operate, a bug bounty programme has the potential to address the information asymmetry that exists between organisations and platforms. With limited resources to spend on operating a bug bounty programme, it may be advantageous for operators to select the platform with the most active user-base.

The increasing number of inactive hackers over time, together with the small proportion of accounts that have submitted at least one vulnerability report, suggests that participation is not viable for a significant number of individuals. Consequently, the operators of programmes and platforms may wish to further consider how best to motivate both new and experienced hackers to continue searching for vulnerabilities.

It is important that we note some limitations. First, only Bugcrowd allows for the contributions of hackers over time to be collected and analysed. Acquiring user data from only one platform (albeit a major platform) may lead to poor generalisation of results when the methodology is applied to other platforms. Second, data could only be collected from individuals with public accounts and that have previously participated in a public programme. For a given Hall of Fame page, on average approximately 35% of accounts listed are anonymous, preventing the collection of user data. The inability to collect information about all hackers on Bugcrowd may lead to results being unrepresentative of hacker behaviour, particularly within private programmes. Third, the granularity (temporal resolution) of the data collected limits limits the accuracy of the calculated model parameters. Access to higher resolution data (e.g.

daily data for all hackers) would allow for the temporal heterogeneity of the parameters to be investigated, giving rise to greater in-depth behavioural analysis.

In the next chapter we investigate the policies that help mediate the interactions between hackers, programmes, and platforms. This is achieved through the analysis of a wide range of policies used by the operators of stand-alone and platform-based CVD programmes.

Chapter 5

Understanding Programme Policies

The previous chapters have, in one form or another, touched upon the role that policy and regulation play in shaping the interactions between stakeholders in the field of crowdsourced vulnerability discovery. Hackers, organisations, and platforms are all subject to a litany of contractual obligations, legislation applicable to one or more jurisdictions, and social conventions that may have arisen to help bring stability to markets where trustworthiness is not inherently present.

In Chapter 2, operators voiced their opinions on how best to operate a CVD programme, and how to make use of programme policies to help guide the actions of hackers in an effort to mitigate the systemic issues that are faced by many. As will be discussed in this chapter, these policies can be used to set clear boundaries between acceptable and undesirable behaviour (of which the ramifications are explored). The global distribution of hackers, as discussed in Chapter 3, brings with it complexities that must be resolved by organisations and platforms, who themselves can be subject to strict restrictions on the hackers that they can interact with during the search for vulnerabilities. The uncertainty brought forth by the multitude of restrictions may help to explain, in part, the behaviour of hackers observed in the results presented in Chapter 4. As some have noted, the potential repercussions that may come with vulnerability research cause an uneasiness in hackers that may not be faced in typical employment [227, 272].

Although many platforms offer guidance on how to communicate the information necessary

for effective and compliant participation in an organisation's programme[1], there is a great deal of variation in the content of programme policy documents. For hackers (particularly those who are new participants), the differing language used across programmes, including those on the same platform, adds further complications that they must address. This can be particularly problematic if an individual does not fully understand the nuances of certain eligibility conditions, participation requirements, or legal clauses used across a range of programmes. As such, it is useful to further understand how key components and concepts are conveyed to hackers in the policy documents associated with a given CVD programme — the outcomes of which can be beneficial to both hackers and organisations looking to understand content and variability across CVD programmes. To this end, this chapter seeks to answer the research question *What information do organisations convey to hackers through public policy documents?*

In Section 5.1, existing literature concerning the role of policy documents is explored. Of particular importance are the current links between CVD policies and the economic theory of institutions. Viewing policies through this lens opens up CVD programmes to further analysis using a well established framework. Section 5.2 presents the rationale behind a continued investigation into the use of policy documents in CVD programmes, and outlines the research questions of interest in this chapter. As in prior chapters, data collected from bug bounty platforms is of interest. However, as explained in Section 5.3, the policies from a wider range of platforms and stand-alone programmes are considered. Furthermore, an annotation process is described, transforming the data for use in the supervised models introduced in Section 5.4. Using the concepts from institutional theory, the formal constraints are explored in Section 5.5, informal constraints and variations are presented in Section 5.6, and a consideration of the institutional components present in policy documents is found in Section 5.7. To end the chapter, the three key findings, alongside some limitations of the proposed approaches to

[1]For example, in Bugcrowd's guide to participation, it is recommended that hackers read the policy documents carefully in order to be aware of details surrounding programme scope, access or authentication requirements, description of the assets, and the rewards offered by the organisation (see `https://www.bugcrowd.com/resources/report/your-guide-to-hacking-with-bugcrowd-and-why-you-should-get-started-today/`). Similarly, HackerOne offers guidance to hackers, including recommendations on the key components of high-quality reports such as providing detailed descriptions that enable reproducible results, including a proof-of-concept if permitted, and conducting a comprehensive evaluation of the potential impact of the identified vulnerability (see `https://docs.hackerone.com/hackers/quality-reports.html`).

policy analysis, are summarised in Section 5.8. The work presented in this chapter is based on [257].

5.1 Role of policies in CVD

Although there has been significant research interest in recent years surrounding the use of CVD programmes, there has been rather less research covering the policies employed by organisations.

Earlier work by Kuehn and Mueller [140] explores software vulnerability markets, paying particular attention to BBPs through the lens of North's theory of institutional economics [177]. In *Institutions, Institutional Change and Economic Performance* [177], North outlines the key concepts underpinning institutional economics and the effect institutions have upon the economic performance of a given system (e.g. with a country or market). Institutions (the "rules of the game" [177]) are comprised of both formal (rules that can be explicitly enforced, such as laws) and informal (conventions, social norms, and codes of conduct) constraints that are imposed upon participants. The interplay between the formal and informal constraints help structure interactions between individuals. While the analysis by Kuehn and Mueller [140] does not extend the theory of institutions to cover bug bounty platforms, it may be prudent to briefly consider the present-day structure through this lens.

In a scenario involving a hacker, an organisation (operating a CVD programme), and a platform acting as the third-party intermediary, all are bound to transact and operate in accordance with the institutions. Relevant legal entities impose a set of formal constraints upon all parties. For example, US-based hackers will be subject to the Computer Fraud and Abuse Act (CFAA) [239] and the Digital Millennium Copyright Act (DMCA) [241] — both of which are enforceable by the Federal government — when conducting security research (activities conducted in certain US Federal assets may also be subject to the National Information Infrastructure Protection Act of 1996 [240]). As another example, US-based organisations and the platform may be subject to the Anti-Money Laundering Act (AMLA) [242], export restrictions, and sanction requirements when transacting with hackers — particularly if bounties are paid

(for a further discussion of the complexities around the global flow of information and money in the context of CVD programmes, see the contribution of Zhao et al. [272]).

Both the organisation and the platform will impose additional informal constraints upon the hackers. Typically, platforms will require hackers to abide by a code of conduct in order to discourage undesirable behaviour (e.g. spam reports, reputation farming, and abusive communication). To enforce such rules, a platform typically has the power to ban or suspend user accounts. Organisations will impose constraints upon the hacker through programme policies that outline rules, guidelines, or conditions for eligible participation. However, unlike the platform in this scenario, the organisation can do little to directly enforce the informal constraints, aside from withholding any potential payments.

As highlighted by North [177], when the cost of transaction is prohibitively high (as it is in vulnerability markets [140, 170]) and enforcement of the formal constraints can be costly, the parties must rely on informal constraints in order for the market to function. Therefore, understanding these informal constraints is of particular importance. Kuehn and Mueller [140] identify four institutional elements that help facilitate a market: procedures, technical specifications, terms and conditions, and acknowledgments and reputation.

Work undertaken by Laszka et al. [145] is of particular relevance to the premise of this study. In [145], the authors analyse the general statistics and policy documents of 111 CVD programmes (as listed on HackerOne in 2016). In order to further investigate the contents of the policy documents, the authors first qualitatively define a general taxonomy that attempts to understand the distinct components that organisations use to communicate programme rules. Consequently, a taxonomy of 12 components is constructed and used to classify sections of the 111 collected policy documents. The presence of the components is then investigated with respect to the reporting statistics of each programme. The authors argue that programmes "are on average associated with better success characteristics if the level of comprehensiveness of their rules of engagement increases" [145]. However, in the absence of causal analysis, it is not possible to determine the impact of a component on the success (or otherwise) of a programme.

5.2 Motivation factors

The focus of the study described in this chapter is, in part, inspired by the qualitative component of the work undertaken by Laszka et al. [145], with a view to expanding the analysis of policy documents beyond the narrow focus of prior work. Furthermore, in consideration of the rules that organisations now include in a policy document, the lens of institutional theory is focused upon CVD programmes [177, 140].

As noted in Chapter 3, bug bounty programmes sometimes offer eye-catching rewards to hackers for the discovery and disclosure of certain highly-prized vulnerabilities, with some programmes advertising potential single-report payouts in the region of millions (and sometimes tens of millions) of dollars (e.g. the $10,000,000 Wormhole programme[2]). While hackers may be motivated to search in a particular organisation's assets, aspiring to earn lucrative rewards, they may not consider the implicit or explicit risks associated with participating in a programme, such as the varying levels of supposed legal protection conferred through an organisation's Safe Harbor clauses [64, 65]. A thorough definition of Safe Harbor is given in Section 2.4.4. It follows that, depending on the programme policy, a hacker will be exposed to varying levels of risk. Note that a Safe Harbor clause can do little, if anything, to immediately protect a hacker from third-party legal action. Furthermore, as a condition for reward eligibility, some organisations may require hackers to assign or license their intellectual property, such as a report, to the organisations. Whether or not these clauses are clearly outlined before or after the submission of a report may impact a hacker's decision to participate.

While grandiose maximum payouts may be advertised, these are but an indication of the maximum amounts that could be rewarded. Organisations are not bound by these figures, nor are they bound to paying out any bounty upon the receipt of a valid vulnerability report. As highlighted by Miller's experience negotiating the sale of software vulnerabilities through a broker [170], and underpinned by Arrow's work concerning information as a commodity [24], once a hacker reveals too much information about a vulnerability (e.g. when proving its existence), the information asymmetry is resolved, diminishing the value of the information offered by the

[2] `https://immunefi.com/bounty/wormhole/`

hacker. Perhaps this highlights the importance of the current informal constraints that lead an organisation to reward a hacker in the case of a bug bounty; however, as noted in Section 3.1, there are formal constraints that may prevent an organisation from doing so. Additional complications may be raised by the proclivity of Web3-based organisations to award payouts in cryptocurrencies, rather than via fiat currencies. These rewards appear to be, at times, slightly disingenuous as the bounties will be advertised in USD, yet the policy will specify that the payouts may be issued in an equivalent amount of cryptocurrency (to the discretion of the organisation). Therefore, in order to avert later complications, it is necessary for hackers to understand the institutions concerning the rewarding process.

There may also be benefits to organisations that want to better understand the current policies published by other organisations that operate CVD programmes. As mentioned in Section 2.7.2, it is not uncommon for organisations to use the policies from other organisations, or publicly available templates, as a reference point when attempting to create their own. It is noted by some operators that this could lead to the copying of 'stale' policies that appear outdated in the wider context. As explored by North [177], albeit in case studies unrelated to this topic, participants should adapt to the institutional change that occurs. In the context of CVD programmes, it may be beneficial for organisations to alter their policies over time as norms change. As such, the identification of current normative policies and rules allows laggards an opportunity to modify their programmes and 'catch up'.

It follows that renewed investigation into the CVD policies put forward by organisations, across a breadth of platforms and sources, would have the potential to provide both hackers and organisations a greater understanding of the formal and informal constraints contained within. To help answer the research question put forward in this chapter *What information do organisations convey to hackers through public policy documents?* we consider the following subsidiary questions:

1. What are the formal constraints communicated to hackers within the policy, particularly within legal statements? How do these impact hackers?

2. What is the variation in the informal constraints that are communicated to hackers?

3. To what extent do organisations include institutional elements in their CVD programme policies?

Question 1 aims to identify the formal constraints imposed upon hackers, particularly those that are present in the relevant legal components as defined by Laszka et al. [145]. It also considers the various impacts these might have on the actions of hackers. In a similar vein, the ambition of Question 2 is to identify the variation in the informal constraints imposed upon hackers, helping to link the taxonomy of Laszka et al. [145] to institutional economic theory [177, 140]. Question 3 seeks to comment on the policies as a whole, using insights garnered from answering the previous questions, in order to better understand the current state of CVD policies across multiple platforms and stand-alone programmes.

5.3 Data

The approach to data collection and annotation in this section. Considerations for the data underpinning the study are presented in Section 5.3.1. This includes details concerning the sourcing of the data from bug bounty platforms and stand-alone programmes, the coverage of the data, and the extent to which data was collected. Section 5.3.2 introduces the annotation schemes used to categorise data in preparation for use in the creation of supervised NLP models. Explosion AI's Prodigy annotation tool was used to create annotated datasets for both Named Entity Recognition (NER) and text-classification downstream tasks.

5.3.1 Data Collection

Unlike the work presented in previous chapters that has been limited to only a handful of sources, particularly from prominent bug bounty platforms (such as HackerOne and Bugcrowd), data for this study was collected from 13 bug bounty platforms and a wide selection of independent (stand-alone) bug bounty and responsible disclosure programmes. This included a selection of former programmes that have been shut down or are otherwise inactive. Platforms were selected using online repositories, such as the 'Open-Sourced Collection of Bug Bounty

Platforms'[3], with the added criteria that some (at least one) programme information be public, content was available in English, and the platform website was available at the time of data collection.

In total, the public policy documents of 1,243 programmes were collected from the aforementioned bug bounty platforms. As commented by Christian [50], bug bounty platforms are host to a significant number of organisations that choose to operate private, invite-only, programmes, that cannot be accessed or viewed without prior arrangement. As such, the policies for these private programmes could not be collected. An additional 3,390 organisations thought to operate a stand-alone CVD programme were investigated, and any relevant data downloaded. After filtering out programmes that didn't have any viewable policy information, 555 stand-alone CVD programmes remained.

A summary of the data sources is provided in Table 5.1. The US-based platform HackerOne is the single largest unified source of data, and, along with Bugcrowd (which is also US-based), are both frequently used as a source of data for research pertaining to the use of bug bounty programmes [271, 273, 145]. In Chapter 4, Bugcrowd was the focus of the effort to model the platform-wide behaviour of hackers, the data outlined in this section is distinct from that collected from user profiles. Unlike the other sources, the Immunefi[4] and HackenProof[5] platforms focus almost exclusively on Web3-based programme assets (e.g. smart contracts). The list of independent programmes was curated from publicly available repositories[6] of organisations known to have — or thought to have — accepted vulnerability disclosures.

For each programme, all relevant data in plaintext or markdown format pertaining to the organisation's CVD programme was collected in early April 2022. Pertinent data presented in tabular formats was also collected using custom Selenium or Beautiful Soup Python scripts. Where possible, raw markdown data was preferred over rendered markdown text owing to the greater flexibility allowed for in later processing steps. The data collected included the following.

[3]See: `https://github.com/disclose/bug-bounty-platforms`

[4]`https://immunefi.com`

[5]`https://hackenproof.com`

[6]Using the external programmes filter on `https://hackerone.com/directory/` and from `https://github.com/disclose/diodb`

Source	Programmes	Country of registration
HackerOne	526	US
Immunefi	280	Singapore
Bugcrowd	224	US
Intigriti	76	Belgium
HackenProof	53	Estonia
YesWeHack	30	France
BugBounty.jp	20	Japan
BugBase	10	India
WhiteHub	8	Vietnam
Security@me	7	Australia
RedStorm	6	Indonesia
Crowdswarm	2	UAE
Huntr	1	UK
Independent	3,390*	Global

Table 5.1: Summary of the sources used to identify and collect public CVD policy documents, as of 16^{th} April, 2022. Note that only 555 stand-alone, independently hosted, programmes of the 3,390 collected contained useful data. Unfortunately, the number of private programmes that could not be collect is not known, as these are typically hidden from public view (the lack of private programme information available has previously been noted as a barrier to research [50]).

- *General programme statistics.* Programmes hosted on bug bounty platforms typically display statistics that reference certain operational characteristics. For example, statistics relating to the response efficiency of the organisation (e.g. average time to respond, time to bounty, etc.) may be useful for hackers to better understand the interactions they may have after the submission of a report. In addition, figures pertaining to the quantity of reports submitted and resolved, alongside the corresponding bounty payout totals, give an indication of the previous activity of hackers in relation to the programme (see Chapter 3). Although the general statistics are not the focus of the chapter herein, the data was collected for the purpose of making the material accessible to researchers via a GitHub repository in order to support wider research[7].

- *Policy documents.* For all programmes identified in Table 5.1, the published publicly-viewable policy documents (the content of which varies widely) were collected. CVD programme policy documents may contain, but are not limited to: company mission state-

[7] `https://github.com/walshe96/cvd-policy-documents`

ments, programme descriptions, scope definitions, potential rewards, guidelines, rules, legal statements, submission criteria, and submission instructions. An organisation may choose to include some, or none, of the aforementioned policy components, and any other information that is believed to relevant to the operation of the CVD programme. Although there may be a notion of standardisation in terms of policy content and format when viewing documents across a single platform, there can be significant variability across platforms or independent programmes.

- *Scope tables.* Formally defined scope tables detail the in-scope and out-of-scope assets, highlighting areas in which a hacker is permitted to search for vulnerabilities, and those from which they are forbidden. For programmes on the HackerOne platform, these were collected separately from the policy documents.

- *Bounty tables.* The value of the bounty paid out for a given vulnerability report may be conditioned on several factors specified by an organisation through the use of a bounty table. Often, organisations will directly link the bounty to the CVSS score assigned to the vulnerability underlying a report (an example of this can be seen in Table 3.4). However, certain classes of vulnerabilities, such as Remote Code Execution (RCE), may hold more importance to the organisation when identified in certain assets. As such, organisations may instead use a detailed series of bounty tables to convey the payouts awarded to specific vulnerabilities. For programmes on the HackerOne platform, these were collected separately from the policy documents.

- *Programme updates.* Certain bug bounty platforms allow programme operators to publish update messages separately from their main policy documents. Typically, these updates contain information that is more time-sensitive than programme information conveyed elsewhere. As such, these updates enable increased visibility for information pertaining to: limited time events and promotions, notice of noteworthy programme changes (e.g. termination of a programme), and opportunities for hackers to provide feedback to the operators through linked surveys. For programmes on the HackerOne platform, these were collected separately from the policy documents.

Type	Count
General	526
Policy	4,633
Policy changes	68,693
Scope table entries	6,576
Scope table changes	14,479
Bounty table entries	379
Bounty table changes	7,030
Updates	601

Table 5.2: Summary of the data collected from the sources of Table 5.1, as of 16^{th} April, 2022.

- *Document version histories.* Both the HackerOne and Intigriti platforms offer the full version histories for the content displayed on programme pages by the way of markdown documents with changes displayed in the unidiff format. This allows for full transparency of the incremental changes made by the operators over a programme's lifetime. Only the HackerOne version histories, in markdown format, were collected.

A summary of the data collected is shown in Table 5.2. Following the collection of the data, all information and associated metadata was stored in various SQL database tables for retrieval in later stages.

5.3.2 Annotated Dataset Curation

To allow for the training of supervised machine learning models that are capable of utilising semantic features in the text, both paragraph-level multi-category labelling and sentence-level entity labelling were employed. This resulted in the creation of over 12,000 unique hand-labelled sentences (representing 9.7% of the 123,440 sentence corpus) and over 3,000 hand-labelled paragraphs (representing 7.6% of the 39,748 paragraph corpus). The publication of these labour-intensive annotated datasets helps enable further research reliant on large annotated corpora.

The annotation process was guided by Bloomberg's guide covering 'Best Practices for Managing Data Annotation Projects' [248], and served as a background to annotation guideline creation. Paragraphs and sentences (paragraphs split using the NLTK sentence tokenizer[8])

[8]Documentation found at: `https://www.nltk.org/api/nltk.tokenize.html`

Category	Count
COMPANY-STATEMENT	662
REWARD-EVALUATION	466
SCOPE-IN	284
VULN-ELIGIBLE	219
VULN-INELIGIBLE	211
PROHIBITED-ACTIONS	182
GUIDELINES-SUBMISSIONS	174
SCOPE-OUT	155
ENGAGEMENT	119
GUIDELINES-DISCLOSURE	77
LEGAL	71
PARTICIPANT-RESTRICTIONS	38

Table 5.3: Summary of the categories of the 2,618 labels assigned to the 3,003 annotated paragraphs. Note that, with the multi-classification approach, a single paragraph may be assigned multiple labels if it contains components pertaining to multiple categories. A paragraph may be assigned no label if it does not conform to the predefined categories. In the context of the taxonomy, 955 of the 3,003 randomly selected paragraphs were not assigned a label as the information was not relevant (e.g., markdown headers, markdown used for formatting without content, and conversational or seasonal messages such as 'Happy New Year, Hackers!'), could not be assigned an appropriate label due to lack of surrounding context (e.g., lone URLs), or the text was non-English.

were randomly selected across the entire corpus for annotation within their respective tasks. The annotation itself was undertaken by a single coder. As such, the quality of the annotation datasets may be limited due to authorship biases.

A summary of the categories and entities in the annotated datasets are shown in Tables 5.3 and 5.4, respectively. Data for annotation was selected with uniform probability across a deduplicated dataset from all data sources and includes all content types. The annotation work was carried out using the Prodigy[9] annotation tool developed by Explosion AI using a research licence.

As described in Section 5.2, Laszka et al. [145] conducted policy analysis on 111 bug bounty programmes hosted on the HackerOne platform. Within the qualitative component of the work, the authors develop a taxonomy of 12 policy components that organisations use to convey information to hackers about the programme. The components within the taxonomy created by Laszka et al. [145] form the basis for the categories used to annotate the policy documents

[9] `https://prodi.gy`

within this chapter, which are given below. (Shown in brackets are the actual labels used in the annotations.) Many of the elements of the taxonomy are highlighted by Akgul et al. [7] as being considered important to hackers (e.g. communication of the scope and payment rules). Examples of the taxonomy elements can be found in Appendix A.2.

- *In-scope areas (SCOPE-IN).* Paragraphs or statements that include an explicit definition of in-scope areas or assets that are considered to be part of the CVD programme.

- *Out-of-scope areas (SCOPE-OUT).* Content that includes the explicit definition of out-of-scope areas or assets that a hacker is forbidden to explore for the purpose of security research.

- *Eligible vulnerabilities (VULN-ELIGIBLE).* Definition of specific vulnerability types or classes that an organisation considers to be eligible for submission.

- *Non-eligible vulnerabilities (VULN-INELIGIBLE).* Specific vulnerability types or classes that are ineligible for submission. Typically, this includes vulnerabilities considered to be of low or no severity to the organisation, omission of best practices, or non-technical vulnerabilities (e.g. phishing of employees or customers). For example, Cross-Site Request Forgery (CSRF), vulnerabilities requiring outdated or otherwise unsupported web browsers, username or email enumeration, and the discovery of descriptive error messages are often marked as ineligible.

- *Deepening engagement with organisations (ENGAGEMENT).* As defined by Laszka et al. [145], this category of statement includes instructions to hackers as to how they can better 'engage' in the discovery of security vulnerabilities. This can include directions from the organisation to the hacker on how to set up dedicated test accounts, use specific testing domains, and configure user accounts to access hidden development / testing features.

- *Prohibited or unwanted actions (PROHIBITED-ACTIONS).* A hacker can take a number of actions during the process of vulnerability discovery that may be considered undesirable to the organisation. These include somewhat benign violations such as causing undue

load on production servers resulting from the use of automated vulnerability scanning tools. Organisations may also explicitly state violations that they consider to be malign, or are otherwise illegal, such as Denial of Service (DoS) attacks, accessing user or customer data (particularly if it contains Personally Identifiable Information (PII)), or attacking client systems.

- *Legal clauses (LEGAL).* Paragraphs or statements put forward by the organisation as part of the CVD policy may include a number of legal clauses relevant to the process of vulnerability discovery and the submission of subsequent reports. These include, but are not limited to, exemption from the Digital Millennium Copyright Act (DMCA) [241] during the discovery process, authorisation of research activity in accordance with the Computer Fraud and Abuse Act (CFAA) [239], and assignment of intellectual property rights upon the submission of a report.

- *Participation restrictions (PARTICIPANT-RESTRICTIONS).* Organisations commonly restrict previous or current employees, contractors, as well as their immediate family members, from participating in an organisation's CVD programme and claiming any subsequent rewards in order to prevent misuse of an individual's privileged knowledge of, or access to, company or client systems. Furthermore, as many organisations are subjected to the laws of the US, they are prohibited from awarding payouts to any individual currently located in, or ordinarily a resident of any sanctioned country [246], and, as such, they may prohibit participants from certain geographies. For US-based organisations, this includes any hackers residing in entities in Country Group E (e.g. Cuba, Iran and North Korea).

- *Submission guidelines (GUIDELINES-SUBMISSIONS).* Specific guidelines about the information required, or desired, to be included in vulnerability reports may be put forward by organisations. This also includes any specific instructions on how to submit either the vulnerability report or accompanying material (e.g. video recordings).

- *Disclosure guidelines (GUIDELINES-DISCLOSURE).* The programme policy may contain further information concerning the possibility of disclosure by the hacker, together

Type	Count
ORG	3,692
ASSET	3,663
VULN	3,374
STKHLDR	1,208
TECH	1,030
PRODUCT	662
SYMBOL	500
LAW	467
PI	454
SEVERITY	450
DATE	378
GPE	139
PROMO	132
PERSON	61
IP	55

Table 5.4: Summary of the entity types of the 16,262 entities annotated within the 12,526 sentences. Note that a sentence may contain multiple entities of the same type.

with guidelines that a hacker is asked to follow before disclosing any details (e.g. waiting a period of 90 days following report submission).

- *Reward evaluation (REWARD-EVALUATION).* If it is the case that separate bounty tables are not used by an organisation, any information pertaining to the potential rewards offered following the successful submission of a vulnerability will be outlined within the policy document. This includes paragraphs or statements that detail the value of a report in terms of CVSS score or vulnerability class. This includes any additional conditions that will materially alter the payout, such as: increased bounty payouts from high-quality reports, rewards offered to the second discoverer of a vulnerability (typically diminished or non-monetary), and bonuses awarded to repeat discoverers.

- *Company statements (COMPANY-STATEMENT).* Any paragraph or statement that includes general information about an organisation's posture towards security or the CVD programme in question.

In order to gain a deeper understand of the content included within policy documents, it is necessary to also explore the more fine-grained semantic features that are present. As such, it is useful to annotate the entities that existing within the content for later use in a Named

Entity Recognition (NER) model. Similar to the qualitative approach employed by Laszka et al. [145], we attempt to capture the generic components (entities) that form a policy document — with this being done at a sentence-level. We do not focus only on capturing security-related elements; rather, we identify more general information typically targeted by NER systems, allowing for broader insights to be drawn from the analysis. To this end, the following 15 unique entity types are defined.

- *Organisation name (ORG).* Any reference to an organisation, not necessarily the organisation responsible for the CVD programme. This is an entity type commonly included in NER datasets.

- *Asset (ASSET).* References to any assets owned by an organisation or other third-party stakeholder. This includes assets that are both in- and out-of-scope to hackers.

- *Vulnerability or attack class (VULN).* Specific vulnerabilities or attacks that are mentioned within by an organisation in any context.

- *Stakeholder (STKHLDR).* Any party that could be considered a stakeholder to the organisation, or a stakeholder of a third party. Examples include users, customers, employees, and contractors.

- *Technical reference or acronym (TECH).* Any reference to any non-branded technology, term or related acronym that is not inherently a security vulnerability or attack class. Examples include TLS, SSL, automated tools, and jailbreaks.

- *Product reference or acronym (PRODUCT).* References to a specific product or service produced by an organisation. Examples include iOS, Android, and Windows.

- *Currency symbol (SYMBOL).* To distinguish currency symbols from other acronyms or entities of interest, it is necessary to identify any symbols as a separate entity. Due to the prevalence of Web3-related programmes in the collected data, it is particularly useful to be able to correctly identify the plethora of cryptocurrency symbols that are present throughout. For programmes focused on protecting Web3-based assets, it is common

for them to offer rewards in a specified cryptocurrency, rather than USD equivalent. For example, payouts may be issued in USDT (Tether stablecoin) or LUNA (the now defunct Terra stablecoin).

- *Law or legal term (LAW).* Any legal clauses or terms (aside from those relating to intellectual property and proprietary information, see the IP type) that are present in a given sentence. This includes references to specific laws and regulations (e.g. DMCA, GDPR, and CFAA), and other legalese (e.g. good faith).

- *Personal information (PI).* References to any information that may be considered personal or personally identifiable information such as phone numbers, email addresses, credit card numbers, and physical addresses. Note that this entity does not refer to the underlying information, but, rather, references to the types of information that are mentioned in the data. For example, the statement "please refrain from accessing or retaining any customer credit card or personal contact information" contains "credit card" and "personal contact information" as entities of the type PI.

- *Severity rating (SEVERITY).* Any reference to the severity of a vulnerability.

- *Date (DATE).* Dates or time periods mentioned within a sentence. This is an entity type commonly included in NER datasets.

- *Geopolitical entity (GPE).* The identification of any geopolitical entities is useful for further analysis into mentions of legal jurisdictions (at either the country or state level), sanctioned countries, locations of registration, and laws specific to certain countries or other GPEs (e.g. the European Union). This is an entity type commonly included in NER datasets.

- *Promotional event (PROMO).* Some organisations will routinely encourage hackers to search for vulnerabilities in recently launched products or services through the use of promotional events, and will often include bonuses on payouts for any vulnerabilities successfully discovered and reported. Furthermore, in an attempt to increase the interest of hackers in a programme (perhaps after the operators notice a drop in the perceived

motivation of hackers), some organisations will use limited time events with increased bounties or special rewards to encourage hackers back. As such, it is of interest to identify any references to these promotional events.

- *Person (PERSON).* Organisations may include the names of the operators, or references to successful hackers (either by name or alias) in the policy document or published update. This is an entity type commonly included in NER datasets.

- *Intellectual property and proprietary information (IP).* References to any terms relating to intellectual property that exist within a sentence. For example, hackers may be required to grant irrevocable and unlimited licenses of their work (the vulnerability reports) to an organisation in order to be eligible for a payout. Although separate to intellectual property law, this type of entity also includes related references to proprietary information and trade secrets.

5.4 Natural Language Processing for Policies

The NLP models, alongside a collection of more traditional pattern matching models, are discussed within this section, and provide the basis for the results presented throughout the remainder of the paper.

Three distinct modelling approaches were used to extract textual information from the policy documents. As described in Section 5.4.1, a supervised paragraph-level classifier was used to categorise the structural elements of a document. Two separate approaches were used in conjunction to extract sentence-level features from each document. As such, Section 5.4.2 describes the use of a deep learning based Named Entity Recognition (NER) system to detect pre-defined entity types and Section 5.4.3 outlines the use of traditional pattern-matching techniques to detect consistent entities.

A summary of the end-to-end data pipeline used for document processing is presented in Section 5.4.4, and includes a description of the inference and extraction steps necessary to analyse the CVD policy documents. A discussion of the factors affecting the replicability and

Category	F-1
COMPANY-STATEMENT	74.0
REWARD-EVALUATION	78.6
SCOPE-IN	55.2
VULN-ELIGIBLE	59.7
VULN-INELIGIBLE	60.7
PROHIBITED-ACTIONS	38.3
GUIDELINES-SUBMISSIONS	75.4
SCOPE-OUT	31.6
ENGAGEMENT	50.0
GUIDELINES-DISCLOSURE	58.1
LEGAL	84.6
PARTICIPANT-RESTRICTIONS	40.0

Table 5.5: Results of the multi-label text classification models, per-class averaged F-1 score across the 5-folds

reproducibility of the study is discussed in Appendix A.1, and further details can be found in the GitHub repository for the study[10].

5.4.1 Text classification

In order to classify a given piece of text into the pre-defined categories (as described in Section 5.3.2), a multi-label textual classification model was constructed using the spaCy[11] NLP library [213]. A multi-label model is needed to accommodate the possibility of a piece of text simultaneously belonging to multiple categories. As the annotated corpus is formed of paragraphs, it is reasonable to expect that a paragraph may contain information pertinent to one, several, or none of the categories defined by Laszka et al. in their taxonomy of policy components [145]. For the sake of brevity, the architecture and hyper-parameter search for the text classification model are not discussed in depth. However, further details are provided alongside the model in the project repository[12]. In essence, a standard spaCy pipeline with an additional 'textcat_multilabel' component was used for the model.

For training and testing, 5-fold cross-validation was performed using the annotated data. An average macro F-1 score of 0.589 was achieved. (The per-class results are shown in Ta-

[10] All material relating to the study is found at: `https://github.com/walshe96/cvd-policy-documents`
[11] `https://spacy.io`
[12] `https://github.com/walshe96/cvd-policy-documents`

ble 5.5.) At present, a considerable class imbalance exists within the data that may impact the performance of classifiers trained over the dataset. Future work may help to address this imbalance and alleviating the poor performance seen within some classes through the annotation of additional 'hard' positive examples in under-represented categories or through more extensive modifications to the model architecture [150].

For a given passage of text, the output of the model is in the form

$$\hat{\boldsymbol{Y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N]^T \tag{5.1}$$

such that

$$\hat{y}_i \in [0, 1] \quad \forall i \in \{1, \ldots, N\}$$

Here, $\hat{\boldsymbol{Y}}$ is the output vector containing elements $\hat{y}_i$, corresponding to the estimated probability that the input belongs to the i^{th} class for each of the N classes. Note that the individual class probabilities are not normalised across all classes, and, as such,

$$\sum_N \hat{y}_i \in [0, N] \tag{5.2}$$

5.4.2 Named Entity Recognition (NER)

An NER model was trained on the annotated sentences to detect the entity types described in Section 5.3.2. Unlike traditional pattern matching techniques (e.g. through the use of gazetteers or hand-crafted features [176, 175]), the NER models employed can leverage large pre-trained transformer [250] (see BERT [62] and RoBERTa [154]) and non-transformer (CNN and LSTMs [268]) deep learning based NLP architectures to identify entities within sequences of word, wordpiece, or character-level embeddings [36]. The use of contextual embeddings [167] further benefits the NER models [149], improving upon their ability to generalise and detect out-of-sample entities when compared to traditional techniques [132]. As with the classification model outlined in Section 5.4.1, the spaCy NLP library is also used to construct the NER model.

Type	F-1
ORG	85.7
ASSET	78.7
VULN	75.1
STKHLDR	80.5
TECH	74.1
PRODUCT	67.5
SYMBOL	81.3
LAW	78.2
PI	76.8
SEVERITY	86.2
DATE	78.9
GPE	72.8
PROMO	55.7
PERSON	23.2
IP	74.5

Table 5.6: Exact-match results of the NER model, per-type averaged F-1 score across the 5-folds

As before, the architectural details of the NER model are omitted for the paper, however further details are provided alongside the model in the project repository[13]. In essence, a standard spaCy pipeline with an additional 'NER' component was used for the model.

For training and testing, 5-fold cross-validation was performed using the annotated data. The per-type results are shown in Table 5.6. As is typical in NER tasks, there is a dearth of entities in comparison to negative samples (tokens that do not belong to any of the pre-defined types) in the dataset. Little can be done to resolve the resulting imbalance between entities and non-entities that exists within the data. However, selective annotation may be used to help address the imbalance that exists between entity types by selecting 'hard' samples containing low-count entities for further human annotation.

5.4.3 Pattern recognition

For entities that are consistent in their format, the use of regular expressions for pattern recognition is preferred due to the time saved in comparison to developing supervised models and annotating data. In the datasets collected for this study, there are a number of entities recognised using regex. These are listed below.

[13] `https://github.com/walshe96/cvd-policy-documents`

- *Email addresses.* Occasionally, organisations will provide an email address by which hackers are able to disclose the details of a vulnerability, or to be used as a line of communication between hackers and the operators.

- *URLs.* Any domain prepended with http, ftp, https and www is identified as part of the URL extraction process.

- *Domains.* Any remaining domain-like objects after URL extraction are detected using a further domain regex pattern that relaxes the conditions on the prepended information.

- *IPv4 addresses.* Organisations may specify IP addresses within their policy documents or scope tables. Due to time constraints, only an IPv4 regex was implemented. (However, it was found that the number of IPv6 addresses in the data was small.)

- *Bounty values and ranges.* Organisations will common publish bounty values, or a range of values, that may be paid out after the successful submission of a vulnerability report. For a range of currencies (US dollar, British pound, Euro, and Yen), these values and ranges are identified.

- *Markdown tables rows.* For organisations that publish documents in markdown format, data may be contained within markdown tables. As the first stage of table information extraction, each row and the corresponding data is identified.

- *Markdown tables.* Groups of continuous markdown table rows are identified as part of the second stage of the markdown table extraction.

5.4.4 Document Pipeline

The overall document processing pipeline is displayed in Figure 5.1 and can be applied to all of the collected datasets, including data in markdown and unidiff formats. The process is as follows.

1. Cleaning. Basic cleaning is applied to the input text, including the removal and separate storage of any unidiff marks, and removal of undesirable unicode characters.

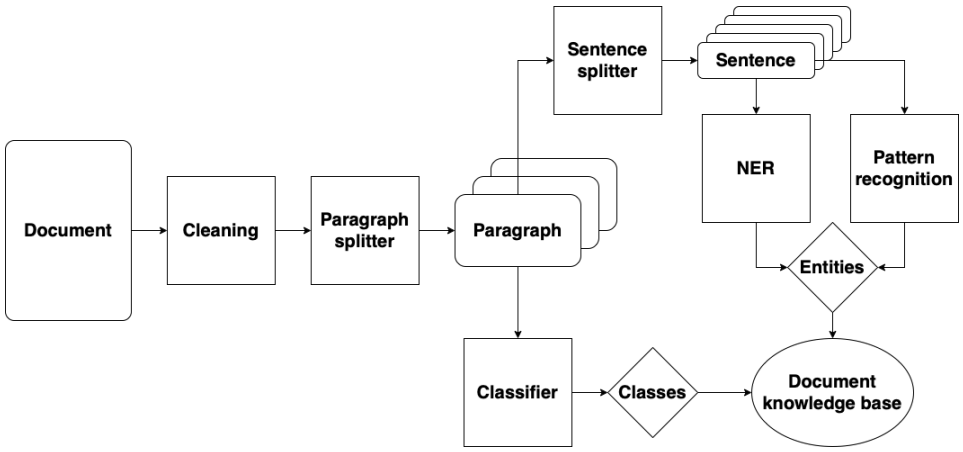

Figure 5.1: Data-processing pipeline applied to documents

2. Paragraph splitting. Documents are split over double newline characters and stored as a list of paragraphs to be individually processed.

3. Classifier. Each paragraph is classified using the model outlined in Section 5.4.1 and the output vector stored in the document knowledge base.

4. Sentence splitter. Each paragraph is further decomposed into sentences using the NLTK sentence tokenizer.

5. NER. The entity recognition model outlined in Section 5.4.2 is used to identify and extract 15 entity types before storage in the document knowledge base.

6. Pattern recognition. The regular expression based pattern recognition techniques introduced in Section 5.4.3 are used to identify and extract a further seven types of entities before storage in the document knowledge base.

7. Document knowledge base. Information derived and extracted from an input document is stored in a knowledge base. This centralised repository created for each document is used throughout the analysis.

5.5 Formal constraints

As discussed in Section 5.1, institutions, as defined by North [177], are composed of formal and informal constraints. Pursuant to the first research question, we explore the current usage of formal constraints in CVD policy. As discussed by Hodgson [117], in order to avoid imprecision, we restrict our definition of formal constraints to government-defined legal constraints [260].

The methodology outlined in Section 5.4.2 allows for the identification of formal constraints using the LAW and IP entity tags that may be assigned to sentences as part of the NER model. Furthermore, from paragraphs classified as containing legal clauses, analysis of the formal constraints promulgated is performed.

5.5.1 Exploring the formal constraints

A consideration of the sources of constraints

A hacker can be subject to a litany of legal obligations, arising from a number of unilateral contracts, en route to participating in an organisation's CVD programme. For the hacker, they must not only be aware of the rights and obligations outlined in each contract, but they must also understand the order of precedence in the event of conflict. For example, from Bugcrowd we have: "in the event of a conflict within the Legal Terms, the order of precedence shall be, in order of highest priority to lowest priority: any Bounty Brief, the Researcher Terms and Conditions, these Terms of Service, the Privacy Policy and then any other terms that comprise the Legal Terms"[14]. When utilising a bug bounty programme hosted on a platform, hackers may be legally bound to, or affected by any of the following contracts:

- Platform General Terms and Conditions. Applicable to any individual that makes use of a platform's websites[15].
- Platform User Terms and Conditions. Applicable to hackers that create, and make use of, a user account on a given platform[16].

[14]From `https://www.bugcrowd.com/terms-and-conditions/`
[15]For an example, see `https://www.hackerone.com/terms/general`
[16]For an example, see `https://www.hackerone.com/terms/finder`

- Platform Customer Terms and Conditions. Hackers may be affected by the contracts between the platform and a hosted organisation. For instance, on Bugcrowd, in the event of conflict, the Customer Terms and Conditions may supersede any 'Terms of Use' agreements from organisations that the hacker may be required to accept on the route to accessing the target systems[17].

- Programme CVD policy. Dependent on the content of the policy, the programme policy may also act as a legally binding document between the hacker and an organisation.

- Organisation Terms of Use / Service or EULAs. Access to, or use of, the target organisation's assets (including websites, source code, software / hardware products, etc.) may also involve the hacker being in agreement to additional binding documents.

- Non-Disclosure Agreements (NDAs). Upon the invitation to a private BBP, or after the disclosure of a vulnerability report to an organisations, a hacker may be asked to sign an NDA.

While each of the aforementioned contracts may contain terms that contribute to the formal constraints imposed upon a hacker, for the purpose of the work presented in this chapter we only consider those that arise from the policy documents. However, the number of contracts involved serves as an indication as to the complexity of the legal situation that a hacker must navigate.

Generic constraints

Table 5.7 provides a breakdown of the inclusion of formal constraints by type of hosting method. As noted by Elazari in a 2018 presentation [69], organisations will frequently make vague or generic references to the laws with which hackers must comply. It was found that there are 691 (non-unique) phrases associated with a generic reference to formal constraints mentioned within the policy documents across all organisations, with the most common being: "applicable laws" (212), "State laws" (122), "applicable Federal, State, and local laws" (27), "local laws" (19), and "laws or regulations of any country" (8).

[17] From `https://www.bugcrowd.com/termsandconditions/`

	Any		Generic		Specific	
	#	%	#	%	#	%
Platform	351	28.2%	268	21.6%	243	19.6%
Stand-alone	57	10.3%	57	10.3%	6	1.2%
Bugcrowd	158	70.5%	146	65.2%	140	62.5%
HackerOne	117	22.2%	103	19.6%	37	7.0%
Immunefi	40	14.3%	7	2.5%	34	12.1%
YesWeHack	29	96.7%	7	23.3%	29	96.7%
Intigriti	5	6.6%	3	4.0%	3	4.0%
HackenProof	2	3.8%	2	3.8%	0	0%
RedStorm	2	33.3%	2	33.3%	0	0%

Table 5.7: Number of programmes that contain either generic or specific formal constraints

For stand-alone programmes, it was observed that there exists a tendency to make generic references to formal constraints. Of those specifying any formal constraints, 57 (100%) included some generic references, while only 6 (8.8%) included specific constraints. Across all programmes on platforms, of those that include formal constraints, 268 (74.7%) included generic references and 243 (67.7%) included specific references. Further differences in the proclivity to include formal constraints are explored in Section 5.7. Significant differences exist between the platforms, particularly between the largest non-Web3.0 specific platforms (HackerOne and Bugcrowd). While the specification of generic constraints varies considerably, it is found that they are included on a majority of all Bugcrowd programmes.

Specific constraints

Table 5.7 also shows a breakdown of the usage of specific formal constraints used by organisations. The Computer Fraud and Abuse Act of 1986 (CFAA) [239] and the Digital Millennium Copyright Act of 1998 (DMCA) [241] are the two most common specific constraints mentioned within CVD programme policy documents, with 279 and 274 respective mentions. Know Your Customer (KYC) laws are mentioned 78 times within the documents. There are 16 references to EU regulation 2016/679, otherwise known as the General Data Protection Regulation (GDPR) [231]. There are also references to several laws specific to California: Section 502(c) of the California Penal Code [226, 59] (4), the California Consumer Privacy Act of 2018 (CCPA) [186, 147] (3), and the California Privacy Rights Act of 2020 (CPRA) [92] (1).

Specific to the UK, the Computer Misuse Act 1990 [187] is mentioned once. Despite the global nature of CVD programmes, it is perhaps surprising to see references to specific laws and regulations predominantly confined to the US and Europe. However, this may stem from the English language bias of the policies analysed (a further discussion of limitations is presented in Section 5.8).

As noted in Section 5.5.1, there exists significant differences in the inclusion of specific constraints between stand-alone programmes and those hosted on a bug bounty platform, with very few stand-alone programmes (1.2%) including references to specific formal constraints. The French platform YesWeHack is found to have the greatest proportion of programmes (96.7%) that include specific formal constraints.

5.5.2 Analysis

It is clear from the results that hackers are presented with, and are subject to, a wealth of applicable legal agreements and language. As discussed by Elazari [69], the inclusion of generic formal constraints (e.g. "you must comply with all relevant local, state, national or international laws") places a considerable burden on the hacker to understand, across a number of jurisdictions, the laws and regulations that they must navigate during the process of vulnerability discovery and disclosure. Furthermore, unresolved conflicts may exist within the myriad applicable contracts, potentially exposing hackers to civil or criminal liabilities [69].

The advent of cloud computing, and continued employment of the technology for the storage of personal data [6, 108], may also add further complexity to the process of lawful vulnerability research. Hackers may have difficulty identifying applicable laws if the locations of data centres are not made obvious [48]. Irrespective of the location of the hackers, organisations should endeavour to clearly communicate the applicable jurisdictions to those wishing to participate in a programme.

An earlier study by the National Telecommunications and Information Administration (NTIA) found that the threat of legal action impacted the disclosure decisions of 60% of surveyed hackers [237]. It is, perhaps, necessary to further simplify and standardise the legal language used within CVD policy documents in order to clarify the legal position afforded to

hackers conducting legitimate, 'good faith' security research.

When considering the specific constraints referenced in policy documents, the CFAA and DMCA are most prevalent, indicating the relevance of US Federal and copyright laws to hackers across the world, and highlights the importance of providing authorisation to hackers under both acts in Safe Harbor clauses [72]. Previous research has highlighted the fear amongst hackers at the legal risk brought about by the CFAA and DMCA when conducting security research [227]. Zhao et al. [272] encourage a change in regulatory policies (such as the DMCA) to help protect those conducting legitimate security research.

As noted in Section 5.5.1, aside from the individual programme policies published by programme operators, the users of bug bounty platforms (both the customer organisations and the hackers) will be subject to the particular policies set forth by the platform operator. Depending on the nature of the services provided by the platform, these may supersede those of the programme. For example, the payments for bounties awarded by programme operators on the HackerOne platform are facilitated by HackerOne[18]. As a US company, they are unable to provide payments to any sanctioned individuals, or individuals normally resident in a country under US sanction (see Section 5.1 and Section 3.1). In the case of issuing payments, the participant restrictions of the payment facilitator (e.g. HackerOne) will have precedent over individual programme policies, unless an organisation chooses to bypass the platform and issue payments by other means. This may cause an absence of payment restriction clauses for programmes hosted on US-based platforms. This is perhaps reinforced by the views of an operator expressed in Section 2.7.2, who noted the ease that comes with the integrated payment systems on a platform as the burden of regulator compliance (in the context of payments) is shifted away from the programme operator.

Certain platform policies may also exist alongside those published by the programme operator. Both HackerOne and Bugcrowd publish hacker 'codes of conduct'[19] that outline acceptable behaviour when participating in programmes and communicating with other parties on the platform. Failure to comply with the code may result in punishment, yet the severity of the

[18]Policy found at: `https://www.hackerone.com/disclosure-guidelines`
[19]For an example, see: `https://www.hackerone.com/policies/code-of-conduct`

platform punishments in many cases appears incongruous with the potential legal ramifications following investigation by the programme operator. For example, it may take four violations of 'service degradation / unsafe testing' on HackerOne before an account is permanently banned. Hackers should be aware of the policies that exist on bug bounty platforms, and future research should endeavour to consider the interplay between platform and programme policies.

When considering the specification of formal constraints, there is a clear disparity between platform-based and stand-alone programmes (see Table 5.7), with platform-based programmes more likely to include generic and specific references to formal constraints. As reported in Chapter 2, the operators of bug bounty programmes note that the operation of a programme on a platform often represents a more mature security activity than that of many stand-alone programmes. The relative immaturity of many stand-alone programmes in comparison to platform-based programmes may explain, in part, the lack of attention placed on including constraints in a programme's policy. However, this does not explain the large variation between platforms.

5.6 Informal constraints

5.6.1 Exploring the informal constraints

Before discussing the variation in the institutional elements that organisations include within CVD programme policies, it is useful to briefly describe the characteristics of the policy documents in the context of the results presented by Laszka et al. [145].

Shown in Figure 5.2 is a comparison of the lengths of policies collected from bug bounty platforms to stand-alone programmes. From the policies collected, it is clear that those from platforms (average of 1,165 words) are typically far longer than those from stand-alone programmes (average of 370 words). In the analysis of HackerOne programmes by Laszka et al. [145], the authors found an average length of 481 words, a minimum of 72 words, and a maximum of 1,744 words. In comparison, focusing on only HackerOne programmes, it is found that the average length is 964 words, a minimum of 6 words (from the now defunct MixMax programme: "See policy and bounties at [URL]", included URL redacted using the

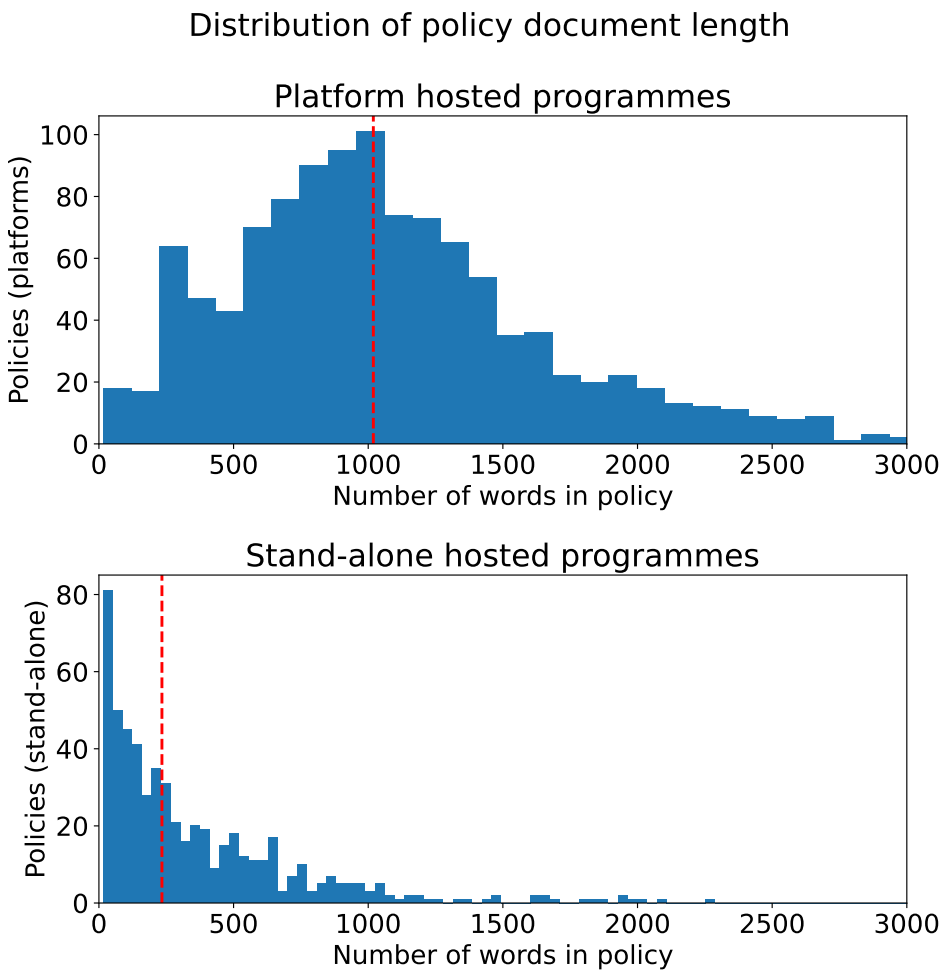

Figure 5.2: Length of programme policies, in words, for programmes hosted on a platform or those that are stand-alone. The dashed lines represent the median values of 1019 words and 234 words, respectively. The averages are 1165 words and 370 words, respectively.

data pipeline), and a maximum of 4,442 words.

To continue the comparison with the work by Laszka et al. [145], Figure 5.3 shows the distribution of Flesch reading ease scores [80, 133] by policy length. The Flesch reading ease score is a popular heuristic that represents the readability of a passage of text using a 0–100 scale: a score closer to 100 indicates a simpler passage of text; a score below 30 represents a difficulty appropriate for an individual with a university degree [80]. The average score for the policy documents is 35.32, which is lower than the average score of 39.6 reported in Laszka et al. [145]. Furthermore, 550 policies (approximately a third) have a score lower than 30, indicating a slight increase in the proportion of policies (up from 23.4% [145]) that can be

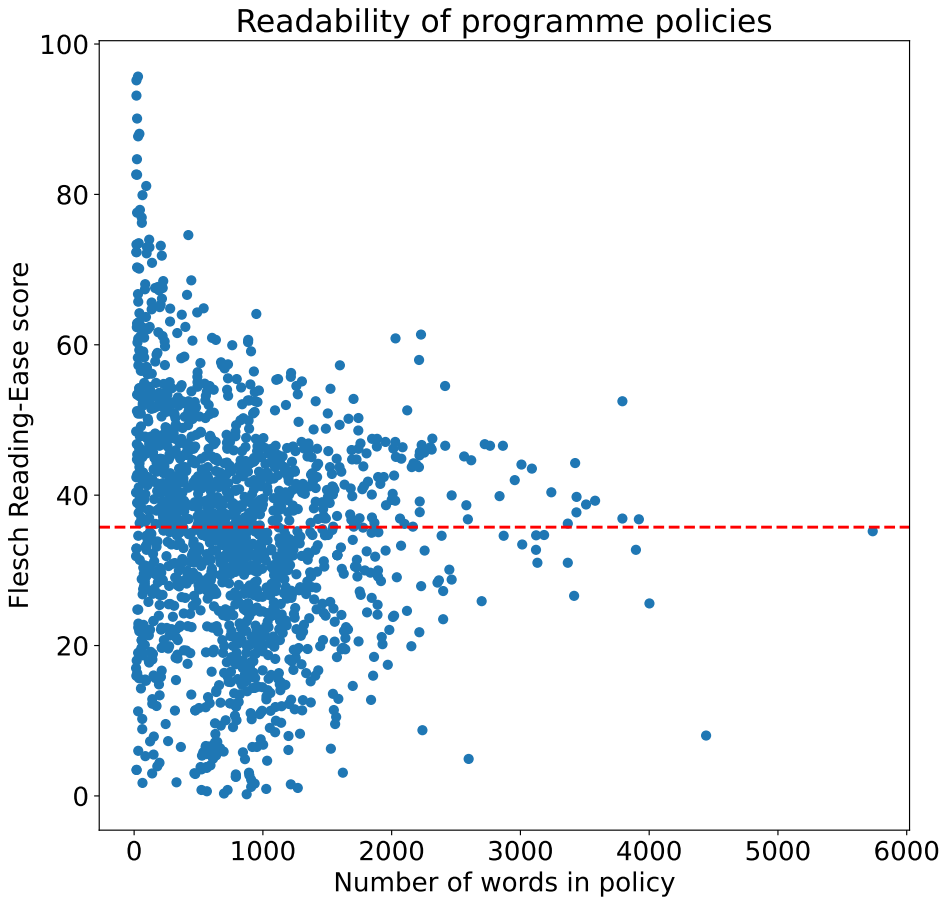

Figure 5.3: Readability scores of all programme policies, as determined by the Flesch reading ease formula [80, 133], plotted against policy length. The dashed line represents the median readability value of 35.75 (average of 35.32).

considered somewhat difficult to read.

In order to quantify the variation in the informal constraints communicated to hackers, we consider the output of the multi-label classification model (as described in Section 5.4.1) as a measure of the presence of constraints (both informal and formal) that are contained within a given paragraph. Table 5.8 shows a count of the dominant labels (those of the highest likelihood) assigned to the paragraphs of all policy documents. A measure of the variance of the constraints contained within the paragraphs is calculated by dimensionally reducing the 12-dimensional output vector for all paragraphs to a 1-dimensional space using a combination of Principle Component Analysis (PCA) [1] and t-SNE [249]. The variance is then computed amongst all points within a given category as dictated by the dominant labels. The lower the

	All		Platform		Stand-alone	
Dominant label	#	Var	#	Var	#	Var
COMPANY-STATEMENT	7368	1470.7	5742	1455.1	1626	1526.2
REWARD-EVALUATION	3225	2602.8	2829	2526.6	396	2523.6
SCOPE-IN	2454	1202.2	2310	1241.1	144	488.4
VULN-ELIGIBLE	1372	1061.7	1278	1052.8	94	1126.0
VULN-INELIGIBLE	1335	1348.3	1164	1351.0	171	1326.8
GUIDELINES-SUBMISSIONS	1143	1437.5	751	1506.2	392	988.1
SCOPE-OUT	987	1034.4	940	1053.9	47	651.1
PROHIBITED-ACTIONS	937	1714.3	741	1730.0	196	1644.7
ENGAGEMENT	799	440.3	728	442.1	71	405.7
GUIDELINES-DISCLOSURE	421	825.5	323	888.7 7	98	608.5
LEGAL	368	98.2	310	88.0	58	155.1
PARTICIPANT-RESTRICTIONS	105	419.7	89	415.1	16	364.0

Table 5.8: Counts of the dominant label assigned to all policy paragraphs using the multi-label classifier and the variance of the dimensionally reduced (using t-SNE) content vectors. A lower variance for a given label indicates that policy paragraphs in that category are less likely to contain information pertaining to other labels, thus are more focused.

variance for a given category, the more similar that paragraphs (of the given category) are across all policy documents and the less likely the paragraphs are to communicate a variety of different types of constraint.

In order to provide hackers with clear and standardised policy documents, it is optimal to decrease the variance of the institutional elements communicated throughout. It was found that informal constraints communicated through REWARD-EVALUATION and PROHIBITED-ACTIONS have the most variance of all label categories. It was also found that the formal constraints communicated through LEGAL paragraphs show the least variance in the corpus, thus indicating that LEGAL paragraphs (as defined in Section 5.3.2 and by Laszka et al. [145]) are somewhat standardised and are less likely to contain a mixture of constraints.

Table 5.8 also allows for comparisons between the institutional elements found in the policies of stand-alone and platform-based programmes. Most notable is the difference in variation in the paragraphs concerning the communication of scope (SCOPE-IN and SCOPE-OUT) and submission guidelines (GUIDELINES-SUBMISSIONS). In the three categories, stand-alone programmes exhibit less variation, suggesting greater similarity across paragraphs. This may suggest that stand-alone programmes have less complex scope specifications and guidelines,

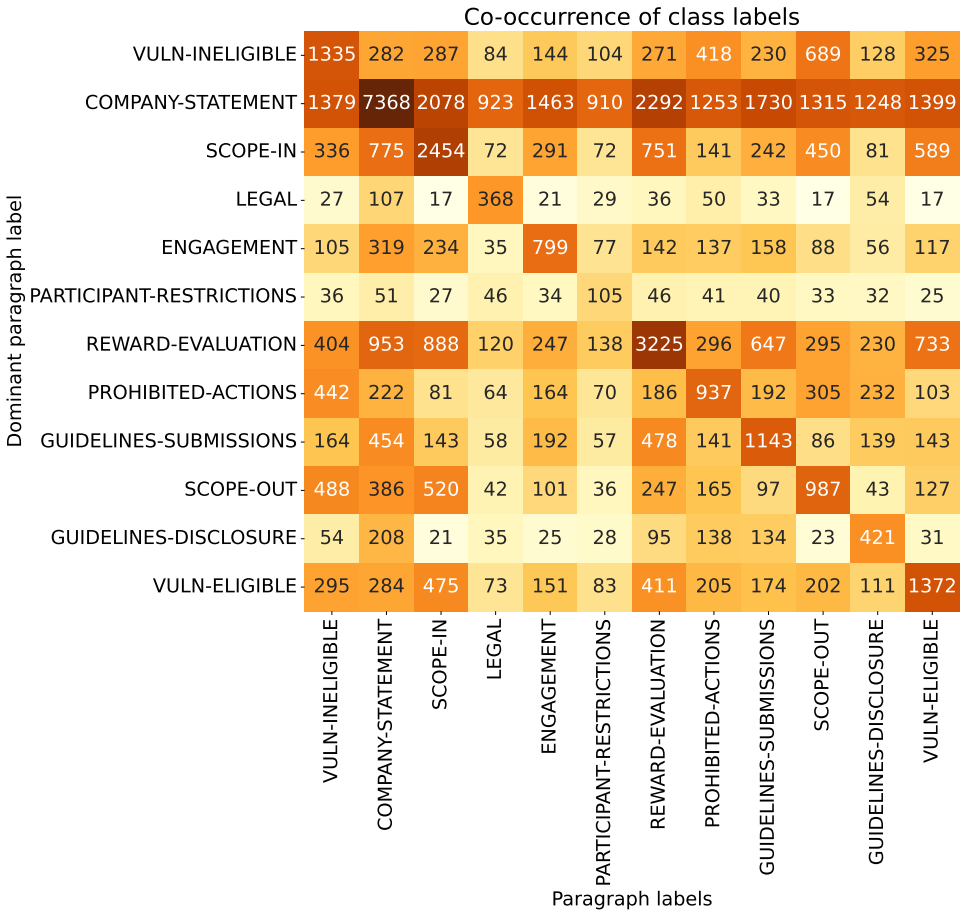

Figure 5.4: Relationship between the dominant class label from the multi-label text classification and top assigned class labels for a given policy paragraph, for all programmes. Colours are expressed using the log of the counts.

and are therefore somewhat standardised in comparison to platforms based programmes.

Figure 5.4 shows the co-occurrence of all labels with the dominant label for all paragraphs from the output of the classification model. For example, of the 1,335 paragraphs that primarily convey information about the vulnerabilities that are ineligible (VULN-INELIGIBLE as the dominant label), there are 689 that also communicate out-of-scope assets or domains (SCOPE-OUT as a non-dominant label). Shown in Figure 5.5 and Figure 5.6 is the label co-occurrence for platform-based and stand-alone programme, respectively. In order to aid comparison, the relationship between dominant labels and corresponding most common second labels is displayed in Table 5.9. Although there are many similarities between the the two types of programme, the prevalence of COMPANY-STATEMENT as a second label for stand-alone

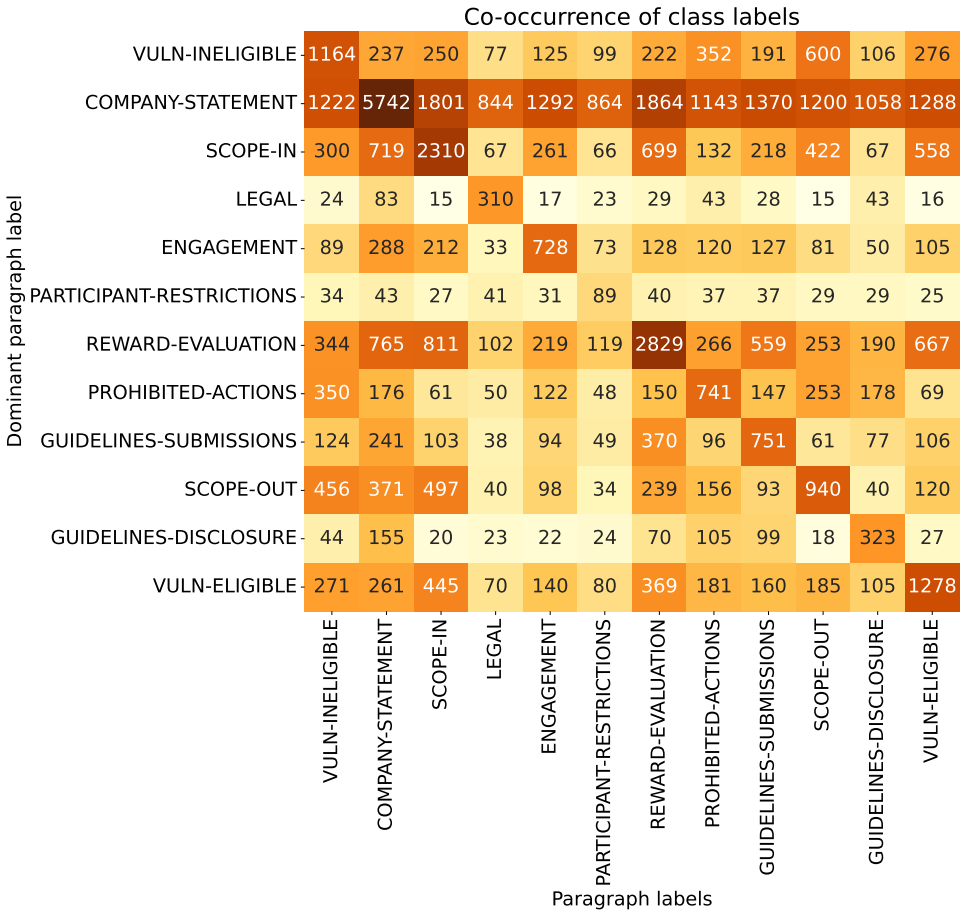

Figure 5.5: Relationship between the dominant class label from the multi-label text classification and top assigned class labels for a given policy paragraph, for programmes hosted on a platform. Colours are expressed using the log of the counts.

programmes may suggest more generic policies.

5.6.2 Analysis

A study of information security policies of the top 200 US universities (of which 54% had a publicly accessible policy) by Weidman and Grossklags [259] considered, among other factors, the readability and length of policy documents. Interestingly, they find that the readability of information security policies is far less than the policies found in CVD programmes (average Flesch Reading Ease score of 12.54 versus an average of 35.3). They note that a score of 0–30 is suitable for university graduates, and therefore may be difficult to parse for the individuals at a university without a degree. A score of 30–50 is suitable for university students (non-

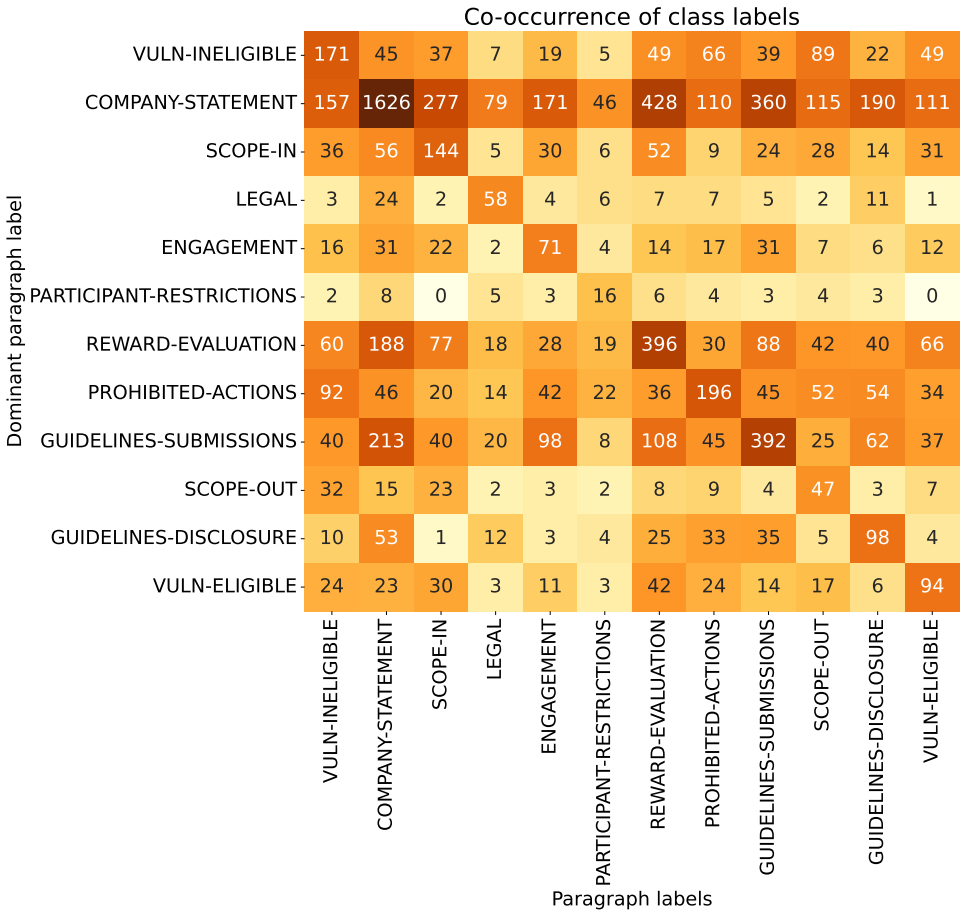

Figure 5.6: Relationship between the dominant class label from the multi-label text classification and top assigned class labels for a given policy paragraph, for stand-alone programmes. Colours are expressed using the log of the counts.

graduates). However, as noted by Akgul et al. [7], many hackers have not reached this level of educational attainment. Furthermore, many hackers do not have English as a first language [99], perhaps adding to the difficulty of understanding the information contained within CVD policies. The commonality in relative readability (given the target audience), or lack thereof, of university information security policies and CVD programme policies suggests that the communication of technical policies may be ill suited for the audience.

Analysing the institutional elements reveals that certain informal constraints have significant variation between programmes. While this may be expected of scope definitions, company statements and specific access instructions (e.g. those in the ENGAGEMENT category) — all of which will vary due to the peculiarities of a particular programme and operating organisation

	Platform		Stand-alone	
Dominant label	Second label	%	Second label	%
VULN-INELIGIBLE	SCOPE-OUT	51.5	SCOPE-OUT	52.0
COMPANY-STATEMENT	REWARD-EVALUATION	32.5	REWARD-EVALUATION	26.3
SCOPE-IN	COMPANY-STATEMENT	31.1	COMPANY-STATEMENT	38.9
LEGAL	COMPANY-STATEMENT	26.8	COMPANY-STATEMENT	41.4
ENGAGEMENT	COMPANY-STATEMENT	39.6	GUIDELINES-SUBMISSIONS	43.7
PARTICIPANT-RESTRICTIONS	COMPANY-STATEMENT	48.3	COMPANY-STATEMENT	50.0
REWARD-EVALUATION	SCOPE-IN	28.7	COMPANY-STATEMENT	47.5
PROHIBITED-ACTIONS	VULN-INELIGIBLE	47.2	VULN-INELIGIBLE	46.9
GUIDELINES-SUBMISSIONS	REWARD-EVALUATION	49.3	COMPANY-STATEMENT	54.3
SCOPE-OUT	SCOPE-IN	52.9	VULN-INELIGIBLE	68.1
GUIDELINES-DISCLOSURE	COMPANY-STATEMENT	48.0	COMPANY-STATEMENT	54.1
VULN-ELIGIBLE	SCOPE-IN	34.8	REWARD-EVALUATION	44.7

Table 5.9: Relationship between the dominant institutional elements and the second most prevalent element within each paragraph.

— a lack of commonality in the requirements for rewards, instructions for report submission and methods by which reports are assessed may lead to hackers missing the nuances of a particular programme.

5.7 Institutional analysis

5.7.1 An institutional analysis of policies

The results in Table 5.10 show the proportion of policy documents that contain each of the institutional elements of interest. Almost all organisations (98.7%) include general information about their security posture or CVD programme (COMPANY-STATEMENT) within the programme policy document. A majority of organisations will include information about the assets that are considered to be in-scope (74.2%) and out-of-scope (65.8%); however, this does not account for organisations that decide to provide scope details in a separate scope table. As noted in Section 2.7.3, controlling the scope of a programme is recommended by operators as a countermeasure to high volumes of low-quality reports. Perhaps of the greatest concern, a minority of organisations provide details of legal considerations (38.6%) and participant restrictions (31.7%) within the policy, requiring hackers to search elsewhere for information pertaining to such details (or otherwise remain in the dark).

	All		Platform		Stand-alone	
Institutional element	#	%	#	%	#	%
COMPANY-STATEMENT	1724	98.7	1139	99.2	585	97.7
REWARD-EVALUATION	1447	82.8	1041	90.7	406	69.5
GUIDELINES-SUBMISSIONS	1335	76.4	918	80.0	417	67.8
VULN-INELIGIBLE	1314	75.2	1008	87.8	306	51.1
SCOPE-IN	1297	74.2	996	86.8	301	50.3
SCOPE-OUT	1150	65.8	918	80.0	232	38.7
PROHIBITED-ACTIONS	1120	64.1	858	74.7	262	43.7
VULN-ELIGIBLE	1102	63.1	874	76.1	228	38.1
GUIDELINES-DISCLOSURE	991	56.7	698	60.8	293	48.9
ENGAGEMENT	965	55.2	709	61.8	256	42.7
LEGAL	675	38.6	538	46.9	137	22.9
PARTICIPANT-RESTRICTIONS	554	31.7	477	41.6	77	12.9

Table 5.10: Proportion of policy documents that contain at least one example of a given institutional element as defined by the classification labels.

Across all institutional elements, a higher proportion of platform-based programmes contained a given element in comparison to stand-alone programmes, thus further demonstrating the disparities between the two in the context of the information contained within policy documents. Perhaps of particular concern are the differences between the proportion of programmes that include information on in-scope assets (86.8% versus 50.3%), out-of-scope assets (80.0% versus 38.7%), and prohibited actions (74.7% versus 43.7%). Ambiguity can arise without clearly-defined boundaries, and may lead to hackers accidentally performing undesirable actions (see Section 2.5).

5.7.2 Analysis

The inclusion of specific institutional elements varies considerably between programmes; the difference is particularly evident between stand-alone programmes and those hosted on a bug bounty platform (see Table 5.10). Although policy documents are used to help hackers understand the constraints that govern their actions, many organisations fail to explicitly define the gamut of formal and informal constraints with which they wish hackers to comply. The lack of information available to hackers, or lack of clarity within the information provided, may result in the submission of sub-optimal vulnerability reports, exacerbating the issue of low-quality reports facing many organisations [9].

Furthermore, when considered a legally binding contract, the absence of certain information, or ambiguity of important information, may introduce additional risks and liabilities for organisations and hackers alike [208]. For instance, the absence of sufficient Safe Harbor clauses weakens the enforceability of the contract [79], while also exposing hackers to risks due to a lack of authorisation. Operators in Section 2.7.3 noted the importance for an organisation to have the backing of an in-house legal team prior to setting up a programme. Given the significance of the information contained within a policy document, it may be advantageous for the CVD programme operations team to work closely with in-house counsel when developing or refining policies. However, unless there exists the required legal background, it may be prudent for an organisation to seek external advice.

Throughout the results presented in this chapter, a common theme is that of a lack of information communicated via the policies of stand-alone programmes in comparison to those belonging to programmes hosted on bug bounty platforms. Previous qualitative work has focused on exploring the challenges faced by the operators of both programme types, revealing significant differences in operating characteristics. Through the use of surveys and interviews, Al-Banna et al. [9], and the results of Chapter 2, reveal that organisations may choose to make use of a bug bounty platform, such as HackerOne, where there is not the required in-house knowledge to set up and run a CVD programme. Instead, individuals from the platform assist the organisation in setting scopes, setting bounties, creating policies, managing the triage of incoming reports, and will use similar organisations or existing norms as a point of reference. Having the assistance of the platform operators may explain, in part, the significant differences between platform-based and stand-alone programmes, as organisations on a platform have guidance far more readily available. It may also account for the lower variance exhibited across certain institutional elements, as certain types of information may be more homogeneous across the same bug bounty platform.

Furthermore, it is reported by some programme operators that the increased visibility that comes with a platform can lead to unmanageable volumes of incoming reports (see Section 2.5, many of which are of low or no value [272]. As suggested by operators in Section 2.8, the employment of more complete and robust policies can help ease the burden on the operators, and

may explain the higher prevalence of information pertaining to out-of-scope assets, prohibited actions, and ineligible vulnerabilities in the policies of platform-based programmes.

5.8 Summary of findings

Motivated by the question *What information do organisations convey to hackers through public CVD policy documents?*, we have described the use of deep-learning models to assist in the understanding of CVD programme policy documents in the context of institutional economic theory. In comparison to the earlier work by Laszka et al. [145], analysis in this chapter goes beyond the study of 111 policies from programmes on HackerOne, and presents insights gained from a larger and more diverse dataset of 4,633 policies across 13 bug bounty platforms and from stand-alone programmes. The results help to provide a more comprehensive and nuanced understanding of the variation that exists between programmes and platform alike, and reveals significant variability in policy content and gaps in information where it may be most needed. It is hoped that a better understanding of the institutional elements that commonly form policy documents will enable organisations to better convey the required, or desired, information to hackers that wish to safely participate in the search for vulnerabilities in an organisations assets. A summary of the findings to the subsidiary research questions presented in Section 5.2 are as follows:

1. Within CVD policy documents, hackers are exposed to a litany of generic (e.g. 'applicable laws') and specific (e.g. 'DMCA') references to legal constraints by which they must abide. Understanding the applicable laws and regulations places a considerable burden upon hackers, which could be alleviated by further simplification and standardisation of the legal language used throughout policy documents.

2. Considerable variation exists across programmes within the requirements for rewards, instructions for report submission, and the methods by which reports are assessed. It may be prudent for organisations to emphasise any deviations from, or choose to align with, standard practices within these areas to avoid potential misunderstandings with hackers. Comprehensive policies have previously been linked with greater programme

success [145].

3. Organisations, particularly those operating on a bug bounty platform, typically include information on a programme's scope, rewards, and submission guidelines. However, a majority of programmes fail to include pertinent legal information, potentially introducing hackers to additional risks and liabilities [208]. Furthermore, there is often a lack of specificity as to the restrictions on participants due to their background.

There are a number of limitations to the work presented in this chapter. First, although a wide sample of programme policies were sought for analysis — drawn from stand-alone programmes and those appearing on a wide range of bug bounty platforms — the coverage is incomplete. This will be particularly problematic in the case of stand-alone programmes — for which only 555 policies could be obtained. As such, the results may not be reflective of the wider body of programme policies.

Second, the quality of the annotated data may be limited by the annotator's understanding of the underlying data and their attentiveness to the presence of entities or ability to determine the most appropriate categorical label for a passage of text. It should be recognised that the performance of models trained over data that is incorrectly annotated, or contains incorrect samples, may be degraded [266].

Third, as the analysis is reliant on the performance of the NER and text multi-classification deep-learning models, poor model performance may lead to incomplete or incorrect conclusions being drawn. As shown in Table 5.5 and Table 5.6, the results for both the classification and NER models exhibit significant variability between classes and entities types. Certain types, such as SCOPE-OUT (classification) and PERSON (NER), perform particularly poorly. This is, in part, due to the low number of annotations and the inability of the models to generalise fully across all types. Future research making use of the models should consider the limitations of their performance, and perhaps attempt to improve performance through the annotation of additional samples or architectural refinement (e.g. using a pre-trained transformer).

In the next chapter we conclude the dissertation by affirming the key contributions and findings of the work presented throughout each chapter. Also discussed are the assumptions

that have been relied upon, and the associated limitations that affect the wider applicability of the research outcomes.

Chapter 6

Conclusion

Over the previous four chapters, various aspects concerning the operation of CVD programmes have been explored with the ambition to address the overarching research question presented in Chapter 1: *How can bug bounty programmes support data-driven software development lifecycles?* Within this chapter, the findings and contributions to each chapter are summarised in Section 6.1. The underlying research assumptions, and a discussion of the wider applicability of the findings are presented in Section 6.2. The results presented in each chapter raise further questions that may be suitable for further investigation by the wider research community, and these are outlined in Section 6.3. Finally, concluding remarks are made in Section 6.4.

6.1 Contributions and findings

6.1.1 Survey of the current state of bug bounty and responsible disclosure programmes

Chapter 2 presented insights into the underlying organisational perspectives on the operation of crowdsourced vulnerability discovery programmes, with a particular emphasis on understanding the decisions made by programme operators. Through the use of qualitative surveys and interviews, three key insights are found:

1. In the years since the study by Al-Banna et al. [9], seemingly little has changed in the eyes of programme operators. Of particular note is the continued high-volume stream of low

quality reports faced by organisations, burdening their security teams and diminishing the benefits of the CVD programme.

2. Certain organisations, such as Verizon Media, appear ahead of most when it comes to integrating the results of their CVD programme to institute data-driven change throughout the SDLC. The apparent shallowness of most approaches perhaps suggests a disconnect between CVD programmes and wider organisational security strategies, as organisations predominantly use programmes as a reactive post-deployment measure without greater consideration of the results.

3. For organisations, the key contribution of the chapter is the set of recommendations put forward to aid in the creation of a programme, and to help address common issues that may be faced during operation. It is hoped that these will enable organisations to successfully operate a CVD programme in a cost-effective manner. These are detailed in full in Section 2.7.

6.1.2 Quantification of the costs and benefits to operating a bug bounty programme

Motivated by earlier attempts to quantify the costs and benefits that come with the operation of a bug bounty programme, Chapter 3 provides an updated empirical analysis, focusing on programmes hosted on two major bug bounty platforms. Analysis of the empirical data yields the following findings:

1. The average cost of running a bug bounty programme, when considering the bounty payouts awarded to hackers, is comparable to the cost of hiring two additional software engineers. Although a fairly abstract measure, it is hoped that organisations can use these figures to weigh-up the benefits of both investments to determine, along with other factors, the feasibility of programme operation.

2. The diverse technical backgrounds of hackers enables them to identify vulnerabilities across a broad range of asset classes, and disclose reports across all severity levels.

3. Although the launch of a programme may be followed by a period of significant activity by hackers, most organisations see the volume of reports decline to a more constant level as the 'low-hanging' vulnerabilities are found.

4. The wider incentives that come with continued participation on a bug bounty platform (e.g., invitation to private programmes) may benefit programmes that are unable, or unwilling, to offer significant bounty payouts in exchange for vulnerability information.

6.1.3 Development of models to aid understanding of platform dynamics

Chapter 4 presents the results stemming from novel models focused on capturing the behaviour of hackers on a bug bounty platform, and gives understanding to the dynamics of a particular platform. In doing so, three insights are provided:

1. In contrary to the advertised number of hackers said to be based on a bug bounty platform, it is found that only a small proportion have ever been successful in the search for vulnerabilities. Furthermore, many of the previously successful hackers are no longer active on a given platform, raising questions about the efficacy of the incentives offered by platforms and programmes. On Bugcrowd, approximately 30% of hackers are thought to be inactive.

2. Following an analysis of the archetypal behaviours exhibited by hackers, many are shown to submit one or more vulnerability reports in a short interval before no longer participating on the platform. The prevalence of this behaviour is undesirable for organisations that value repeat interaction with hackers, and may indicate systemic issues that dissuade continued efforts.

3. As indicated by the previous points, long-term participation is currently not viable for a significant proportion of hackers, with at least 50% of formerly active and successful accounts (with an age greater than 2 years) being deemed inactive.

6.1.4 Development of models to aid understanding of programme policies

Unlike prior studies of CVD programme policies, Chapter 5 presents the analysis of a far wider range of programmes, considering both stand-alone programmes and those hosted on 13 bug bounty platforms. Understanding the information conveyed through policy documents is vital for the hacker to safely and successfully participate in an organisation's programme. With this in mind, the following insights are presented:

1. Across policy documents, the necessity to understand the multitude of specific and generic references to legal constraints places considerable burden upon a hacker. A lack of clarity as to the applicable laws and regulations that a hacker is subject to may contribute to the uneasiness reported by many.

2. Organisations should endeavour to address the lack of standardisation that exists across policies, or emphasise components that deviate from the expected norms. In doing so, it is hoped that potential misunderstandings with hackers can be avoided that may otherwise arise from the nuances of a particular programme not being understood.

3. The key contributions of the chapter are the models that allow for the analysis of policy documents. It is hoped that organisations are able to make use of the models and research findings to not only understand their own policies in greater depth, but also the policies of their peers. Performing the analysis on the documents collected as part of the work presented in the chapter reveals a disturbing lack of legal information in a majority of policies.

6.2 Assumptions and limitations

The assumptions that underpin the results presented in each chapter may impose limitations as to their wider applicability.

Starting with Chapter 2, it is assumed that the sample of organisations is representative of the wider population of organisations operating CVD programmes. However, as noted in Section 2.8, there may exist a bias towards organisations based in the US, UK, and Canada.

Particularly for organisations based in non-English speaking countries, the operating environment may impose additional difficulties not faced by the sampled population. For example, legitimate organisations operating from US-sanctioned countries will be unable to make use of US-based bug bounty platforms, ruling out any benefits provided by major platforms (i.e., HackerOne and Bugcrowd) that may normally be relied upon. A further assumption is that the results are generalisable to organisations of all sizes and industries. Absent in the qualitative study is representation from large organisations. Although these organisations may have the resources available to invest into a CVD programme, their reputations may draw attention from a greater number of hackers, thus exacerbating issues concerning high volumes of reports. As with any qualitative study, cognitive biases may impact the responses provided by respondents and the interpretation of the results by the author. Further biases may arise from the type of individual likely to respond to the request for information of this nature. For example, some employees may be unwilling to risk discussing the details of an organisation's CVD programme without seeking the explicit permission from the relevant internal parties. As such, it is perhaps not surprising that the typical profile of the respondents was that of an individual in a senior position within the organisations interviewed.

The implicit assumption that exists within Chapter 3 is that organisations are able to somewhat accurately able to quantify the productivity of software engineers in the context of security operations. One can also imagine that such a figure would exhibit temporal heterogeneity (the engineer may perform better or worse as their tenure increases) and that any single metric may fail to capture many of the intrinsic qualities desirable in an employee. It is therefore expected that benefits arising from comparison in the statement 'the average cost of operating a programme for a year is now less than the cost of hiring two additional software engineers' be evaluated holistically by organisations.

There are several explicit assumptions that affect the applicability of the results. First, the fees often required to make use of a bug bounty platform are neglected in the analysis of the running costs. This is, in part, due to the secrecy surrounding the costs associated with the use of platform features. Furthermore, certain platforms allow organisations to list a VDP for free, thus bringing some of the benefits of a platform for no additional cost. Second, the

analysis focuses only on the two largest US-based bug bounty platforms. Unfortunately, a lack of monetary data for stand-alone programmes and for programmes based on many other platforms prevents broader results from being presented, perhaps restricting the applicability of the results. Third, the absence of data representing any private programmes constitutes a significant barrier to applicability to non-public programmes.

For Chapter 4, the assumption is made that the behaviour of users with private profiles is similar to that of the public profiles collected in Section 4.3. There is little to suggest why hackers would keep their profiles hidden, aside from the obvious desire for increased privacy. A second assumption is that the results on Bugcrowd will generalise to other platforms. At present, time-series data corresponding to the reporting activity of hackers is seldom available on platforms, thus limiting attempts of wider replication. If data were to become transparent, future research may consider the opportunity to link hacker aliases across platforms, allowing for a more complete picture of individual platform dynamics and enable the inter-platform dynamics to be studied.

In Chapter 5, the ambition for the study, and in an effort to aid future researchers, was to create a repository of programme data sourced from platforms and stand-alone programmes. There are several limitations to the current approach. First, as noted earlier, on bug bounty platforms, data cannot be collected from private programmes. Second, although 13 platforms had public data available, there may be more that were not captured in the data collection phase. Third, while it is assume that the sample of stand-alone programme policies is representative of the population, this may only be true for policies written in English.

At present, there are a number of limitations with the annotation and modelling approaches. First, using only a single annotator prevents the inter-coder agreement metrics from being computed, and may lead to biases in the data. Further annotation efforts by the research community, using the published datasets, may help to resolve this. Second, the NLP models implemented do not use state-of-the-art architectures (e.g., transformers) due to hardware constraints. It is expected that the performance of the models could be improved via the use of a transformer. Owing to the publication of the data, and the use of the spaCy library for all NLP models, this may be easily verified by members of the research community that possess

the prerequisite computer hardware.

6.3 Future research directions

Through the research presented in each chapter, several avenues for future research have been identified. It is hoped that the following suggestions will help further the understanding of topics surrounding the use of CVD programmes, providing benefit to hackers and organisations alike.

6.3.1 CVD programme effectiveness: the hacker's perspective

In Chapter 2, the views of organisations were sought in an attempt to understand the current state of CVD programme operation from the perspective of the operators. The results complement the earlier work by Al-Banna et al. [9] in uncovering the problems that plague operators, and highlight countermeasures that may alleviate pervasive problems.

Comparatively, there has been little work exploring similar themes from the perspective of the hackers. A study by Votipka et al. [252] explored, amongst several topics, the challenges faced by hackers when carrying out security research. Hackers reported the added value that comes with the possibility of open discussion with the operators and developers help resolve misunderstandings. This perhaps highlights the value of clear policies and the need for open communication channels.

It is hoped that continued research efforts may bring to light similar recommendations that can be enacted by organisations to improve upon their interactions with hackers, mutually benefiting both parties.

6.3.2 Pricing information: models and methods for understanding bounties

While the study of markets for information has long been of interest to researchers [24], the peculiarities of the ad-hoc legitimate bug bounty markets dominated by monopsonies, in contrast to the less-transparent grey and black markets, bring with it new challenges in determining the fair market price for vulnerability information.

Using the data collected for the work in Chapter 3 and Chapter 5 (bounty information was collected for the purpose of dataset publication), and building on the wealth of literature exploring the role of vulnerability markets [60, 181, 11, 164, 12], there is a need to better understand the pricing of bounties, in relation to the fair market value, in order to further motivate hackers and dissuade them from turning to less-legitimate markets.

The creation of pricing models helps address the need for information by operators on how to appropriately price bounties (see Section 2.7.1), and quell the fears of organisations that believe hackers are turning to alternative markets in the search for competitive compensation for their efforts.

6.3.3 Exploring the effectiveness of incentivisation

Recent research efforts have proposed methods thought to increase the motivation of hackers through techniques such as 'gamification' [183]. However, little has been done to test the effectiveness of the incentivisation techniques in the context of CVD programmes. From the results of Chapter 4, it is clear that a lack of motivation is prevalent amongst the hacker community, and as described in Section 2.5 it is an issue faced by many organisations.

Furthermore, as part of the data collection effort described in Chapter 5, it is clear that many organisations attempt to incentivise hackers through the use of limited-time events or promotions, often offering bounty bonuses to successful hackers. However, academic research has yet to explore the effectiveness of such techniques.

There exists scope for further research examining the effectiveness of incentivisation techniques in the context of CVD programmes. The results of which may help organisations combat waning motivation, and amplify the benefits received through the operation of a programme.

6.3.4 Platforms and policy: the effect of institutions on the performance of platforms

As noted in Chapter 5, each platform brings with it a number of policies, agreements, and conditions that affect both organisations and hackers alike. Although not part of the scope of the chapter, it is clear that the underlying institutions will shape the 'market' that exists on

a platform, influencing the constituent programmes. For example, if a platform also happens to be the sole payment processor, restrictions may be placed on the background of the parties that can be involved in monetary exchange (as is seen on HackerOne).

A comparative study through the lens of the theory of institutions [177] has the potential to rationalise the differences between the myriad of bug bounty platforms, and the impact this has on the market participants. Furthermore, as discussed by North [177], examining institutional change through a longitudinal study of constraints enables insights concerning the changing dynamics between operators, platforms, and hackers to be uncovered. It is hoped that operators and hackers will benefit from the results, enabling them to choose the platform that aligns best with their risk tolerances.

6.4 Concluding remarks

To conclude the dissertation, we briefly re-examine the overarching research question *How can bug bounty programmes support data-driven software development lifecycles?*

Through the four contributions, we have addressed pertinent issues affecting the operation of bug bounty programmes, with the ambition to provide actionable data-driven insights that enable organisations to better manage a programme in order to support their secure development processes. Furthermore, we have identified the tangible benefits to security that come with the operation of a CVD programme. It is hoped that the work presented in this dissertation inspires organisations to consider employing the security activity, and take onboard the recommendations put forth to manage a programme effectively.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] Accenture. Cost of cybercrime study. `https://www.accenture.com/us-en/insights/security/cost-cybercrime-study`. Last accessed: May 28, 2019.

[3] ACM. Code of ethics and professional conduct. `https://www.acm.org/code-of-ethics`. Last accessed: February 03, 2022.

[4] Edward N Adams. Optimizing preventive service of software products. *IBM Journal of Research and Development*, 28(1):2–14, 1984.

[5] Ali Ahmed and Brian Lee. Organizational learning on bug bounty platforms. In *Proceedings. 2020 26th Americas Conference on Information Systems, AMCIS 2020*, page 33. AIS, 2020.

[6] Sanjay P Ahuja, Sindhu Mani, and Jesus Zambrano. A survey of the state of cloud computing in healthcare. *Network and Communication Technologies*, 1(2):12, 2012.

[7] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Daniel Votipka, and Aron Laszka. The hackers' viewpoint: Exploring challenges and benefits of bug-bounty programs. In *6th Workshop on Security Information Workers*, pages 1–7, 2020.

[8] Mortada Al-Banna, Boualem Benatallah, and Moshe Chai Barukh. Software security professionals: Expertise indicators. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 139–148. IEEE, 2016.

[9] Mortada Al-Banna, Boualem Benatallah, Daniel Schlagwein, Elisa Bertino, and Moshe Chai Barukh. Friendly hackers to the rescue: How organizations perceive crowdsourced vulnerability discovery. In *Pacific Asia Conference on Information Systems (PACIS)*, page 230, 2018.

[10] Saeed S Alahmari, Dmitry B Goldgof, Peter R Mouton, and Lawrence O Hall. Challenges for the repeatability of deep learning models. *IEEE Access*, 8:211860–211868, 2020.

[11] Abdullah Algarni and Yashwant Malaiya. Software vulnerability markets: Discoverers and buyers. *International Journal of Computer, Information Science and Engineering*, 8(3):71–81, 2014.

[12] Abdullah M Algarni. The historical relationship between the software vulnerability lifecycle and vulnerability markets: Security and economic risks. *Computers*, 11(9):137, 2022.

[13] Abdullah M Algarni and Yashwant K Malaiya. Most successful vulnerability discoverers: Motivation and methods. In *Proceedings of the International Conference on Security and Management (SAM)*, page 1. The Steering Committee of The World Congress in Computer Science, 2013.

[14] Luca Allodi. Economic factors of vulnerability trade and exploitation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1483–1499. ACM, 2017.

[15] Noura Alomar, Primal Wijesekera, Edward Qiu, and Serge Egelman. "you've got your nice list of bugs, now what?" vulnerability discovery and management processes in the wild. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 319–339, 2020.

[16] Alphabet. Google Play Security Reward Program (GPSRP). `https://bughunters.google.com/about/rules/5604090422493184`. Last accessed: February 03, 2022.

[17] Ross Anderson. Security in open versus closed systems—the dance of Boltzmann, Coase and Moore. Technical report, University of Cambridge, Computer Laboratory, 2002.

[18] Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Michel JG Van Eeten, Michael Levi, Tyler Moore, and Stefan Savage. Measuring the cost of cybercrime. In *The Economics of Information Security and Privacy*, pages 265–300. Springer, 2013.

[19] Ross Anderson, Tyler Moore, Shishir Nagaraja, and Andy Ozment. Incentives and information security. *Algorithmic Game Theory*, pages 633–649, 2007.

[20] Shanai Ardi, David Byers, Per Hakon Meland, Inger Anne Tondel, and Nahid Shahmehri. How can the developer benefit from security modeling? In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 1017–1025. IEEE, 2007.

[21] Jart Armin, Bryn Thompson, and Piotr Kijewski. Cybercrime economic costs: No measure no solution. In *Combatting Cybercrime and Cyberterrorism*, pages 135–155. Springer, 2016.

[22] Ashish Arora, Jonathan P Caulkins, and Rahul Telang. Research note—sell first, fix later: Impact of patching on software quality. *Management Science*, 52(3):465–471, 2006.

[23] Ashish Arora, Ramayya Krishnan, Anand Nandkumar, Rahul Telang, and Yubao Yang. Impact of vulnerability disclosure and patch availability - an empirical analysis. In *Third Workshop on the Economics of Information Security*, pages 1268–1287, 2004.

[24] Kenneth Arrow. Economic welfare and the allocation of resources for invention. In *The rate and direction of inventive activity: Economic and social factors*, pages 609–626. Princeton University Press, 1962.

[25] Tej D Azad, Jeff Ehresman, Ali Karim Ahmed, Victor E Staartjes, Daniel Lubelski, Martin N Stienen, Anand Veeravagu, and John K Ratliff. Fostering reproducibility and generalizability in machine learning for clinical prediction modeling in spine surgery. *The Spine Journal*, 21(10):1610–1616, 2021.

[26] Sara Sun Beale and Peter Berris. Hacking the internet of things: Vulnerabilities, dangers, and legal responses. *Duke Law & Technology Review*, 16:161, 2017.

[27] Kristian Beckers, M Heisel, and D Hatebur. *Pattern and Security Requirements, Engineering-Based Establishment of Security Standards*. Springer, 2015.

[28] RE Bell. The prosecution of computer crime. *Journal of Financial Crime*, 2002.

[29] Andreea Bendovschi. Cyber-attacks – trends, patterns and security countermeasures. *Procedia Economics and Finance*, 28:24–31, 2015.

[30] Igor Bernik. Cybercrime: The cost of investments into protection. *Varstvoslovje: Journal of Criminal Justice & Security*, 16(2), 2014.

[31] Marc Blatter, Samuel Muehlemann, and Samuel Schenker. The costs of hiring skilled workers. *European Economic Review*, 56(1):20–35, 2012.

[32] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.

[33] Rainer Böhme. Vulnerability markets. *Proc. of 22C3*, 27:30, 2005.

[34] Rainer Böhme. A comparison of market approaches to software vulnerability disclosure. In *Emerging Trends in Information and Communication Security*, pages 298–311. Springer, 2006.

[35] Scott R Boss, Laurie J Kirsch, Ingo Angermeier, Raymond A Shingler, and R Wayne Boss. If someone is watching, i'll do what i'm asked: mandatoriness, control, and information security. *European Journal of Information Systems*, 18(2):151–164, 2009.

[36] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. Characterbert: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. *arXiv preprint arXiv:2010.10392*, 2020.

[37] Robert M Brady, Ross J Anderson, and Robin C Ball. Murphy's law, the fitness of evolving species, and the limits of software reliability. Technical report, University of Cambridge, Computer Laboratory, 1999.

[38] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

[39] Alan Bryman. *Social research methods*. Oxford University Press, 2016.

[40] Elizabeth A Buchanan and Erin E Hvizdak. Online survey tools: Ethical and methodological concerns of human research ethics committees. *Journal of Empirical Research on Human Research Ethics*, 4(2):37–48, 2009.

[41] Bugcrowd. Bug bounty platform. `https://www.bugcrowd.com`. Last accessed: January 14, 2019.

[42] Bugcrowd. Integrating crowdsourced security with the SDLC. `https://www.bugcrowd.com/blog/integrating-crowdsourced-security-with-the-software-development-lifecycle/`. Last accessed: January 15, 2019.

[43] Bugcrowd. Managed bug bounty. `https://www.bugcrowd.com/products/bug-bounty/`. Last accessed: February 12, 2022.

[44] Bugcrowd. Public bug bounty program list. `https://www.bugcrowd.com/bug-bounty-list/`. Last accessed: February 03, 2022.

[45] Bugcrowd. The state of crowdsourced security, Priority One report 2019 edition. https://www.bugcrowd.com/resources/reports/priority-one-report/. Last accessed: March 21, 2020.

[46] Bugcrowd. University. https://www.bugcrowd.com/hackers/bugcrowd-university/. Last accessed: May 10, 2021.

[47] Frankie E Catota, M Granger Morgan, and Douglas C Sicker. Cybersecurity education in a developing nation: the Ecuadorian environment. *Journal of Cybersecurity*, 5(1):tyz001, 2019.

[48] Edoardo Celeste and Federico Fabbrini. Competing jurisdictions: Data privacy across the borders. *Data Privacy and Trust in Cloud Computing*, pages 43–58, 2020.

[49] Kelly Cesare. Prosecuting computer virus authors: The need for an adequate and immediate international solution. *Transnational Law.*, 14:135, 2001.

[50] Joseph L Christian. *Bug Bounty Programs: Analyzing the Future of Vulnerability Research*. PhD thesis, Utica College, 2018.

[51] Cobalt. Bug bounty platform. https://cobalt.io/. Last accessed: January 14, 2019.

[52] Cobalt. Core security pentesters. https://app.cobalt.io/pentesters. Last accessed: March 21, 2020.

[53] Cobalt. Customer list. https://cobalt.io/customers. Last accessed: February 03, 2022.

[54] Wm Arthur Conklin, Raymond E Cline, and Tiffany Roosa. Re-engineering cybersecurity education in the US: an analysis of the critical factors. In *2014 47th Hawaii International Conference on System Sciences*, pages 2006–2014. IEEE, 2014.

[55] William Crumpler and James A Lewis. *The cybersecurity workforce gap*. Center for Strategic and International Studies (CSIS) Washington, DC, USA, 2019.

[56] Crunchbase. Bugcrowd financials. https://www.crunchbase.com/organization/bugcrowd/company_financials. Last accessed: February 10, 2021.

[57] Crunchbase. Hackerone company financials. https://www.crunchbase.com/organization/hackerone/company_financials. Last accessed: May 10, 2021.

[58] Crunchbase. Hackerone financials. https://www.crunchbase.com/organization/hackerone/company_financials. Last accessed: February 10, 2021.

[59] Katja C DeGroot. An overview of recent changes in California computer crime laws: The criminalization of computer contamination and strengthened penalty provisions, California Penal Code sections, 502, 502.01, 1203.047, 1203.048, 1990.

[60] Matthias Dellago, Daniel W Woods, and Andrew C Simpson. Characterising 0-day exploit brokers. In *The 21st Workshop on the Economics of Information Security*, 2022.

[61] Shane P Desselle. Construction, implementation, and analysis of summated rating attitude scales. *American Journal of Pharmaceutical Education*, 69(5), 2005.

[62] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[63] Disclose. Research threats: Legal threats against security researchers. https://github.com/disclose/research-threats. Last accessed: February 12, 2022.

[64] Disclose. Safe harbour. https://github.com/disclose/policymaker/blob/main/templates/disclose-io-safe-harbor/en-US.md. Last accessed: February 12, 2022.

[65] Disclose. Simple safe harbour. https://github.com/disclose/policymaker/blob/main/templates/disclose-io-simple-safe-harbor/en-US.md. Last accessed: February 12, 2022.

[66] Disclose.io. Program database. https://disclose.io/programs/. Last accessed: February 03, 2022.

[67] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.

[68] Serge Egelman, Cormac Herley, and Paul C Van Oorschot. Markets for zero-day exploits: Ethics and implications. In *Proceedings of the 2013 New Security Paradigms Workshop*, pages 41–46. ACM, 2013.

[69] Amit Elazari. Hacking the law: Are bug bounties a true safe harbor? In *Enigma*, 2018.

[70] Ryan Ellis and Yuan Stevens. Bounty everything: Hackers and the making of the global bug marketplace. Available at SSRN 4009275, 2022. SSRN.

[71] Pardis Emami-Naeini, Yuvraj Agarwal, Lorrie Faith Cranor, and Hanan Hibshi. Ask the experts: What should be on an IoT privacy and security label? In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 447–464. IEEE, 2020.

[72] Daniel Etcovitch and Thyla van der Merwe. Coming in from the cold: A safe harbor from the CFAA and the DMCA §1201 for security researchers (June 1, 2018). Berkman Klein Center Research Publication No. 2018-4. SSRN.

[73] Facebook. Whitehat program. https://www.facebook.com/whitehat. Last accessed: January 14, 2019.

[74] Yuanrui Fan, Xin Xia, David Lo, and Ahmed E Hassan. Chaff from the wheat: Characterizing and determining valid bug reports. *IEEE Transactions on Software Engineering*, 46(5):495–525, 2018.

[75] Matthew Finifter, Devdatta Akhawe, and David Wagner. An empirical study of vulnerability rewards programs. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 273–288, 2013.

[76] Arlene Fink. *The survey handbook*. SAGE Publications, Inc., 2003.

[77] FireBounty. The right path to coordinated vulnerability disclosure. `https://firebounty.com`. Last accessed: February 03, 2022.

[78] First. User guide to cvss 3.0. `https://www.first.org/cvss/user-guide`. Last accessed: March 19, 2019.

[79] Peter Fisk. De-bugging legal issues in bug bounty programs and VDP. In *RSA Conference 2019*, 2019.

[80] Rudolph Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221, 1948.

[81] Luca Follis and Adam Fish. State hacking at the edge of code, capitalism and culture. *Information, Communication & Society*, 25(2):242–257, 2022.

[82] Park Foreman. *Vulnerability management*. Auerbach Publications, 2009.

[83] Paul Formosa, Michael Wilson, and Deborah Richards. A principlist framework for cybersecurity ethics. *Computers & Security*, 109:102382, 2021.

[84] Panagiotis Fotaris, Theodoros Mastoras, Richard Leinfellner, and Yasmine Rosunally. Climbing up the leaderboard: An empirical study of applying gamification techniques to a computer programming class. *Electronic Journal of e-learning*, 14(2):94–110, 2016.

[85] Edward H Freeman. Vulnerability disclosure: The strange case of Bret McDanel. *Information Systems Security*, 16(2):127–131, 2007.

[86] Huw Fryer and Elena Simperl. Web science challenges in researching bug bounties. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 273–277. ACM, 2017.

[87] Steven Furnell, Pete Fischer, and Amanda Finch. Can't get the staff? the growing need for cyber-security skills. *Computer Fraud & Security*, 2017(2):5–10, 2017.

[88] Mirta Galesic and Michael Bosnjak. Effects of questionnaire length on participation and indicators of response quality in a web survey. *Public Opinion Quarterly*, 73(2):349–360, 2009.

[89] Darko Galinec and William Steingartner. Combining cybersecurity and cyber defense to achieve cyber resilience. In *2017 IEEE 14th International Scientific Conference on Informatics*, pages 87–93. IEEE, 2017.

[90] Josh Gardner, Yuming Yang, Ryan Baker, and Christopher Brooks. Enabling end-to-end machine learning replicability: A case study in educational data mining. *arXiv preprint arXiv:1806.05208*, 2018.

[91] Glassdoor. Software engineer salaries in London. `https://www.glassdoor.co.uk/Salaries/london-software-engineer-salary.htm`. Last accessed: January 29, 2019.

[92] Eric Goldman. An introduction to the California Consumer Privacy Act (CCPA). *Santa Clara University Legal Studies Research Paper*, 2020.

[93] Google. Trends: Search popularity of bug bounty platforms. `https://trends.google.com/trends/explore?geo=US&q\=%2Fg%2F11bwh9x27y,Bugcrowd,safehats,Intigriti,Synack`. Last accessed: March 19, 2019.

[94] Google. Vulnerability reward program. `https://www.google.com/about/appsecurity/reward-program/index.html`. Last accessed: January 14, 2019.

[95] Audrey Guinchard. The Computer Misuse Act 1990 to support vulnerability research? proposal for a defence for hacking as a strategy in the fight against cybercrime, 2017. SSRN.

[96] Odd Erik Gundersen and Sigbjørn Kjensmo. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[97] HackerOne. 188 fascinating facts. `https://www.hackerone.com/blog/118-Fascinating-Facts-HackerOnes-Hacker-Powered-Security-Report-2018`. Last accessed: February 25, 2019.

[98] HackerOne. 2018 hacker report. `https://www.hackerone.com/sites/default/files/2018-01/2018-Hacker-Report`. Last accessed: February 25, 2019.

[99] HackerOne. The 2019 hacker report. `https://www.hackerone.com/sites/default/files/2019-02/the-2019-hacker-report_3.pdf`. Last accessed: September 23, 2019.

[100] HackerOne. The 2020 hacker report. `https://www.hackerone.com/resources/reporting/the-2020-hacker-report`. Last accessed: February 10, 2021.

[101] HackerOne. Code of conduct. `https://www.hackerone.com/policies/code-of-conduct`. Last accessed: March 4, 2022.

[102] HackerOne. Hacker101. `https://www.hackerone.com/for-hackers/hacker-101`. Last accessed: May 10, 2021.

[103] HackerOne. Hackerone bug bounty platform. `https://www.hackerone.com/`. Last accessed: January 14, 2019.

[104] HackerOne. Product offerings. `https://docs.hackerone.com/programs/product-offerings.html`. Last accessed: February 12, 2022.

[105] HackerOne. Program directory. `https://hackerone.com/directory/programs`. Last accessed: February 03, 2022.

[106] HackerOne. Response target metrics. `https://docs.hackerone.com/programs/response-target-metrics.html`. Last accessed: February 25, 2019.

[107] Munawar Hafiz and Ming Fang. Game of detections: how are security vulnerabilities discovered in the wild? *Empirical Software Engineering*, 21(5):1920–1959, 2016.

[108] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of 'big data' on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[109] Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using Hidden Markov Model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 192–196. IEEE, 2005.

[110] Hideaki Hata, Mingyu Guo, and M Ali Babar. Understanding the heterogeneity of contributors in bug bounty programs. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 223–228. IEEE Press, 2017.

[111] Benjamin J Heil, Michael M Hoffman, Florian Markowetz, Su-In Lee, Casey S Greene, and Stephanie C Hicks. Reproducibility standards for machine learning in the life sciences. *Nature Methods*, 18(10):1132–1135, 2021.

[112] Chad Heitzenrater, Rainer Böhme, and Andrew Simpson. The days before zero day: Investment models for secure software engineering. In *Proceedings of the 15th Workshop on the Economics of Information Security (WEIS 2016)*, 2016.

[113] Chad Heitzenrater and Andrew Simpson. A case for the economics of secure software development. In *Proceedings of the 2016 New Security Paradigms Workshop*, pages 92–105, 2016.

[114] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[115] Michael Hilton, Nicholas Nelson, Timothy Tunnell, Darko Marinov, and Danny Dig. Trade-offs in continuous integration: assurance, security, and flexibility. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 197–207, 2017.

[116] XA Hoang and Jiankun Hu. An efficient Hidden Markov Model training scheme for anomaly intrusion detection of server applications based on system calls. In *Proceedings. 2004 12th IEEE International Conference on Networks (ICON 2004)(IEEE Cat. No. 04EX955)*, pages 470–474. IEEE, 2004.

[117] Geoffrey M Hodgson. What are institutions? *Journal of Economic Issues*, 40(1):1–25, 2006.

[118] Hossein Homaei and Hamid Reza Shahriari. Seven years of software vulnerabilities: The ebb and flow. *IEEE Security & Privacy*, 15(1):58–65, 2017.

[119] Allen D Householder, Garret Wassermann, Art Manion, and Chris King. The CERT guide to Coordinated Vulnerability Disclosure. Technical report, Carnegie-Mellon University, 2017.

[120] IBM. IBM survey: Consumers think cybercrime now three times more likely than physical crime. `https://www-03.ibm.com/press/us/en/pressrelease/19154.wss`. Last accessed: March 19, 2020.

[121] iNTiGRiTi. Public bug bounty programs. `https://www.intigriti.com/programs`. Last accessed: February 03, 2022.

[122] L Jean Camp and Catherine Wolfram. Pricing security: A market in vulnerabilities. *Economics of information security*, pages 17–34, 2004.

[123] Russell L Jones and Abhinav Rastogi. Secure coding: building security into the software development life cycle. *Information Systems Security*, 13(5):29–39, 2004.

[124] Dheeraj Joshi. Awesome bug bounty. `https://github.com/djadmin/awesome-bug-bounty`. Last accessed: February 03, 2022.

[125] Shaivesh Kamra and James Scott. Impact of data breaches to organizations and individuals. SSRN.

[126] Moon Young Kang. Sustainable profit versus unsustainable growth: Are venture capital investments and governmental support medicines or poisons? *Sustainability*, 12(18):7773, 2020.

[127] Amanpreet Kapoor, Andrew Penton, and Hamish Pierpont. Eliciting course feedback through a bug bounty program. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 2*, pages 595–596, 2022.

[128] Michael L Katz and Carl Shapiro. Network externalities, competition, and compatibility. *The American Economic Review*, 75(3):424–440, 1985.

[129] SADTP Kaushalya, RMRSB Randeniya, and ADS Liyanage. An overview of social engineering in the context of information security. In *IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–6. IEEE, 2018.

[130] Fred Nichols Kerlinger. *Foundations of behavioral research*. Holt, Rinehart and Winston: New York, 1966.

[131] Jay P Kesan and Carol M Hayes. Bugs in the market: Creating a legitimate, transparent, and vendor-focused market for software vulnerabilities. *Arizona Law Review*, 58:753, 2016.

[132] Wahab Khan, Ali Daud, Jamal A Nasir, and Tehmina Amjad. A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait Journal of Science*, 43(4), 2016.

[133] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[134] Justin Kitzes, Daniel Turek, and Fatma Deniz. *The practice of reproducible research: case studies and lessons from the data-intensive sciences.* University of California Press, 2018.

[135] Tetsunori Kobayashi and Satoshi Haruyama. Partly-Hidden Markov Model and its application to gesture recognition. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3081–3084. IEEE, 1997.

[136] Eduard Kovacs. Securityweek: Intel's record bounty. `https://www.securityweek.com/intel-pays-100000-bounty-new-spectre-variants`. Last accessed: January 14, 2019.

[137] Marleen Weulen Kranenbarg, Thomas J Holt, and Jeroen van der Ham. Don't shoot the messenger! a criminological and computer science perspective on coordinated vulnerability disclosure. *Crime Science*, 7(1):16, 2018.

[138] Markus Krause and René Kizilcec. To play or not to play: Interactions between response quality and task complexity in games and paid crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[139] Jon A Krosnick. Questionnaire design. In *The Palgrave handbook of survey research*, pages 439–455. Springer, 2018.

[140] Andreas Kuehn and Milton Mueller. Analyzing bug bounty programs: An institutional perspective on the economics of software vulnerabilities. In *2014 TPRC Conference Paper*, 2014.

[141] Anand Kulkarni, Matthew Can, and Björn Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1003–1012. ACM, 2012.

[142] Leonard Kwan, Pradeep Ray, and Greg Stephens. Towards a methodology for profiling cyber criminals. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 264–264. IEEE, 2008.

[143] Noam Lapidot-Lefler and Azy Barak. Effects of anonymity, invisibility, and lack of eye-contact on toxic online disinhibition. *Computers in Human Behavior*, 28(2):434–443, 2012.

[144] Aron Laszka, Mingyi Zhao, and Jens Grossklags. Banishing misaligned incentives for validating reports in bug-bounty platforms. In *European Symposium on Research in Computer Security*, pages 161–178. Springer, 2016.

[145] Aron Laszka, Mingyi Zhao, Akash Malbari, and Jens Grossklags. The rules of engagement for bug bounty programs. In *International Conference on Financial Cryptography and Data Security*, pages 138–159, 2018.

[146] Thomas D LaToza and Andre van der Hoek. Crowdsourcing in software engineering: Models, motivations, and challenges. *IEEE software*, 33(1):74–80, 2016.

[147] Cal. Legis. California Consumer Privacy Act. *California Civil Code, Section 1798.100*, 2018.

[148] John Leyden. The Register: Fatigue fears over bug bounty programs. `https://www.theregister.com/2016/11/09/bug_bounty_fatigue_fears/`. Last accessed: August 27, 2020.

[149] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[150] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. Dice loss for data-imbalanced NLP tasks. *arXiv preprint arXiv:1911.02855*, 2019.

[151] Yuni Li and Ling Zhao. Collaborating with bounty hunters: How to encourage white hat hackers' participation in vulnerability crowdsourcing programs through formal and relational governance. *Information & Management*, 59(4):103648, 2022.

[152] Petra Lietz. Research into questionnaire design: A summary of the literature. *International Journal of Market Research*, 52(2):249–272, 2010.

[153] Chao Liu, Cuiyun Gao, Xin Xia, David Lo, John Grundy, and Xiaohu Yang. On the replicability and reproducibility of deep learning in software engineering. *arXiv preprint arXiv:2006.14244*, 2020.

[154] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[155] Wei Luo, Dinh Phung, Truyen Tran, Sunil Gupta, Santu Rana, Chandan Karmakar, Alistair Shilton, John Yearwood, Nevenka Dimitrova, Tu Bao Ho, et al. Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view. *Journal of Medical Internet Research*, 18(12):e5870, 2016.

[156] Gordon Lyon. Full disclosure mailing list. `https://seclists.org/fulldisclosure/`. Last accessed: January 27, 2022.

[157] Thomas Maillart, Mingyi Zhao, Jens Grossklags, and John Chuang. Given enough eyeballs, all bugs are shallow? revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity*, 3(2):81–90, 2017.

[158] Suresh S Malladi and Hemang C Subramanian. Bug bounty programs for cybersecurity: Practices, issues, and recommendations. *IEEE Software*, 37(1):31–39, 2019.

[159] Katja Lozar Manfreda, Michael Bosnjak, Jernej Berzelak, Iris Haas, and Vasja Vehovar. Web surveys versus other survey modes: A meta-analysis comparing response rates. *International Journal of Market Research*, 50(1):79–104, 2008.

[160] Mary Manjikian. *Cybersecurity ethics: an introduction*. Routledge, 2017.

[161] Dhendra Marutho, Sunarna Hendra Handaka, Ekaprana Wijaya, et al. The determination of cluster number at K-mean using elbow method and purity evaluation on headline news. In *2018 International Seminar on Application for Technology of Information and Communication*, pages 533–538. IEEE, 2018.

[162] Gary McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, 2004.

[163] Gary McGraw and Brian Chess. The Building Security in Maturity Model (BSIMM). *Online*, 2009.

[164] David McKinney. Vulnerability bazaar. *IEEE Security & Privacy*, 5(6):69–73, 2007.

[165] Joss Meakins. A zero-sum game: the zero-day market in 2018. *Journal of Cyber Policy*, 4(1):60–71, 2019.

[166] Verizon Media. Brighttalk webinar: Breaking barriers: Introducing the bug bounty lifecycle. `https://core.brighttalk.com/webcast/13109/458461`. Last accessed: January 27, 2022.

[167] Alessio Miaschi and Felice Dell'Orletta. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119. Association for Computational Linguistics, 2020.

[168] Microsoft. Bug bounty programs. `https://www.microsoft.com/en-us/msrc/bounty`. Last accessed: January 14, 2019.

[169] Microsoft. Security development lifecycle. https://www.microsoft.com/en-us/securityengineering/sdl/about. Last accessed: May 27, 2022.

[170] Charles Miller. The legitimate vulnerability market: the secretive world of 0-day exploit sales. In *The sixth Workshop on the Economics of Information Security*, 2007.

[171] Jon L Mills and Kelsey Harclerode. Privacy, mass intrusion, and the modern data breach. *Fla. L. Rev.*, 69:771, 2017.

[172] Mariella Moon. Engadget: Apple's expanded bug bounty program opens to everyone. https://www.engadget.com/2019/12/20/apple-expanded-bug-bounty-program-live/. Last accessed: March 19, 2020.

[173] Ernest Mougoue. Synopsys: Secure SDLC 101. https://www.synopsys.com/blogs/software-security/secure-sdlc/. Last accessed: July 21, 2019.

[174] Nuthan Munaiah and Andrew Meneely. Vulnerability severity scoring and bounties: Why the disconnect? In *Proceedings of the 2nd International Workshop on Software Analytics*, pages 8–14. ACM, 2016.

[175] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[176] David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 266–277. Springer, 2006.

[177] Douglass C North. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, 1990.

[178] Kenneth Olmstead and Aaron Smith. Americans and cybersecurity. *Pew Research Center*, 26:311–327, 2017.

[179] Helena Holmström Olsson and Jan Bosch. Towards data-driven product development: A multiple case study on post-deployment data usage in software-intensive embedded systems. In *International Conference on Lean Enterprise Software and Systems*, pages 152–164. Springer, 2013.

[180] Graziella Orrù, Merylin Monaro, Ciro Conversano, Angelo Gemignani, and Giuseppe Sartori. Machine learning in psychometrics and psychological research. *Frontiers in Psychology*, 10:2970, 2020.

[181] Andy Ozment. Bug auctions: Vulnerability markets reconsidered. In *Third workshop on the economics of information security*, pages 19–26, 2004.

[182] Andy Ozment and Stuart E Schechter. Milk or wine: does software security improve with age? In *USENIX Security Symposium*, volume 6, pages 10–5555, 2006.

[183] Jamie O'Hare and Lynsay A Shepherd. Developing a gamified peer-reviewed bug bounty programme. In *International Conference on Human-Computer Interaction*, pages 514–522. Springer, 2022.

[184] Pierluigi Paganini. Security affairs: Wooyun group arrested. `https://securityaffairs.co/wordpress/49916/hacking/wooyun-group-arrested.html`. Last accessed: January 18, 2019.

[185] Rajesh Pandey. iPhoneHacks: Apple's bug bounty program fails to take off. `http://www.iphonehacks.com/2017/07/apples-bug-bounty-program-fails-take-off-ios-bugs-valuable-disclose.html`. Last accessed: March 19, 2020.

[186] Stuart L Pardau. The California Consumer Privacy Act: Towards a European-style privacy regime in the United States. *Journal of Technology Law & Policy*, 23:68, 2018.

[187] Parliment. Computer Misuse Act 1990 (c. 18), 1990.

[188] Chintan Parmar, Joseph D Barry, Ahmed Hosny, John Quackenbush, and Hugo JWL Aerts. Data analysis strategies in medical imagingdata science designs in medical imaging. *Clinical Cancer Research*, 24(15):3492–3499, 2018.

[189] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022.

[190] Michael Peneder. The impact of venture capital on innovation behaviour and firm growth. *Venture Capital*, 12(2):83–107, 2010.

[191] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program. *Journal of Machine Learning Research*, 22, 2021.

[192] Tara Poteat and Frank Li. Who you gonna call? an empirical evaluation of website security.txt deployment. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 526–532. ACM, 2021.

[193] Roger S Pressman. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

[194] Daniel Ramage. Hidden Markov Models, fundamentals. *CS229 Section Notes*, 1, 2007.

[195] Eric Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

[196] Przemysław Roguski. An inspection regime for cyber weapons: A challenge too far? *American Journal of International Law*, 115:111–115, 2021.

[197] Margaret Rouse. Search security: Whitehat hacker definition. `https://searchsecurity.techtarget.com/definition/white-hat`. Last accessed: January 14, 2019.

[198] Brent R Rowe and Michael P Gallaher. Private sector cyber security investment strategies: An empirical analysis. In *The fifth Workshop on the Economics of Information Security (WEIS06)*, 2006.

[199] Jukka Ruohonen and Luca Allodi. A bug bounty perspective on the disclosure of web vulnerabilities. *arXiv preprint arXiv:1805.09850*, 2018.

[200] Jukka Ruohonen and Kai K Kimppa. Updating the Wassenaar debate once again: Surveillance, intrusion software, and ambiguity. *Journal of Information Technology & Politics*, 16(2):169–186, 2019.

[201] Kavita Sahu, R Shree, and R Kumar. Risk management perspective in SDLC. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(3), 2014.

[202] Maria Elena Sanchez. Effects of questionnaire design on the quality of survey data. *Public Opinion Quarterly*, 56(2):206–217, 1992.

[203] Arthur D Santana. Virtuous or vitriolic: The effect of anonymity on civility in online newspaper reader comment boards. *Journalism Practice*, 8(1):18–33, 2014.

[204] Natalie Sappleton and Fernando Lourenço. Email subject lines and response rates to invitations to participate in a web survey and a face-to-face interview: the sound of silence. *International Journal of Social Research Methodology*, 19(5):611–622, 2016.

[205] Stuart Schechter. How to buy better testing using competition to get the most security and robustness for your dollar. In *Infrastructure Security: International Conference, InfraSec 2002 Bristol, UK, October 1–3, 2002 Proceedings*, pages 73–87. Springer, 2002.

[206] Stuart Edward Schechter. *Computer security strength and risk: a quantitative approach*. Harvard University, 2004.

[207] Howard Schuman and Stanley Presser. The open and closed question. *American Sociological Review*, pages 692–712, 1979.

[208] Amy Terry Sheehan. Proactive steps to prevent legal pitfalls in bug bounty programs, 2017.

[209] Mario Silic and Paul Benjamin Lowry. Breaking bad in cyberspace: Understanding why and how black hat hackers manage their nerves to commit their virtual crimes. *Information Systems Frontiers*, 23(2):329–341, 2021.

[210] Justin Smith, Christopher Theisen, and Titus Barik. A case study of software security red teams at Microsoft. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–10. IEEE, 2020.

[211] Soren Sonnenburg, Mikio L Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCunn, Klaus-Robert Muller, Fernando Pereira, Carl Edward Rasmussen, et al. The need for open source software in machine learning, 2007.

[212] Anna Sperotto, Ramin Sadre, Pieter-Tjerk de Boer, and Aiko Pras. Hidden Markov Model modeling of SSH brute-force attacks. In *International Workshop on Distributed Systems: Operations and Management*, pages 164–176. Springer, 2009.

[213] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras.* Packt Publishing Ltd, 2018.

[214] Paul N Stockton and Michele Golabek-Goldman. Curbing the market for cyber weapons. *Yale L. & Pol'y Rev.*, 32:239, 2013.

[215] John Suler. The online disinhibition effect. *Cyberpsychology & Behavior*, 7(3):321–326, 2004.

[216] Gail M Sullivan and Anthony R Artino Jr. Analyzing and interpreting data from Likert-type scales. *Journal of Graduate Medical Education*, 5(4):541–542, 2013.

[217] Michael Sutton and Frank Nagle. Emerging economic models for vulnerability research. In *The fifth Workshop on the Economics of Information Security*, 2006.

[218] Dan Swinhoe. CSO Online: The biggest data breach fines, penalties and settlements so far. `https://www.csoonline.com/article/3410278/the-biggest-data-breach-fines-penalties-and-settlements-so-far.html`. Last accessed: March 19, 2020.

[219] Synack. Bug bounty platform. https://www.synack.com/solution/. Last accessed: March 19, 2019.

[220] Synopsys. BSIMM12 2021 Foundations report. https://www.bsimm.com/content/dam/bsimm/reports/bsimm12-foundations.pdf. Last accessed: March 4, 2022.

[221] Synopsys. BSIMM12: Building Security in Maturity Model version 12. https://www.bsimm.com/download/. Last accessed: March 4, 2022.

[222] Synopsys. BSIMM6: Building Security in Maturity Model version 6. https://www.inf.ed.ac.uk/teaching/courses/sp/2015/lecs/BSIMM6.pdf. Last accessed: March 4, 2022.

[223] Synopsys. BSIMM9: Building Security in Maturity Model version 9. https://www.bsimm.com/download/. Last accessed: December 18, 2019.

[224] Leonie Maria Tanczer. 50 shades of hacking: How IT and cybersecurity industry actors perceive good, bad, and former hackers. *Contemporary Security Policy*, 41(1):108–128, 2020.

[225] Accenture Security. The cost of cybercrime. https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf. Last accessed: November 24, 2020.

[226] Cal. Legis. Comprehensive Computer Data Access and Fraud Act. *California Penal Code, Section 502*, 1989.

[227] Center for Democracy and Technology. Taking the pulse of hacking: A risk basis for security research. https://cdt.org/insights/report-taking-the-pulse-of-hacking-a-risk-basis-for-security-research/. Last accessed: May 27, 2022.

[228] Department for Business Innovation and Skills. Business population estimates for the uk and regions 2018. https://assets.publishing.service.gov.uk/government/

uploads/system/uploads/attachment_data/file/746599/OFFICIAL_SENSITIVE_-_BPE_2018_-_statistical_release_FINAL_FINAL.pdf. Last accessed: March 30, 2019.

[229] DIGIT. Directorate-General for Informatics: Commission launches open source bug bounties. https://ec.europa.eu/newsroom/informatics/item-detail.cfm. Last accessed: March 19, 2019.

[230] European Commission. Publications Office of the European Union: User guide to the SME definition. http://www.innovation.public.lu/en/brochures-rapports/s/sme-definition/sme-definition-user-guide-2015.pdf. Last accessed: March 27, 2019.

[231] European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Publications Office of the EU*, 679:2016, 2016.

[232] Government Communications Headquarters (GCHQ). The equities process. https://www.gchq.gov.uk/information/equities-process. Last accessed: March 4, 2022.

[233] International Organization for Standardization. ISO/IEC 29147:2018: Information technology, security techniques, vulnerability disclosure. https://www.iso.org/standard/72311.html. Last accessed: March 4, 2022.

[234] McKinsey and Company. Perspectives on transforming cybersecurity. https://www.mckinsey.com/~/media/McKinsey/McKinsey%20Solutions/Cyber%20Solutions/Perspectives%20on%20transforming%20cybersecurity/Transforming%20cybersecurity_March2019.ashx. Last accessed: November 18, 2021.

[235] National Institute of Standards and Technology. Guide for conducting risk assessments. https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final. Last accessed: November 18, 2021.

[236] National Institute of Standards and Technology. National vulnerability database. https://nvd.nist.gov. Last accessed: January 27, 2022.

[237] National Telecommunications and Information Administration. Vulnerability disclosure attitudes and actions. https://www.ntia.doc.gov/files/ntia/publications/2016_ntia_a_a_vulnerability_disclosure_insights_report.pdf. Last accessed: May 27, 2022.

[238] Organisation for Economic Co-operation and Development (OECD. Average wages. https://data.oecd.org/earnwage/average-wages.htm. Last accessed: March 4, 2022.

[239] Senate Committee on the Judiciary. The Computer Fraud and Abuse Act of 1986, 1986. United States Congress.

[240] Senate Committee on the Judiciary.
The National Information Infrastructure Protection Act of 1996, 1996.
United States Congress.

[241] Senate Committee on the Judiciary. The Digital Millennium Copyright Act of 1998, 1998. United States Congress.

[242] Senate Committee on the Judiciary. The Anti-Money Laundering Act of 2020, 2020. United States Congress.

[243] The National Cyber Security Centre (NCSC). The cyber security body of knowledge (cybok). https://www.cybok.org/media/downloads/cybok_version_1.0.pdf. Last accessed: November 18, 2021.

[244] US Bureau of Industry and Security. License exceptions, country groups. https://www.bis.doc.gov/index.php/documents/regulation-docs/2255-supplement-no-1-to-part-740-country-groups-1/file. Last accessed: February 21, 2022.

[245] US Bureau of Industry and Security. EAR cyber rule FAQs. https://www.bis.doc.gov/index.php/documents/pdfs/2872-cyber-tools-le-ace-faqs-final-version-nov-2021/file. Last accessed: February 21, 2022.

[246] US Department of the Treasury. Office of foreign assets control - sanctions programs and information. https://home.treasury.gov/policy-issues/office-of-foreign-assets-control-sanctions-programs-and-information. Last accessed: February 21, 2022.

[247] U.S. Government. Vulnerabilities equities policy and process. https://www.whitehouse.gov/sites/whitehouse.gov/files/images/External%20-%20Unclassified%20VEP%20Charter%20FINAL.PDF. Last accessed: September 27, 2019.

[248] Tina Tseng, Amanda Stent, and Domenic Maida. Best practices for managing data annotation projects. *arXiv preprint arXiv:2009.11654*, 2020.

[249] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[250] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[251] Vladislav D Veksler, Norbou Buchler, Blaine E Hoffman, Daniel N Cassenti, Char Sample, and Shridat Sugrim. Simulations in cyber-security: a review of cognitive modeling of network attackers, defenders, and users. *Frontiers in Psychology*, 9:691, 2018.

[252] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. Hackers vs. testers: A comparison of software vulnerability discovery processes. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 374–391. IEEE, 2018.

[253] David S Wall. Cybercrime, media and insecurity: The shaping of public perceptions of cybercrime. *International Review of Law, Computers & Technology*, 22(1-2):45–63, 2008.

[254] T Walshe and AC Simpson. Coordinated vulnerability disclosure programme effectiveness: issues and recommendations. *Computers & Security*, page 102936, 2022.

[255] Thomas Walshe and Andrew Simpson. An empirical study of bug bounty programs. In *2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF)*, pages 35–44. IEEE, 2020.

[256] Thomas Walshe and Andrew Simpson. A longitudinal study of hacker behaviour. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22)*, pages 1465–1474. ACM, 2022.

[257] Thomas Walshe and Andrew Simpson. Towards a greater understanding of coordinated vulnerability disclosure policy documents. *Digital Threats: Research and Practice*, 2023.

[258] Bryan Watkins. The impact of cyber attacks on the private sector. *Briefing Paper, Association for International Affair*, 12:1–11, 2014.

[259] Jake Weidman and Jens Grossklags. What's in your policy? an analysis of the current state of information security policies in academic institutions, 2018.

[260] Claudia R Williamson. Informal institutions rule: institutional arrangements and economic performance. *Public Choice*, 139(3):371–387, 2009.

[261] Jim Witschey, Olga Zielinska, Allaire Welk, Emerson Murphy-Hill, Chris Mayhorn, and Thomas Zimmermann. Quantifying developers' adoption of security tools. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 260–271, 2015.

[262] Marty J Wolf and Nir Fresco. Ethics of the software vulnerabilities and exploits market. *The Information Society*, 32(4):269–279, 2016.

[263] Wooyun. Vulnerability disclosure platform. `https://www.wooyun.org`. Last accessed: January 17, 2019.

[264] Xin Xia, David Lo, Ying Ding, Jafar M Al-Kofahi, Tien N Nguyen, and Xinyu Wang. Improving automated bug triaging with specialized topic model. *IEEE Transactions on Software Engineering*, 43(3):272–297, 2016.

[265] YogOsha. Our clients. `https://yogosha.com/yogosha-clients/`. Last accessed: February 03, 2022.

[266] Gang Yu, Yiwen Yang, Xuying Wang, Huachun Zhen, Guoping He, Zheming Li, Yonggen Zhao, Qiang Shu, and Liqi Shu. Adversarial active learning for the identification of medical concepts and annotation inconsistency. *Journal of Biomedical Informatics*, 108:103481, 2020.

[267] Chunhui Yuan and Haitao Yang. Research on K-value selection method of K-means clustering algorithm. *Multidisciplinary Scientific Journal*, 2(2):226–235, 2019.

[268] Zenan Zhai, Dat Quoc Nguyen, and Karin Verspoor. Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition. *arXiv preprint arXiv:1808.08450*, 2018.

[269] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[270] Mingyi Zhao, Jens Grossklags, and Kai Chen. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proceedings of the 2014 ACM Workshop on Security Information Workers*, pages 51–58. ACM, 2014.

[271] Mingyi Zhao, Jens Grossklags, and Peng Liu. An empirical study of web vulnerability discovery ecosystems. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1105–1117. ACM, 2015.

[272] Mingyi Zhao, Aron Laszka, and Jens Grossklags. Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, 7(1):372–418, 2017.

[273] Mingyi Zhao, Aron Laszka, Thomas Maillart, and Jens Grossklags. Crowdsourced security vulnerability discovery: Modeling and organizing bug-bounty programs. In *The HCOMP Workshop on Mathematical Foundations of Human Computation, Austin, TX, USA*, 2016.

[274] Weiqin Zou, David Lo, Zhenyu Chen, Xin Xia, Yang Feng, and Baowen Xu. How practitioners perceive automated bug report management techniques. *IEEE Transactions on Software Engineering*, 46(8):836–862, 2018.

[275] Aviram Zrahia, Neil Gandal, Sarit Markovich, and Michael H Riordan. The simple economics of an external shock on a crowdsourced 'bug bounty platform'. Available at SSRN 4154516, 2022. SSRN.

Appendix A

Supplementary material relating to programme policies

A.1 Discussion of replicability and reproducibility

A.1.1 Background

Across academic domains there is growing discussion surrounding the replicability and reproducibility of results put forward in published literature that stems from the employment of, often black-box, Machine Learning (ML) models [25]. Using the definitions provided by Liu et al. [153], replicability refers to the ability to reproduce the reported experimental results using the same models and data and reproducibility refers to the ability to apply the same model and protocols to other, previously unseen, datasets and yield similar findings [153]. This is represented in Table A.1.

A plethora of reasons inhibit the ability of researchers to validate findings that reply upon complex ML models, these include: lack of published source code [96], lack of published trained models [134], lack of published hyperparameters or variable descriptions [96], lack of published random seeds [114], use of model hacking (cherry-picking the single best model from many trained models) [180], unknown train and test splits in the data [134], unknown occurrences of data leakage [155, 188], uncontrollable randomness [10], hardware constraints

		Data	
		Same	Different
Approach	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Table A.1: Terminology used to describe experiments that seek to vary the data or modelling approach of the original study. Adapted from [191] and `https://github.com/WhitakerLab/ReproducibleResearch`.

(an issue particularly prevalent with large language model research in the field of NLP [269, 189]), lack of development environment details [32], and code rot [90]. Furthermore, leveraging novel or non-public datasets raises additional concerns about the validity of the claims put forward, as replication may be difficult, if not infeasible, for researchers external to the original study [25].

Sonnenburg et al. [211] propose six underlying reasons that may contribute to a lack of published software or open-sourcing alongside paper publication. These include: misunderstanding that the software / code is not part of the scientific contribution, misconceptions surrounding licensing, lack of incentives to publish the source code, a reluctance to publish 'bad' or messy code, perceived risk of critique from other researchers, and a tendency for reviewers to allow for paper publication without the requirement of source code to be provided. In the context of software engineering literature, Liu et al. [153] found that 74.2% of studies failed to provide sufficient source code and data, and only 10.8% of studies discussed the impact on replicability and reproducibility.

A.1.2 Best practices

In 2019 the NeurIPS conference for research in ML introduced a reproducibility programme to help promote 'open and accessible research' through the use of revised code submission policies, a reproducibility challenge (a conference track, and successor to the original challenge presented at ICLR 2018, focused on replicating the experiments of publication in the wider conference), and the creation of a reproducibility-focused submission checklist [191]. For practitioners, the checklist[1] contains 21 high-level checks across five

[1]Found at `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`

domains, enabling them to help ensure the reproducibility of their work.
Although aimed at the life sciences, the general ML guidance put forward by Heil et al. [111] outlines a series of standards (bronze, silver, and gold) that describe the use of seven best practices for increasing levels of trustworthy analyses.
In order to increase transparency with public datasets, the repositories on Hugging Face[2] (a private company that hosts ML models and relevant datasets) are encouraged to contain a dataset card[3] that pertinent information (e.g. data sources, splits, annotation process, etc.) that may be of interest to researchers looking to leverage the dataset for their own work, or to replicate other studies that make use of the dataset.

A.1.3 Evaluation of replicability and reproducibility

The three aforementioned frameworks of best practices are used to evaluate and communicate the extent to which the results replicable and reproducible, and provide necessary information on the models and datasets. For conciseness, the Reproducibility Checklist [191] is considered in Table A.2, with the standards proposed by Heil et al. [111] and the Hugging Face dataset information card provided in the GitHub repository.

A.2 Examples of Taxonomy Elements

The following examples of the elements set forth in the taxonomy produced by Laszska et al. [145] have been collected from public CVD programmes, and serve to better communicate the elements that make up policy documents.

- *SCOPE-IN* - "The following PayPal brands are in scope: PayPal, Venmo, ..." Example from PayPal's programme on HackerOne.
- *SCOPE-OUT* - "Security issues discovered in the AWS IP Space are not in scope for Amazon Vulnerability Research Program. As an infrastructure provider, AWS customers operate assets in this space. Discovering and testing against AWS and AWS

[2]https://huggingface.co
[3]Template found at: `https://github.com/huggingface/datasets/blob/main/templates/README.md`

Best practice	Comments
Data published and downloadable	**Yes** - All datasets, both raw and annotated, are available in the repository.
Models published and downloadable	**Yes** - Trained spaCy models are available in the repository.
Source code published and downloadable	**Yes** - Python source code files for the data processing pipelines are available in the repository.
Dependencies set up in a single command	**Yes** - A requirements.txt is provided in the repository, allowing for an environment to be quickly set up with the required dependencies. However, the inclusion of a software container may allow for greater flexibility in the future.
Key analysis details recorded	**Yes** - A README file is provided in the repository that outlines all instructions required to reproduce the entire analysis.
Analysis components set to deterministic	**Unsure** - As a GPU was unavailable during the analysis period, it cannot be confirmed whether training and inference using the spaCy library on a GPU will contain only deterministic components.
Reproducible with a single command	**No** - For added clarity, the reproducible steps for the NER and text classification are split into two distinct commands.

Table A.2: Evaluation of study against the best practices of Heil et al. [111]. As determinism cannot be guaranteed for varying hardware set-ups, the study meets the bronze standard for replicability.

customer assets is strictly out of scope for Amazon Vulnerability Research Program and against the AWS AUP." Example from Amazon's programme on HackerOne.

- *VULN-ELIGIBLE* - "Example Topics of Interest: Escalation of Privilege, Information disclosure, Denial of Service, ..." Example from Intel's programme on intigriti.
- *VULN-INELIGIBLE* - "The Netflix Bug Bounty program follows Bugcrowd's Vulnerability Rating Taxonomy with some additional vulnerability classes we consider to be excluded below: ..." Example from Netflix's programme on Bugcrowd.
- *ENGAGEMENT* - "Use your [username]@wearehackerone email alias when testing or reporting bugs." Example from Deliveroo's programme on HackerOne.
- *PROHIBITED-ACTIONS* - "Do not try to exploit service providers we use, prohibited

actions include, but are not limited to bruteforcing login credentials of Domain Registrars, DNS Hosting Companies, Email Providers and/or others. The Firm does not authorize you to perform any actions to a non-GS owned property/system/service/data." Example from Goldman Sachs' programme on HackerOne.

- *LEGAL* - "If legal action is initiated by a third party against you for conduct that Meta determines to have complied with these Bug Bounty Programme Terms, Meta will take steps to make it known, either to the public or the court, that your actions were authorised under this program." Example from Meta's stand-alone programme.

- *PARTICIPANT-RESTRICTIONS* - "We are unable to issue rewards to individuals who are on sanctions lists, or who reside in countries (e.g. Cuba, Iran, North Korea, Syria, Crimea, and the so-called Donetsk People's Republic and Luhansk People's Republic) on sanctions lists." Example from Google's stand-alone programme.

- *GUIDELINES-SUBMISSIONS* - "Provide details of the vulnerability, including information needed to reproduce and validate the vulnerability and a Proof of Concept (POC). Any vulnerability that implicates functionality not resident on a research-registered vehicle must be reported within 168 hours and zero minutes (7 days) of identifying the vulnerability." Example from Tesla's stand-alone programme.

- *GUIDELINES-DISCLOSURE* - "You may not use, disclose or distribute any such Confidential Information, including without limitation any information regarding your Bug Bounty submitted report, without our prior explicit consent. You must get explicit consent by submitting a disclosure request to our program. Please note, not all requests for public disclosure will be approved." Example from Uber's programme on HackerOne.

- *REWARD-EVALUATION* - "Bounty payments are determined by the level of access or execution achieved by the reported issue, modified by the quality of the report. A maximum amount is set for each category. The exact payment amounts are determined

after review by Apple." Example from Apple's stand-alone programme.

- *COMPANY-STATEMENT* - "Microsoft strongly believes close partnerships with researchers make customers more secure. Security researchers play an integral role in the ecosystem by discovering vulnerabilities missed in the software development process. Each year we partner together to better protect billions of customers worldwide." Example from Microsoft's stand-alone programme.