"You've Got Your Nice List of Bugs, Now What?" Vulnerability Discovery and Management Processes in the Wild

Noura Alomar, *University of California, Berkeley;* Primal Wijesekera, *University of California, Berkeley, and International Computer Science Institute (ICSI);* Edward Qiu, *University of California, Berkeley;* Serge Egelman, *University of California, Berkeley, and International Computer Science Institute (ICSI)*

https://www.usenix.org/conference/soups2020/presentation/alomar

This paper is included in the Proceedings of the Sixteenth Symposium on Usable Privacy and Security.

August 10–11, 2020

978-1-939133-16-8

"You've Got Your Nice List of Bugs, Now What?" Vulnerability Discovery and Management Processes in the Wild

Noura Alomar[1], Primal Wijesekera[1,2], Edward Qiu[1], and Serge Egelman[1,2]
[1] *University of California, Berkeley*
[2] *International Computer Science Institute (ICSI)*
{nnalomar, primal, edwardqiu, egelman}@berkeley.edu

Abstract

Organizational security teams have begun to specialize, and as a result, the existence of red, blue, and purple teams have been used as signals for an organization's security maturity. There is also now a rise in the use of third-party contractors who offer services such as incident response or penetration testing. Additionally, bug bounty programs are not only gaining popularity, but also are perceived as cost-effective replacements for internal security teams. Due to the many strategies to secure organizations, determining which strategy is best suited for a given situation may be a difficult task. To understand how these varying strategies are applied in practice and to understand non-technical challenges faced by professionals, we conducted 53 interviews with security practitioners in technical and managerial roles tasked with vulnerability discovery or management. We found that organizations often struggle with vulnerability remediation and that vulnerability discovery efforts are hindered by significant trust, communication, funding, and staffing issues. Based on our findings, we offer recommendations for how organizations can better apply these strategies.

1 Introduction

Security vulnerabilities have caused substantial financial and reputational damage to organizations and users over the years and continue to do so at an alarming rate, despite the availability of advanced tools for hunting vulnerabilities down [4, 6]. Manual code inspection, black- and white-box vulnerability scanners and penetration testing are only a few examples of techniques commonly used for security vulnerability detection before releasing software to users or after putting it into production [12, 18, 26, 28, 30, 33]. We have also witnessed the proliferation of bug bounty and vulnerability disclosure programs that encourage security researchers to assist organizations with vulnerability discovery. Additionally, regulatory bodies have released standards, such as ISO/IEC 29147 and ISO/IEC 30111, that guide organizations on how to design their vulnerability disclosure policies and what to consider after receiving vulnerability reports from external researchers [1, 2]. Despite all these tremendous efforts by the security and regulatory communities, the prevalence of cyber intrusions and leaks suggest that processes involving vulnerability management require urgent improvements.

We shed light on the current state of vulnerability detection and management processes by conducting 53 semi-structured interviews with managerial and technical security practitioners operating in the US and other countries. We contribute to the understanding of interactions, inter-group issues, and the challenges security teams face working with each other. Our findings are informed by industry experiences of red, blue, purple, and penetration testing teams, working as security practitioners in internal security teams or as third-party contractors, who are involved in vulnerability discovery or remediation on a daily basis. We reveal insights into how different roles fit into organizational security strategies and how human factors affect vulnerability management processes.

We interviewed personnel with experience managing and implementing the security posture of their organizations. We identify challenges managerial personnel face and the factors they consider when they make decisions on vulnerability discovery and management (e.g., creating bug bounty programs, outsourcing tasks, etc.). We also identify the challenges professionals from red, blue, purple, pen testing, and bug bounty teams face in practice and how management decisions affect their work. Our study attempts to piece together the perspectives of different actors involved in vulnerability discovery and management processes and identify challenges that require immediate attention by the security community.

USENIX Symposium on Usable Privacy and Security (SOUPS) 2020.
August 9–11, 2020, Virtual Conference.

Vulnerability discovery and remediation processes are hindered by a host of behavioral and organizational challenges. Our results also show:

- an inherent *trust* problem between organizations and testers or bug bounty hunters.
- a lack of effective communication channels between testers who discover vulnerabilities and those responsible for taking proper remediation actions.
- a lack of clarity around who is responsible for fixing discovered vulnerabilities, which turns vulnerability management into a "blame management" problem.
- a misalignment between security and business priorities, organizations represented by our participants often took reactive rather than proactive approaches to vulnerability management.
- a tendency to adopt *compliance-oriented* approaches to security rather than grounding vulnerability handling processes in an understanding of potential risks.

We believe that our findings can propel further research to better understand the human factors impacting vulnerability discovery and management processes in the wild.

2 Related Work

Howard and Lipner [23] studied the security development lifecycle in organizations and observed that implementing security process improvements requires commitment from upper management, which is often hard to get due to the difficulty of associating security improvements with concrete business benefits. Another line of research has focused on the economics of bug bounty programs, the role of white-hat hackers in vulnerability discovery, and how to encourage participation in such programs [13, 17, 19, 22, 25, 27, 34, 36, 37]. Finifter et al. highlighted vulnerability reward programs' potential for helping organizations discover more security vulnerabilities than internal teams [19]. Zhao et al. found that the diversity of contributions in bug bounty platforms correlates positively with the number of discovered vulnerabilities [36]. Maillart et al. found that current incentive structures on bug bounty programs attract security researchers to newly released programs [27]. Researchers also found that the number of vulnerabilities reported by bug bounty hunters correlates positively with the monetary payouts disclosed in program policies [37]. Chatfield and Reddick reported on the Pentagon's successful experience with outsourcing security testing through bug bounty programs [17]. Votipka et al. found that hackers and testers follow similar approaches when searching for vulnerabilities [34]. Building on this work, we combine the perspectives of bug bounty hunters and security managers to shed light on the factors that organizations consider before deciding to create bug bounty programs, and the expectations of bug bounty hunters from bug bounty program managers.

Haney and Paul [21] conducted 14 interviews with members of the blue and red teams at one public organization in the US and identified a number of challenges that stem from a lack of effective information sharing between these teams. Botta et al. found that most organizations follow a decentralized approach to managing cybersecurity and highlighted the need for approaches that facilitate reporting and information sharing across security teams [11]. Björck et al. interviewed security practitioners in Sweden to explore how to address the cost-inefficiencies associated with security management [10]. Werlinger et al. examined incident response processes and emphasized the need for tools that facilitate active collaboration between security teams whenever a response to a security incident is needed [35]. Consistent with our findings, Thomas et al. demonstrated the communication challenges that security and development teams face in practice, and emphasized the role of automation in helping security teams address security issues earlier in the Software Development Lifecycle (SDLC) [32]. Ceccato et al. conducted an experiment to evaluate the usefulness of reports generated by static and dynamic vulnerability detection tools for maintenance teams who are tasked with developing security patches [15]. Assal and Sonia highlighted the need for lightweight approaches to security that can help organizations adopt sustainable approaches to managing security throughout the SDLC [3, 32].

Other research has focused on privacy processes in organizations (e.g., [5]); however, we consider this orthogonal to our study, as the strategic decisions made throughout vulnerability management processes are of higher relevance to securing an organization's entire technical infrastructure, while taking into account human and procedural aspects. Several studies have also addressed the problem of how to improve employees' compliance with security policies, and help security managers assess the security culture in their organizations [7–9, 29]. Furthermore, Stevens et al. [31] conducted a thorough investigation of three security compliance standards and identified 148 security concerns that might render an organization that is considered compliant vulnerable to high severity risks.

While prior work has examined certain individual aspects of vulnerability management processes, our focus is on gaining a holistic understanding of the pipeline, highlighting issues that relate to organizations' strategic planning of vulnerability management, and describing how different teams' efforts help organizations improve their security postures.

3 Methodology

From April to June 2019, we conducted semi-structured interviews with 53 security professionals in different security testing roles (described in the next section). We also recruited current and previous managers who had made important decisions on vulnerability discovery processes in their organizations. We used several channels for recruitment, including advertising on Twitter, and two Slack workspaces (*bugboun-*

tyworld, and *bugbountyforum*), which are used by testers and bug bounty hunters to share resources, collaborate, and communicate with bug bounty platform employees. We also sent emails to several pen testing companies, inviting their security testers and managers to participate in the study.

We conducted six pilot interviews to validate our interview process. Participants recruited for the pilot interviews were contacted via Twitter or through professional connections. One pilot participant did not consent to use his interview data; hence, for the remainder of the paper, we report on the data from 5 pilot interviews and 48 recruited participants. Our study was approved by the relevant Institutional Review Board (IRB). As a token of appreciation, we offered a lottery drawing for five $100 Amazon gift cards.

We asked participants to complete a pre-interview questionnaire, which asked about their security testing experience and their current and prior roles. Next, we conducted a 45-70 minute interview via online video chat software and recorded the audio for transcription purposes (see Appendix A). The lead researcher led every interview, and another researcher participated in an observatory capacity. We conducted interviews until the themes for each category reached saturation [16], which was independently determined by two researchers.

We tailored our semi-structured interviews to participants' roles and prior security testing experiences. The interview process for non-managerial participants evolved around the discovery processes they follow, tools they use, how they communicate with the rest of the teams, challenges they face that might hinder their work, nature of the vulnerabilities they work on, and their expectations. For those with managerial experience, the interview covered their experience working with different teams, reasons for deploying specific strategies in their managerial capacity, and their expectations when deploying different security testing strategies.

Our sample included blue, red, and purple teamers, penetration testers working in internal security teams, security consultants, bug bounty hunters, and security managers (e.g., CTOs and CISOs). At the time of our interviews, a sizable portion of engineers and managers that we interviewed were employed by some of the world's largest tech companies, serving billions of end-users. Twenty of the participants had more than ten years of experience doing security testing, eight participants had between 6 to 10 years of experience, 21 participants had 1 to 5 years of experience, and 4 participants had less than one year of experience doing security testing. Many of our participants held multiple previous positions. For example, some had industry experience in red teaming and pen testing at the same time or held managerial positions after spending some time working as security engineers (see Appendix C). In particular:

- 38 worked as internal or external penetration testers;
- 28 had experience doing bug bounty hunting;
- 23 had held internal or external red teaming positions;
- 17 reported experience working in blue teams;
- 17 were CISOs, CTOs, or engineering managers;
- 11 had experience working as purple teamers; and
- 5 reported working as quality assurance engineers.

The majority of our participants came from the US, but we also had participants from Canada, Germany, UK, India, Israel, Singapore, Serbia, Brazil, and Bangladesh. Some participants managed their own companies, offering external pen testing services. We asked each participant about the number of employees in their organizations and their security teams. For around 40% of our participants, the number of employees in their organizations were in the thousands; eight worked in organizations with 100 to 5,000 employees, while the rest had 1 to 70 employees in their organizations.

Three external transcribers transcribed the interviews. Next, the lead researcher and two other researchers iteratively built the codebook. We used basic thematic coding, wherein each code mapped to an emerging theme (e.g., *blue-red-conflict*, *bugbounty-trust*). Our codebook has 32 major code categories based on the role of the speaker, and those categories are further divided into 88 sub-codes (see Appendix B).

Once the codebook was finalized, the lead researcher and another researcher independently re-coded the interviews and merged the codes to calculate the inter-rater agreement. Disagreements in coding were discussed and resolved after reaching consensus among all the coders. Our recorded inter-rater agreement was 85.36% (Cohens' $\kappa = 0.71$), indicating substantial agreement. We arranged codes into the three phases of the pipeline: strategic decision making, vulnerability discovery, and vulnerability management.

4 Security Teams

There are a number of security roles that engineers might be tasked with in organizations. Many of our participants seemed to be confused as to how to define their work and how their role contributes to vulnerability discovery. For instance, a director of security who manages several security teams and is responsible for product security at a tech company that serves hundreds of millions of users responded:

> *"Yeah pen testing, red teaming, those are very confusing terms. What would you say is the meaning of pen testing?"* (P52).

A CTO of a security consultancy offering vulnerability discovery services to organizations mentioned that many of his clients think that they should do red teaming or pen testing, when what they need is an incident-response team:

> *"I have had clients that have suffered a breach in the past and immediately they are like, 'Hey, we want to get a penetration test or a red team to figure out how the attacker broke in'"* (P37).

We argue that not being able to distinguish between the roles different teams are expected to play might lead to making sub-optimal decisions and ultimately rendering vulnerability discovery processes ineffective. Below, we present a brief definition for each of the different security testing strategies based on themes found in participant responses and our review of the related literature (e.g., [24]).

Internal Testing. We define this role as part of a quality assurance process in which people with security backgrounds either do testing or work with developers to build secure code:

> *"We have kind of check points along the development process. We also have kind of programmatic deployment gates that a handful of these tests must be met before things can even be deployed"* (P10, security manager tasked with managing internal testing teams at a large organization).

Pen Testing. The goal of penetration testing is to find as many vulnerabilities as possible, focusing on a finished product or a service within a pre-defined period:

> *"I see penetration testing as audit. So it is assessing for all vulnerabilities, at every single point in time, within a defined scope"* (P40, a full-time bug bounty hunter with extensive experience as a pen tester and a red teamer).

Red Teaming. Red teaming is goal-oriented and is about going deep (compared to breadth coverage in pen testing); that is, infiltrating the network in question and exploiting any vulnerabilities in the process to reach a pre-defined goal. It is commonly understood that red teaming emulates real-world hackers in the most realistic way possible:

> *"In a red team exercise, you are working with no knowledge of the product. It is as if you are an external hacker kind of thing, trying to hack without access or insider knowledge"* (P31, security engineer tasked with doing internal red and blue team exercises within a large public organization).

> *"The red teaming is really more, hey, we are not interested in identifying every stage of vulnerability in the network, we are interested in seeing how far can we get, what type of data can we access, can we actually exploit the things that are most important to the organization"* (P30, a security manager who has extensive experience leading red teams).

Blue Teaming. Blue teaming is about being defensive—in contrast to red teaming—and concerns monitoring target systems for abnormal behaviors that might indicate the presence of an adversary in the system or an intrusion attempt. Blue teamers are expected to collaborate with red teamers in order to make sure that their efforts complement each other and that they are improving their detection over time:

> *"[A] blue teamer is exactly the opposite, is the defending side. This means finding solutions, techniques, procedures to stop hackers, stop red teamers from succeeding in their attacks"* (P39, an experienced blue teamer).

Purple Teaming. Purple teams might consist of members from red teams and blue teams. The most significant advantage of a purple team is to have both offensive and defensive teams work together to achieve a common goal:

> *"The purple team is the one that navigates between the red team and the blue team and sits with them together as a sort of mediator"* (P39, an experienced blue teamer).

Bug Bounty Programs. Bug bounty programs can be seen as ways to crowdsource pen testing on publicly-facing systems without any time restrictions. Bug bounty programs invite the public or a group of hackers selected by an organization to test its systems according to a pre-defined scope and policy. HackerOne and BugCrowd are examples of bug bounty platforms that act as intermediaries between organizations and external security testers [14, 20].

5 The Vulnerability Management Pipeline

Based on the themes that emerged from the interviews, we identified three phases of an organization's vulnerability management pipeline: from the moment the organization's decision makers realize that they need security testing ("strategic decision making"), to the methods chosen to perform that testing ("vulnerability discovery"), to when discovered vulnerabilities are fixed ("vulnerability remediation").

Strategic Decision Making. We wanted to understand what decision makers expect from the various security testing strategies and what drives them to deploy a particular one. Our goal was to understand whether management are making correct decisions that serve their organizations' goals, whether those organizations are ready for a chosen testing strategy, and whether there are any cultural and/or organizational barriers.

Vulnerability Discovery. In this phase, the chosen security testing teams begin searching for vulnerabilities in target systems. Different security testing teams have different goals, expectations, and time restrictions. Further, some teams could be better suited for specific categories of companies. We believe that it is important to understand the activities carried out by each team and the expectations for each role.

Vulnerability Remediation. This phase concerns deciding how to fix the uncovered vulnerabilities in a timely manner to mitigate potential risks. An effective vulnerability management process entails having a systematic approach to prioritizing discovered vulnerabilities according to their potential impact and having clear communication lines between security testing teams and any other stakeholders.

6 Strategic Decision Making

The potential success of a vulnerability management pipeline hinges upon utilizing the correct vulnerability discovery strategy at the correct time. In this section, we present findings on participants' stated motives and other factors that go into their decision processes. We focus on understanding: (1) when to hire a red, blue, purple, or pen testing team, (2) when to create a bug bounty program, and (3) what considerations are being made before deciding to outsource security testing.

6.1 Deciding Between Different Teams

Below, we present the factors that influence the managerial decision of which security testing strategy to implement.

Red Teaming vs. Pen Testing. We observed uncertainty among our participants about when to do red teaming versus pen testing. The majority of our participants who are considered decision makers did not see the need to do red teaming when their organizations have regular pen testing engagements. Participants expressed different motives for pen testing, a commonly recurring one was to meet various compliance or certification requirements:

> *"Pen tests are kind of a reproducible formula for getting the engagement done whether it is to get that compliance checkbox checked or they are just kind of going through the motions to make sure that basic security structures are in place"*(P10, security manager tasked with vulnerability management operations at a large organization).

Red teaming involves going in depth and emulating a malicious actor. Participants mentioned that another way to look at it is as testing all of the monitoring capabilities installed by the blue team. Thus, to have a fruitful red teaming engagement, an organization should have a reasonably mature security posture with in-depth monitoring:

> *"If my goal is not to find as many vulnerabilities as I can, my goal is to validate my security controls that I have put in place, then I would love to hire a red team"* (P41, a former red teamer and a pen tester who is currently managing application security programs at a large organization).

Regardless of the engagement exercise, participants commonly agreed that neither red teaming nor pen testing should be the first approach to testing in an organization or the first step in their vulnerability management pipeline:

> *"I think most organizations are not mature enough to have true red team exercises performed. I don't think having a red team or having a penetration testing team should come before having an effective vulnerability management program"* (P41, a former red teamer and a pen tester who is currently managing application security programs at a large organization).

We observed that there is a common preference for doing pen tests before putting software products into production and whenever major changes to codebases are made. Red team engagements come at a later stage, when organizations have already tested their products internally and set up monitoring and detection, performed the required pen tests, and are willing to validate the security controls they have in place. However, our results suggest that concerns around exposure to legal liability might drive reluctance around the use of red teams by organizations. Some participants expressed that it would be risky to let external red teaming contractors test their assets without being able to have full visibility into their activities, as they had concerns that their sensitive data might be leaked as a result. This might also create a potential conflict between duty-to-report rules for red teamers and the actual practices of companies. A former red teamer mentioned:

> *"You don't know when someone is in the network, you don't know what he is reading. I know that there are NDAs, but sometimes there is a code of ethic, that says if you find something illegal during a penetration test, you have to report it. And in red teaming, let's be honest not many companies are in regulations with the law"* (P24).

Blue and Purple Teaming. Our participants agreed that organizations should have a blue team that is continuously ready to detect intrusions and respond accordingly. A security director responsible for vulnerability management operations at a leading tech company mentioned:

> *"The first thing that you need to hire is the blue team. What are we defending? What are the state of our systems? What is our attack surface? That stuff you have to first understand"* (P52).

Though, the majority of the participants mentioned that, in most cases, blue teams are not set up until a breach incident happens, which causes upper management to realize the severe consequences of cyber-attacks and the importance of proper security posture. Once an organization properly implements the necessary security controls, red teaming can be utilized to evaluate the extent to which such controls can be exploited by the adversary and help in strengthening blue teams over time. Purple teaming, on the other hand, could be beneficial once a blue team is perceived as not maturing and learning from the results of prior red teaming engagements. However, considering that it would not be realistic for small or medium size organizations to have red, blue, and purple teams, a considerable number of our participants mentioned that they would prefer to start by setting up a purple team, where its members have red team and blue team backgrounds.

Bug Bounty Programs. We found different motives for creating a bug bounty program. Many participants mentioned positive experiences; having many eyes looking into their systems and a large pool of testers with diverse backgrounds can yield interesting vulnerability reports. A security director who started a bug bounty program at a large company, and slowly expanded their scope to cover their vendors as part of their security boundary, pointed out the high costs of pen testing services as the primary motivator for their decision:

> *"One pen tester for one week is going to cost you a lot of money and you can get much higher coverage with just that money and many more people involved"* (P52).

A few participants also had reasons for not creating bug bounty programs. Particularly for financial and governmental organizations, a common theme that emerged in our interviews was that they do not want to risk having unvetted "people from the Internet" accidentally gain unauthorized access to sensitive data stored in their systems. They mentioned that they are more comfortable working with established external security firms, where they can keep tabs on people who are allowed to test their systems for security vulnerabilities:

> *"...trying to convince any government agency to try to offer people a reward to try to break into their financial data, I think we would have a very bad reaction to that, whereas saying, 'it's an audit' would be a much easier sell"* (P44, security leader of a consultancy company that works as a government contractor).

Some participants mentioned that it is often difficult to justify the costs of creating a bug bounty program to upper management, as the costs associated with running such programs are not fixed, relative to paying for an outside audit. Some participants also mentioned that they hire external contractors to conduct pen tests to satisfy regulatory requirements and to convince their customers that they are sufficiently secure because they have been tested by an outside entity:

> *"..the quality of that report you get from pen testing companies is much much higher. It can also hold people who did the pen test accountable"* (P32, a security engineer tasked with red teaming and pen testing at a large security consultancy company).

Participants who had created bug bounty programs mentioned that before creating a bug bounty program, organizations should make sure that they have a robust process for handling reports received from bug hunters: assessing them for validity, reproducibility, severity and impact, and then responding to the bug hunter promptly. Participants said that starting a bug bounty program could easily overwhelm the organization with a lot of noise (false reports). There were also concerns as to whether organizations, especially management, clearly understood the prerequisites:

> *"My CISO likes to be able to tell customers we have a bug bounty program; therefore we are very secure"* (P41, application security manager).

Due to trust-related concerns, several participants preferred to create private bug bounty programs, where the organization gets to pick the bug hunters who are allowed to work on that program after doing proper background checks. One participant also mentioned that his organization decided to do all their bug bounty work on their staging environment, where they do not risk bug hunters getting access to their customers' data, and can make sure that they can reset the testing environment whenever anything goes wrong. Other organizations might prefer to narrow the scope of their public bug bounty program to include a few of their assets and expand the program scope incrementally as they go forward.

6.2 Factors affecting decisions

Below, we summarize other factors that might potentially influence decisions on vulnerability management.

Internal Teams vs. External Vendors. We observed common concerns related to the costs and difficulty of building and maintaining internal teams, which were justifications for outsourcing security testing. The majority of our participants agreed that having internal teams is more beneficial in the long term, as testers accumulate institutional knowledge about their organizations' systems and build understanding of their business logic in more depth over time, as well as have conversations with developers and build relationships with other security team members. Having internal teams also helps organizations adopt a proactive approach to vulnerability discovery. An engineer with more than 10 years of experience in red teaming, pen testing, and managing security teams said:

> *"The internal folks come more or less over time, with a built-in understanding of the business objectives, and the business logic that needs to be expressed. The external folks by definition, don't understand the business as well. And so sometimes their findings are not going to be as relevant to the company"* (P17).

> *"A lot of folks struggle with external bug bounties because folks outside the company will report something and there's no clear implication to the business. And you cannot automatically say this is not a problem. But then you have to decide am I going to spend time even figuring out if this is a problem, versus someone who comes to the table with that work already done for you, then you can immediately start remediating"* (P17).

However, participants also mentioned that external testers can help discover vulnerabilities that were previously not noticed by developers or internal testers, since they often have a lot of exposure to different troubleshooting scenarios and clients. Furthermore, many participants highlighted that most security engagements with external testers are time-scoped and hence bear less cost and logistical overhead compared to maintaining internal teams. This was especially the case for small organizations and is the primary reason why organizations may hire external vendors rather than build and maintain internal teams. Said an experienced red teamer who now works as a consultant in blue and red teaming:

> *"So if I am a company the size of Walmart, I am going to have my own red team. If I'm a small organization, and I don't have the money and IT isn't my focus in the first place, I am not going to try to build a red team and then I would definitely contract with a third party to do it"* (P30).

Furthermore, due to trust-related reasons, participants also mentioned that external testers normally do not get to access every resource they need in order to cover significant parts of their clients' attack surfaces.

Selecting and Vetting External Testing Firms. Participants expressed a tendency to rehire the same external firms:

> *"We, our blue team, prefers a particular pen testing team. We have good relationships with them, their diagnostics and write ups have been just superb"* (P9).

Other explanations included:

- they had a positive experience with a particular firm and built relationships and trust with its members;
- they were satisfied with the level of diagnostic investigation and the quality of write-ups they received;
- they did not want to waste a lot of time vetting different external firms;
- they preferred to work with firms that have a good reputation in the industry; or
- they perceived difficulty in obtaining visibility into how they are improving their security over time once they switch vendors, as different vendors have different methodologies and approaches to security testing.

A few participants, however, mentioned that they prefer to rotate through a pool of vendors to:

- fulfill regulatory requirements;
- get specific testing expertise (e.g., testing cryptographic solutions); and
- get different sets of eyes to test the same codebase.

Regarding the latter, an experienced pen tester stated:

> *"I think it is good to switch from time to time because other people have other techniques and will most likely find other or new issues"* (P15).

Staffing and Budget. Participants mentioned the difficulty of justifying security-related staffing or budget decisions. Participants with management responsibilities admitted that staffing is a big problem, and it is not a luxury that many can afford. In response to why they do not have a bug bounty program, the CISO of a large organization responded:

> *"We already know about more risks than we have the capacity to deal with"* (P1).

This might lead to a lack of visibility into unpatched vulnerabilities and a false sense of security until an incident occurs. Participants also expressed the difficulty of convincing upper management to provide security engineers with the required freedom to do their assessments and find vulnerabilities. Others mentioned that there had been situations where their managers decided to build a security team very quickly to react to security incidents without proper planning and forethought:

> *"...out of the news, all of our [security] costs are basically viewed as not required and extravagant. And it is always a fight to get funding, even though when there is a problem suddenly funding is free and how many people do you need"* (P9).

Participants mentioned that, in some organizations, monetary allocations might give upper management false assurances about the organization's security posture. In cases where an organization has an annual budget allocated for security, upper management might be under the impression that they are *secure* when a regular security assessment with an external contractor is only done as a formality to *'tick the boxes'* and have a report that says so. Said a pen tester:

> *"The CEO does not want to know if you have SQL injection, or XSS. He just wants to see in the report, that we have spent that amount of money on the security, and he always thinks like, 'oh we have spent a lot of money on security. We must be secure!' "* (P24).

Many mentioned that they had to frame everything around a dollar value to get attention. That is, some security managers expressed the importance of conveying risks to the business and describing the impact of potential legal liabilities or monetary losses to upper management:

> *"When you say that this vulnerability has a CVSS score seven, they don't get it. When you can say, like, this risk represents an expected loss of a hundred fifty million dollars, and they're like, 'okay we know what to do with that' "* (P17, security engineer).

Other managers also expressed difficulty in building security teams that have combinations of qualified people with different areas of security knowledge; e.g., network security, mobile security, and cryptography. Some security managers also mentioned the difficulty of making sure that discovered vulnerabilities are addressed promptly by their qualified security engineers, as they are worried they might lose their personnel once they are not satisfied with their workload.

Considering the costs of setting up teams of qualified personnel, several different approaches to managing security in small organizations emerged in our interviews. Outsourcing security to external contractors, using open source security testing tools to automate some tasks, leveraging the security teams of well-known cloud providers by putting most of their security services in the cloud, creating private (invite-only) bug bounty programs, and hiring a few security engineers with purple teaming experience are the main approaches that the participants brought to our attention. Particularly for bug bounty programs, decision-makers working in small organizations were concerned that qualified security researchers would not be attracted to work on their programs, given that they would not be able to compete with organizations that have mature bug bounty programs and provide high payouts. This observation was confirmed by several bug bounty hunters:

> *"I have got two or three programs that I have spent a lot of time on and I keep going back to them because I enjoy working with the team, I know that they pay fairly well and they will turn around my bugs pretty quickly"* (P40, bug bounty hunter).

Some participants mentioned that they have a small team of security engineers wearing multiple hats (i.e., doing offensive, defensive, and quality assurance work at the same time).

Scoping Considerations. Our discussions with internal testers, security managers, bug bounty hunters, and external contractors revealed that there is often uncertainty in how to scope security testing. When defining the scope of a bug bounty program, many decision makers preferred to focus on detecting vulnerabilities that can be reached through publicly available services. Some bug bounty hunters also mentioned that they have accidentally stumbled upon critical vulnerabilities that are considered out of scope, and decided against reporting them, because they fear that the organization might take legal action against them. This fear was confirmed by some bug bounty program managers, who expressed that they would not welcome such discoveries, as they do not want to incentivize bounty hunters to go out of scope.

Other program managers expressed that they would welcome such submissions, as it signals that the bug bounty hunter cares about helping them improve their security posture. Other factors that might influence program managers' decisions include the budget for paying bounty hunters and the capacity of their internal triaging team to process incoming reports. A broader scope is likely to incur higher costs, just in terms of coping with more reports.

In contrast to bug bounty programs, scoping red teaming and pen testing engagements is mostly focused on organizations' internal assets. Although no-scope red teaming engagements are perceived to be common for large organizations, some participants mentioned that narrowing the scope of red teaming could be useful in cases where an organization wants to validate certain controls that they recently implemented. We also observed that some organizations prefer to focus on a specific area at a time; e.g., for the first couple of months they will focus on a specific set of systems and then test other systems later. For pen testing, the majority of our participants indicated that no-scope assessments are often expensive, and this might probably lead them to narrow the scope of systems to be tested in order to afford their costs.

Other security managers mentioned that they are not confident with letting external researchers test every system they have, leading to narrow scopes for pen testing or red teaming engagements. For the same reasons, some organizations resorted to creating duplicate environments, where they put all their code in a live environment, but with sensitive data removed to mitigate the impact of any potential data leakage. From the perspectives of pen testers, we noted that narrowing the scopes of pen testing or red teaming assessments could make the vulnerability discovery process very restrictive and completely counter-intuitive to what the vulnerability discovery process should achieve. Said a red teamer:

> *"...but like it gets to the point where they chop everything off to the point that you don't have a bite anymore and, you know, you want to try as hard as you can, just like any bad adversary"* (P42).

7 Vulnerability Discovery

Vulnerability discovery can start from the earliest phases of the software development lifecycle. It can be part of each cycle or can be brought in after each major code release. An organization can have a dedicated security team testing all of the ongoing projects or assign a team member(s) to work with each development team(s). Below, we outline the processes security teams follow to discover security vulnerabilities based on a synthesis of common understandings in the industry and our observations from the interviews.

Pen Testing Activities. Pen testers aim for breadth by finding as many vulnerabilities as possible within a predefined scope and time frame. Participants also mentioned that they often get the help of external pen testing contractors to test their products before shipping them to users. Pen testers are not supposed to hide their activities from other members of the organization and are mainly expected to demonstrate the vulnerabilities they discover to prove that they exist, without exploiting them. However, one external pen tester mentioned that clients sometimes ask them to exploit identified vulnerabilities to demonstrate their impact. Furthermore, most pen testers' activities are supposed to be continuously monitored by members of the client organization to limit the consequences of any potential unintended exploitation.

Red Teaming Activities. One of the main goals of red teaming is to help blue teamers improve the defenses they have put in place. Pen testing and red teaming bring different benefits to an organization. Red teaming engagements normally take longer than pen testing engagements. Pen testers are usually given access to internal resources, whereas red teamers start their engagement with minimal to zero information. For instance, some participants mentioned that pen testers usually have internal network access, whereas for red teamers, gaining access to internal networks is a goal they should achieve. Furthermore, pen testing engagements are not stealthy, and many in an organization might be aware of them, whereas red teaming engagements are stealthy by nature to evade any monitoring or access control mechanisms placed in the organization and to effectively simulate the activities of malicious attackers. For instance, some of the tactics red teamers follow to decrease the risk of detection are:

- not installing custom software;
- trying to use system functionality as much as possible, rather than using exploits;
- sticking to regular user working hours, so a blue team would not suddenly start noticing loads of traffic;
- rotating their IP address or hiding them behind different VPN providers; and
- not scanning many ports at once.

Participants explained that there are basic steps that are usually followed by red teamers. At a very high level, the first phase is reconnaissance, which involves gathering as much public information about the target as possible. Based on that, red teamers can draw a plan outlining the potential targets that they can attempt to attack or utilize to exfiltrate data. For instance, they might send phishing emails in the hope that someone will click on a malicious attachment, which will then help them establish a foothold into the network. They could also examine whether lateral movement is possible and try to plant a backdoor in order to maintain persistence. To effectively simulate real-world adversaries, red teamers usually ask organizations not to inform their employees—including blue teams—that these offensive security engagements are taking place; once blue teamers detect them or they successfully evade a blue team's security controls, they communicate their findings and provide guidance on how to fix the identified vulnerabilities. Red teams need to have broader skill sets, including social engineering, physical security, and network security. Participants also mentioned that pen testing experience would be beneficial for red teaming.

Blue Teaming Activities. Blue teaming is concerned with fixing vulnerabilities found in the discovery phase or implementing proper monitoring capabilities to prevent intrusions. Some examples of blue teaming activities that were brought up in our interviews are installing firewalls and anti-virus software, monitoring systems for suspicious activities, responding to cyber incidents, trying to confuse potential adversaries once they are detected in order to gain an improved understanding of their activities and capabilities, and auditing and analyzing logs. Some blue teamers also mentioned that most of their work is concerned with fighting bad security culture, convincing people to install patches and teaching engineers the best security practices. It is also worth noting that, in contrast to red teams, blue teams usually have access to significant internal resources and are therefore expected to know the specifics of what they are defending. For this reason, asset management is perceived as a challenging task for internal blue teams. That is, they cannot protect systems that they do not know about.

Purple Teaming Activities. We observed common confusion surrounding what purple teams do in practice; some thought the term signals that an engineer has red and blue team experience, while others thought that applying offensive and defensive security practices at the same time can significantly contribute to hardening defenses in organizations. While many of our participants mentioned the difficulty of building a purple team, we noted that small organizations prefer to start with setting up purple teams without having separate blue and red teams due to decreased staffing costs.

In organizations with red and blue teams, purple teams can act as middlemen who facilitate communication between the two teams by making sure that they have regular meetings and regularly exchange information. Other organizations prefer to assign some members of the red team to work closely with blue teamers, to train them to detect the attacks mounted by red teamers or reflect on the regular feedback they get from red teamers. Another way of doing purple teaming exercises is to have blue and red teams work very closely, where all team members are involved in the same exercises. In such settings, once a red teamer manages to break defenses, the red teamer works with blue teamers to address the identified vulnerability. Afterwards, the two teams can resume their work and reflect on their progress.

Bug Bounty Hunters' Activities. Bug bounty hunters consider reconnaissance a critical phase in their vulnerability discovery processes, in which they attempt to collect as much information as possible about the target systems. This phase includes activities such as enumerating all sub-domains, using a service as a regular user to understand its functionality, parsing public datasets, understanding data flow, and identifying dependencies between different features or services. Some examples of the techniques used for vulnerability discovery that we identified in our interviews are disabling certificate pinning, reverse engineering apps, looking for default credentials, discovering old services, and watching for DNS changes.

Inter- and Cross-Team Communication. One of the main themes that emerged in the majority of our interviews with security engineers concerns the communication challenges security teams face with other internal teams. From the perspectives of red teamers and pen testers, they mentioned that their findings are often not welcomed by blue and development teams and that they are generally perceived as threats to other teams; red teamers' findings might make development/blue teams look bad to upper management or potentially increase their workload. Our results therefore suggest that establishing cooperative and collaborative relationships between blue and red teams seems to be a serious challenge.

In most cases, this might turn into a *"blame-management"* problem, in that each team thinks that fixing the discovered vulnerabilities is not part of their job. Several participants mentioned that they had had situations where they decided to report their discoveries to upper management to ensure that the identified vulnerabilities got fixed. From the perspectives of external contractors, they mentioned that they often report their findings to the person who hired them and might not get a chance to interact directly with internal teams who are in charge of fixing discovered vulnerabilities. Participants also noted that red teams are supposed to not disclose their activities or share much information about their testing strategies to make their simulations realistic, and this can be frustrating for blue teamers. From a managerial perspective, how security teams are organized and structured could introduce some communication problems. Said a security manager:

> *"..a lot of organizations diversify teams too quickly so where they have five or six people on a dozen different islands with different names. And they lose a lot of the collaborative potential of the team whenever they're segmented off that way"* (P10).

To address these communication problems, participants mentioned some strategies followed by their organizations:

- situating security teams within engineering teams;
- letting different security engineers test the same product at different points in time over the year and comparing their findings to keep track of what vulnerabilities have been fixed and what still needs to be done;
- setting up a purple team; and
- holding regular security-related meetings with representatives from each team in the organization.

Some also raised the point that having clear objectives and increasing the level of transparency with security engineers could make vulnerability discovery processes more fruitful.

From the perspective of bug bounty hunters, participants have had a mix of positive and negative experiences when reporting their findings to internal teams. Many of the bounty hunters we interviewed mentioned that the bug bounty ecosystem has allowed them to build productive relationships with internal security teams at various companies, allowing them to collaborate on fixing reported vulnerabilities and receiving prompt feedback. Some mentioned that their work as bug bounty hunters increased their chances of getting hired at the companies to which they reported vulnerabilities. Several participants also mentioned that they had situations where internal teams ignored their vulnerability reports or considered the reports as out of scope because the internal teams did not triage the reports correctly.

8 Vulnerability Remediation

Below, we summarize our observations that relate to fixing the vulnerabilities uncovered during the discovery phase.

Triaging Vulnerability Reports. Internal teams have to triage vulnerability reports they receive in order to assess the severity level of each reported vulnerability and prioritize fixing the critical ones. Participants described their experiences with bug bounty programs that assigned non-technical people to the task of triaging vulnerability reports, which resulted in closing their reports as insignificant when the triaging team did not fully understand the impact of the findings.

Fixing Vulnerabilities. With regard to addressing vulnerabilities, one main theme emerged: internal teams' inabilities to fix reported vulnerabilities due to the lack of detailed information in the reports that allow reproducing the vulnerabilities and thus make informed decisions on how to fix them. For example, bug bounty hunters mentioned that patches applied by internal security teams could often be bypassed. For this reason, some security managers mentioned that they sometimes offer extra bounties or reputation points for bounty hunters willing to help with remediation. However, participants mentioned that providing extra incentives is not common and that the current bug bounty ecosystem does not incentivize external researchers to go beyond surface-level checks to ensure that reported vulnerabilities have been properly fixed:

> *"There is one program that I've worked on that specifically says you will receive a high bounty if you provide a remediation advice. So it is not something that I see across the board in having recommended fixes, I tend to think that the teams themselves are in a much better place to know how to fix it, that I'm here to explain the issue in a way that they can understand it"* (P40, full-time bug bounty hunter).

Others thought it would be helpful to ask bug bounty hunters to retest the fixes released by companies:

> *"I think it's good to send it back to you after they fix it and ask you to check it because they might have put a patch that wasn't fully secure. So I think that is a great way to just get confirmation that it is fixed"* (P25, bug bounty hunter).

A participant with extensive pen testing experience stated that external security testers usually lack the specifics needed to describe how to fix identified vulnerabilities. On the flip side, from internal teams' perspectives, some participants stated that they often do not have a clear idea of how a vulnerability reported by an external tester was discovered in the first place, which is one of the reasons why they might have no way to validate the fix. Other participants mentioned that they arrange for external pen testers to meet with internal teams to discuss possible remediation strategies. External pen testers might also be asked by the organization to do another pen test after a certain period to check whether the identified vulnerabilities still exist.

From security managers' perspectives, many stressed the importance of providing comprehensive and detailed mitigation strategies by security testers to allow internal teams to find an alternative approach to fixing a vulnerability, once they find it infeasible to fix it in a particular way.

When discussing the extent to which internal teams are capable of addressing vulnerabilities promptly, an experienced blue teamer explained:

> *" a lot of them don't know how to test whether or not the patch solved the problem. I think that the biggest reason is skill"* (P9).

Other barriers hinder the remediation process. For example, internal teams might identify dependencies between different applications that might involve other internal departments or external vendors, or require hiring additional software developers with special expertise. This might decrease the likelihood of applying fixes promptly. Some security managers also mentioned that lack of transparency between all the internal and external parties involved in security processes is often an obstacle towards effective vulnerability remediation:

"We have trained our engineers to not come to us, but we have lost visibility of everything else that is happening, and now we are, back to a super reactive mode, where we want to be in a super proactive mode" (P43).

Compliance vs. Security. Depending on the industry, there are specific compliance requirements that must be met in order to make sure that the organization is doing the bare minimum to secure their users or customers. Such requirements have made organizations to adopt compliance-oriented rather than security-oriented mindsets. Most of the participants who have experience working as external pen testers described their experiences with organizations that ask them to downplay the severity rating of a critical vulnerability in the reports they plan to submit to regulators or to take some vulnerabilities out of a report when they do not have enough time to fix them.

This is made worse when pen testers are pressured by their managers to comply with these client requests to maintain relationships. These external pen tests are sometimes perceived as a way to shift the liability and accountability for security breaches from the client organization to the pen testing contractor. Said two pen testers:

"We've been told we don't want you to actually solve this problem, we just want you to make the check box go away" (P9).

"I think that turns pen tests into commodity and not clients that really want to understand the exposure and ultimately not necessarily interested in actually fixing things and getting a better security" (P38).

9 Discussion

In this section, we analyze the challenges that came up in the interviews, which we believe can be addressed by paying more attention to human factors.

Security Is a Reactive Measure. Security testers and engineers mentioned that convincing management that security should be a priority is hard in many organizations. Organizations that struggle to coordinate to place security as a high priority often spend more time reacting to incoming vulnerabilities than proactively searching for vulnerabilities. The reactive approach to vulnerability discovery and remediation may increase the chances of organizations falling victims to security attacks despite advances in vulnerability detection tools and techniques. Staffing and budgeting costs could also push organizations to be conservative in their investments in security, particularly in small organizations.

Insufficient Attention to Vulnerability Remediation. Participants mentioned that management might invest in security for reasons other than securing their systems (e.g., fulfilling compliance requirements). Although the byproduct of these intentions might still lead organizations to invest in vulnerability discovery, the downstream effects of finding a vulnerability may not be effective for fixing the root cause of the vulnerability. Compliance regulations often require organizations to establish and follow their own processes to handle vulnerabilities, with compliance often checked by periodic audits (e.g., every 6 months). Since compliance is checked at discrete intervals instead of continuously, organizations that invest in security for compliance purposes may only address discovered vulnerabilities once an audit is scheduled, so that they may develop a corpus of events for the purpose of passing the audit. This is likely to render vulnerability discovery efforts ineffective since discovered vulnerabilities might not be remediated correctly or promptly.

Trust. When working with external testers, one of the most significant challenges faced by organizations and internal testers is trusting external testers with their systems and data. This can discourage organizations from creating public bug bounty programs or working with external red teaming firms. Trust can also be a factor in deciding what components are in-scope for testing engagements, and this might increase the chances of leaving some vulnerabilities undetected. Our results suggest that many organizations tend to put rules in place that limit what red teamers can realistically do to simulate the adversary. Said the director of a pen testing firm:

"..everybody is talking about they want a red team, they want a red team, but at the end of the day, they want to put a bunch of rules around it, just like regular penetration tests" (P38).

For the same reason, many bug bounty hunters expressed that they often feel hesitant to report vulnerabilities they discovered to responsible entities, as they fear reprisal from organizations. This is likely to discourage external security researchers from reporting their discoveries or collaborating with organizations to improve their security posture.

Communication. Lack of communication had caused issues with the reproducibility of reported vulnerabilities when the person who reported a vulnerability does not provide sufficient detail (e.g., the feasibility of exploitation and whether the reports contain realistic exploits). That is, they lack the knowledge required to put the vulnerabilities in the context of the organization to allow assessing the severity and the impact of reported vulnerabilities correctly. This is a clear indication that the industry lacks a proper standard for communicating discovered vulnerabilities to relevant entities. One standard that does exist for assessing severity levels is the Common Vulnerability Scoring System (CVSS). However, participants stated it is not helpful because it lacks sufficient consideration for business requirements or organizational contexts, which can facilitate communication with testers, internal teams, and management as to whether a reported vulnerability is critical or not. Without an effective vulnerability rating mechanism, vulnerabilities with high impact can go unnoticed or issues that require proper attention might not get it. Another important line of communication is between the security engineers and the management who decide on the budget and staffing. Participants mentioned that they had to put a monetary value on security issues that require immediate attention to garner attention from management. Furthermore, one CISO mentioned that finding a responsible point of contact for each independent system in the event of a breach or incident is especially challenging in large organizations.

Creating a security culture that is driven by a genuine interest in defending technical infrastructure can foster collaboration between developers and testers around resolving security issues. Surprisingly, many of the internal and external testers we interviewed raised the point that they often experience problems concerning receptiveness to their security feedback. One participant mentioned that developers might feel embarrassed or blamed once someone reports a vulnerability in their code. Such attitudes to security are likely to discourage developers from acknowledging the receipt of vulnerability reports and security teams from following up on whether a reported vulnerability has been fixed.

10 Recommendations

In light of our findings, we propose a set of recommendations that we expect to help organizations improve their vulnerability management processes.

First, it is essential to identify the technical and business owners of the various systems an organization has early in the process, so that remediation tasks can be assigned smoothly to their corresponding owners, who can take the lead in deciding how to plan subsequent remediation efforts. We would expect there to be guidelines to help identify which stakeholder to notify to improve the level of transparency between the teams involved and reduce the time from discovery to remediation.

Second, organizations should establish a clear set of risk priorities for upper management that communicates the risks an organization is willing to take, in order to guide decisions concerning vulnerability discovery and remediation. This would also allow security teams to communicate the relevance of discovered security vulnerabilities to the organization's priorities and business goals and therefore facilitate discussions that involve convincing upper management to allocate more resources for vulnerability discovery or remediation. Organizations should also have clear procedures for reporting potential risks to upper management and facilitating decisions that concern resource allocation, coordination among teams and stakeholders, and separation of responsibilities between security, development, and legal teams.

Third, we recommend having clear expectations of the maximum period of time that should elapse from the moment a vulnerability is discovered until it is completely remediated by its technical owners. Such expectations should be based on pre-defined criteria for assessing the severity and potential risks of reported vulnerabilities and assigning a timeline for implementing remediation plans that allow addressing critical vulnerabilities in a timely fashion.

Fourth, we recommend that organizations do not rush to create bounty programs, unless they have solid remediation processes in place. One of the prerequisite to creating a bug bounty program is performing internal security assessments (e.g., pen tests) and remediating all discovered vulnerabilities. A proper assessment of the costs of creating such programs, triaging vulnerability reports, assigning discovered vulnerabilities to their technical and business owners, and carrying out subsequent remediation efforts should also be performed before creating these programs.

Fifth, we recommend improving the current incentive structure implemented in bug bounty programs by providing higher bounties for researchers who include suggestions on how to address reported vulnerabilities or help with testing released fixes. We also recommend involving bounty hunters in vulnerability remediation and encourage internal testers to get in touch with them to explore the available remediation options and get confirmation that a released patch fully remediated the discovered vulnerability. It is also important to define what is in-scope or out-of-scope for bug bounty programs based on a pre-defined threat model, which considers the organization's security objectives and the risks that it aims to mitigate.

Sixth, we recommend customizing the language used in communications between different security teams or other stakeholders to the language that can be clearly understood by the target audience. Communications with upper management should be framed in terms of how a discovered vulnerability might introduce risks to the business to help them make informed decisions. On the other hand, blue teams and development teams might need a mix of technical and business-oriented discussions to guide their decisions on how to fix reported vulnerabilities and how to prioritize addressing them based on their potential impact to the business.

11 Conclusion

We present the results of an investigation of modern vulnerability management processes that focused on exploring the tensions between different security engineering and management roles in small, medium, and large organizations operating in the public and private sectors. We show that while the technical aspects of computer security are imperative for securing organizations' technical infrastructures, these efforts can be hindered by human factors such as a lack of trust, ineffective communication between security teams, and unwillingness to invest in security by upper management.

Acknowledgments

We would like to thank all of our study participants for participating, as well as Amit Elazari and all the anonymous reviewers for their insightful feedback. This research was supported by the Center for Long-Term Cybersecurity at U.C. Berkeley.

References

[1] ISO/IEC 29147: Information Technology – Security Techniques – Vulnerability Disclosure. https://www.iso.org/standard/45170.html, 2019. [Online; accessed: 27-June-2019].

[2] ISO/IEC 30111: Information Technology – Security Techniques – Vulnerability Handling Processes. https://www.iso.org/obp/ui/#iso:std:iso-iec:30111:ed-1:v1:en, 2019. [Online; accessed: 27-June-2019].

[3] Hala Assal and Sonia Chiasson. Security in the software development lifecycle. In *Fourteenth Symposium on Usable Privacy and Security*, pages 281–296, 2018.

[4] Andrew Austin and Laurie Williams. One technique is not enough: A comparison of vulnerability discovery techniques. In *International Symposium on Empirical Software Engineering and Measurement*, pages 97–106. IEEE, 2011.

[5] Kenneth A Bamberger and Deirdre K Mulligan. *Privacy on the ground: driving corporate behavior in the United States and Europe*. MIT Press, 2015.

[6] Jason Bau, Elie Bursztein, Divij Gupta, and John Mitchell. State of the art: Automated black-box web application vulnerability testing. In *IEEE Symposium on Security and Privacy*, pages 332–345, 2010.

[7] Adam Beautement, Ingolf Becker, Simon Parkin, Kat Krol, and Angela Sasse. Productive security: A scalable methodology for analysing employee security behaviours. In *Twelfth Symposium on Usable Privacy and Security*, pages 253–270, 2016.

[8] Ingolf Becker, Simon Parkin, and M Angela Sasse. Finding security champions in blends of organisational culture. *USEC*, 11, 2017.

[9] Ingolf Becker, Simon Parkin, and M Angela Sasse. The rewards and costs of stronger passwords in a university: linking password lifetime to strength. In *27th USENIX Security*, pages 239–253, 2018.

[10] Fredrik Björck. *Discovering information security management*. Department of Computer and Systems Sciences, Stockholm University, 2005.

[11] David Botta, Rodrigo Werlinger, André Gagné, Konstantin Beznosov, Lee Iverson, Sidney Fels, and Brian Fisher. Towards understanding it security professionals and their tools. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 100–111. ACM, 2007.

[12] Mehran Bozorgi, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Beyond heuristics: learning to classify vulnerabilities and predict exploits. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 105–114. ACM, 2010.

[13] Lorenz Breindenbach, Phil Daian, Florian Tramèr, and Ari Juels. Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts. In *27th USENIX Security Symposium*, pages 1335–1352, 2018.

[14] BugCrowd. https://www.bugcrowd.com/. Accessed: September 16, 2019.

[15] Mariano Ceccato and Riccardo Scandariato. Static analysis and penetration testing from the perspective of maintenance teams. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 25. ACM, 2016.

[16] Kathy Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis*. Sage, 2006.

[17] Akemi Takeoka Chatfield and Christopher G Reddick. Cybersecurity innovation in government: A case study of us pentagon's vulnerability reward program. In *Proceedings of the 18th Annual International Conference on Digital Government Research*, pages 64–73. ACM, 2017.

[18] Anne Edmundson, Brian Holtkamp, Emanuel Rivera, Matthew Finifter, Adrian Mettler, and David Wagner. An empirical study on the effectiveness of security code review. In *International Symposium on Engineering Secure Software and Systems*, pages 197–212. Springer, 2013.

[19] Matthew Finifter, Devdatta Akhawe, and David Wagner. An empirical study of vulnerability rewards programs. In *Presented as part of the 22nd USENIX Security Symposium*, pages 273–288, 2013.

[20] HackerOne. https://www.hackerone.com/. Accessed: September 16, 2019.

[21] Julie M Haney and Celeste Lyn Paul. Toward integrated tactical operations for red/blue cyber defense teams. In *Workshop on Security Information Workers at Symposium on Usable Privacy and Security*, 2018.

[22] 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, 2017.

[23] Michael Howard and Steve Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.

[24] Steven J Hutchison. Cybersecurity: Defending the new battlefield. Technical report, Defense Acquisition University, Fort Belvoir, VA, 2013.

[25] 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.

[26] V Benjamin Livshits and Monica S Lam. Finding security vulnerabilities in java applications with static analysis. In *USENIX Security Symposium*, volume 14, 2005.

[27] 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.

[28] Charles P Pfleeger, Shari Lawrence Pfleeger, and Mary Frances Theofanos. A methodology for penetration testing. *Computers & Security*, 8(7):613–620, 1989.

[29] Lena Reinfelder, Robert Landwirth, and Zinaida Benenson. Security managers are not the enemy either. In *Proceedings of the Conference on Human Factors in Computing Systems*, page 433. ACM, 2019.

[30] Eric Rescorla. Is finding security holes a good idea? *IEEE Security & Privacy*, 3(1):14–19, 2005.

[31] Rock Stevens, Josiah Dykstra, Wendy Knox Everette, James Chapman, Garrett Bladow, Alexander Farmer, Kevin Halliday, and Michelle L Mazurek. Compliance cautions: Investigating security issues associated with us digital-security standards. In *Network and Distributed System Security Symposium (NDSS)*, 2020.

[32] Tyler W Thomas, Madiha Tabassum, Bill Chu, and Heather Lipford. Security during application development: an application security expert perspective. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, page 262. ACM, 2018.

[33] Herbert H Thompson. Application penetration testing. *IEEE Security & Privacy*, 3(1):66–69, 2005.

[34] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. Hackers vs. testers: A comparison of software vulnerability discovery processes. In *IEEE Symposium on Security and Privacy (SP)*, pages 374–391. IEEE, 2018.

[35] Rodrigo Werlinger, Kasia Muldner, Kirstie Hawkey, and Konstantin Beznosov. Preparation, detection, and analysis: the diagnostic work of it security incident response. *Information Management & Computer Security*, 18(1):26–42, 2010.

[36] Mingyi Zhao, Jens Grossklags, and Kai Chen. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proceedings of the ACM workshop on security information workers*, pages 51–58. ACM, 2014.

[37] 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.

A Interview Guide

The following subsections include the questions we used in our pre-interview questionnaire and the ones we asked the participants in our semi-structured interviews. Some of the questions were asked in all the interviews, whereas the remaining questions were chosen based on the current role/job title of the participant.

A.1 Pre-Interview Questionnaire

1. How long have you been working as a hacker or a tester? (Choices: Less than 1 year, 1 to 5 years, 6 to 10 years, More than 10 years)
2. How did you learn hacking/vulnerability discovery skills? (select all that apply) (Choices: Reading blog posts, Following hacker write-ups on bug bounty platforms, Following hacker write-ups on GitHub, I am majored in CS, Cybersecurity professional training, Other)
3. What is your job title (employment status)?
4. What other roles have you had in the past (select all that apply)? (Choices: Internal pentester, External pentester, Internal red teamer, External red teamer, Blue teamer, Quality assurance engineer, Purple teamer, Bug bounty hunter, CISO, other)
5. Briefly describe your career path (e.g., bug bounty hunter, pen tester and then red teamer).
6. Do you participate in bug bounty programs? (If yes, how many times have you been awarded a bug bounty? and how many years have you been working as a bug hunter?)

7. Have you received formal security education in computer security or software engineering? if yes, please list the degrees and security certifications you have.
8. In your opinion, what type of education or certifications you think helped you most with the type of software security testing you do?
9. How many people work at your company?
10. How many security professionals does your company have?
11. What is your company's main line of business?

A.2 Interview questions for all groups

1. How would you structure your testing process if you were explaining it to a university student?
2. Why do you think your role is important in testing/vulnerability discovery? (follow up question for managerial positions: How would you decide to use a given testing team and at what point? What sort of factors goes into that decision?
3. Based on your experience, what is the contribution of each team? And how would each of them fit into the testing pipeline?, How do you decide what software gets tested, how it gets tested and when it gets tested?, and How does the time frame affect your approach to testing?)
4. Are there any common mistakes or incorrect practices that you often notice people do when it comes to testing the target software?
5. Can you list a set of metrics that you find useful for quantifying your success/failure with regards to what you found in the software you are testing?
6. How do you define security-related test cases and based on what?
7. What happens after you submit the reports to other teams? (follow up question: In order to make the process more efficient, what are the changes that you want/expect to see after you report a vulnerability? (i.e., what are some ways the flow of the vulnerability reporting would be improved throughout an organization?))
8. How does the testing process work for projects that follow agile methodologies such as Scrum?
9. Briefly describe your recon methodology. (follow up questions: What is the type of information you find useful for the type of testing you do? and What are the main sources of internal and external information you normally rely on to complete your testing work including training and knowledge flows?)
10. What sorts of tools are you using? And what tools do you find useful for testing?
11. Do you prefer static analysis over dynamic analysis? And why?
12. What are the different testing phases that you use tools for?
13. How do you decide what tools to get to your teams? What factors can help you decide on that? Please describe your tool selection process and are there any restrictions in terms of what tools could your engineers use? (this question is for managerial positions such as CISOs)
14. Describe the role of 'automation' in terms of the types of tasks you do.
15. How would you define the scope for software/web application testing?
16. Do you think that scoping is limiting you in terms of what types of testing you can do? If so, why?
17. Are there any other organizational barriers/workplace policies or/and contractual limitations that could restrict your team from carrying out certain types of testing. If yes, please explain.
18. What are the vulnerability types you normally look for?
19. What are the vulnerability types that are commonly and consistently found in different projects?
20. Would you treat zero-days differently? If yes, why? And what is the process you follow for that?
21. Do you interact with other (red, blue, purple, pen testers, etc) teams on your day to day job?
22. How often do you communicate with other teams as part of your job vs how often do you collaborate with another team to get the job done? And, under what circumstances?
23. Have you ever had the chance to discuss the security vulnerabilities you found with software developers? If yes, please comment on the impact of such discussion on improving the security of the software you were testing.
24. Explain how did you acquire your current skill set?
25. How do you stay up-to-date with the latest discovered vulnerabilities and attacks? And how do you reflect this knowledge in your testing work?

A.3 Questions for red teams

1. What are the factors that drive companies to decide to create or contract with a red team?
2. How are red teams structured?
3. What is the frequency in which you do red teaming assessment in a year? What factors affect the number of assessments done in a year?
4. Describe the workflow of red teaming.
5. What is the most common ways you establish an initial foothold on an organization?
6. What are the tactics you employ to remain undetected by blue teams?
7. What are the rules of engagement that you usually make sure are followed?

8. What are the common mistakes that you observed blue teamers or other testing teams do? and what are your recommendations on how to make their job better?
9. Do you talk/teach/explain blue teams about your findings? If so, how regular are these meetings? and, what sorts of information do red teams/blue teams share with each other?
10. The basic idea behind red teaming is to challenge assumptions and identify blind spots, do you have any special standard techniques/checklists? And how do you normally operate? Where do you start?

A.4 Questions for blue teams

1. What are the factors that drive companies to decide to create a blue team?
2. How are blue teams structured?
3. Describe the workflow of blue teaming.
4. What information can help you differentiate between attacks mounted by red teams or real attackers?
5. How is your job different from red teamers' job? Can you describe red teamers' methodology to testing?
6. What are the common mistakes that you observed red teamers or other testing teams do? and what are your recommendations on how to make their job better?
7. Do you communicate with red teams? If so, how regular these meetings are? and, what sorts of information do red teams/blue teams share with each other?
8. If you get an alert, how do you respond? Do you have a process for that? And what bottlenecks do you normally experience throughout the process?

A.5 Questions for purple teams

1. What are the factors that drive companies to decide to create a purple team?
2. What is unique about your purple team from the blue team and the red team? and what is your team's role in the organization?
3. How are purple teams structured?
4. Describe the workflow of purple teaming.
5. Do you expect to have separate blue and red team when there is a purple team?
6. How to improve communication and workflow between red and blue teams?
7. What are the skillsets you share with blue teamers? and what are the skillsets you share with red teamers?
8. Are there differences in the sets of tools used by blue teams and red teams?
9. Do you see areas where improvements can be made in terms of elevating blue and red teamers' chances of finding security vulnerabilities more efficiently and improving the effectiveness of their testing work?

A.6 Questions for penetration testers

1. To what degree do you rely on automated vulnerability detection tools as opposed to manually learn about the systems of interest?
2. In your opinion, how to improve testers' vulnerability discovery capabilities?, how differently companies could use pen testers efficiently than the current practices? and what are the obstacles that could hinder your work?
3. Do you follow a systematic approach to finding vulnerabilities? Please explain.
4. Do you target your recon based on knowledge of an organization's business interests? If so, how?
5. How do you draw your penetration testing plans? is it based on the contract? and does the penetration testing team have the freedom to come up with its own testing plan?
6. How and when do you start to collect information about a target?
7. In your opinion, what are the reasons that drive companies to choose to outsource security testing to an external pen-testing company rather than doing the tests internally by their internal testing teams?
8. What are the types of sensitive details that companies normally disclose to third-party pen-testing services?
9. Can you comment on the nature of non-disclosure agreements that contractors must sign before starting the pentesting work? and can you comment on whether these agreements are affecting the efficiency or effectiveness of penetration testing?
10. What mechanisms are used to assure accountability of actions taken by pen-testing teams? e.g., do companies keep logs that allow them to inspect what pen testers have done and investigate whether there is some degree of damage that resulted from pen testers actions?
11. Do companies archive information collected during pen testing? And how do they make sure that this information is not accessed by unauthorized people?

A.7 Questions for managerial positions such as CISOs

1. What are the factors that drive companies to decide to create an internal red/blue/purple team or involve an external red team in their pentesting processes?
2. In your opinion, when should an organization consider creating a bug bounty program?
3. Describe the dynamics of interaction that occur between different teams.
4. Do you have an internal red team, pentesting or application testing team? What is the main function of each team? And how do they integrate into the rest of the security ecosystem?
5. In your opinion, how can we maximize the effectiveness of all the testing teams in an organization and are there specific areas where you think that significant improvements can be made?
6. In your opinion, what are the challenges to unifying the efforts of all different teams?
7. Can you explain some lessons learned from previous pentesting works?
8. In your opinion, where does the role of bug bounty programs fit in the whole security testing process?
9. What is the organization of your blue/red teams? (i.e., what other types of teams/subteams do you have in your organization?)
10. What is the workflow that your blue team would follow once an incident of a malicious attack has occurred?
11. Do you recall any bad experiences that relate to red/blue/purple teaming?
12. If red/blue/purple teaming is done correctly, how do you make sure that the vulnerabilities reported are actually fixed? And what procedures do you follow to ensure that your security posture has actually improved as a result of the testing you've done?
13. Are people normally receptive/welcoming to the type of feedback they receive from red teamers? how about other teams? If no, how do you address this problem?
14. If you were to involve a red team, would you do it in-house or outsource the red teaming process to another company? Please explain the reasons behind your answer.
15. How do you perceive each team's contribution as a whole? If you were to structure these different teams in tiers, how would you do it?
16. In your opinion, how can an organization make the feedback loop between the different testing teams faster?
17. How do bug bounty programs integrate with whatever other testing teams an organization has in place (e.g., red/blue/purple and pentesting teams)?
18. Are there redundancies between what bug hunters do and what other testing teams do?
19. Do you think that some testing teams are biased towards discovering certain vulnerability types? If yes, how so?
20. Have you ever noticed that there is misconnect between different teams? If yes, please explain why such misconnect exists; i.e., was it due to miscommunication between the teams, or was it because there is a missing team in between.
21. Do have a bug bounty program? And what motivated you to create a bug bounty program in the first place?

A.8 Questions for bug bounty hunters

1. How do you view your role in software/application vulnerability discovery in general?
2. In your opinion, when should an organization consider creating a bug bounty program?
3. Why do companies care about bug bounty programs?
4. How do bug bounty programs integrate with whatever other testing teams an organization has in place (e.g., red/blue/purple and pentesting teams)?
5. Describe some ways in which VRPs/bug bounty programs can be improved.
6. What would push you away from a bug bounty program?

A.9 Questions for internal security engineers

1. What information do you know that is unique to your team that other teams find valuable?
2. What information do you feel like would be valuable to how well you can do your job that other teams have?
3. How does your team deal with security vulnerabilities that are found in production after the software has went through your testing pipeline?
4. How does your team deal with bugs found by other teams?
5. Where do you feel your team is weak in terms of finding security vulnerabilities?
6. What do you feel like is your team's strength in terms of improving the security of the organization as a whole?
7. How are internal testing teams structured?
8. Are your paired with an agile development team or a waterfall development team?
9. Are there any tools that would help you identify security vulnerabilities during the testing pipeline that you have heard are great or wish to have?

B Codebook

Tables 1, 2, 3, and 4 show the codebook that we developed and used to analyze our interview data.

Table 1: Codebook

Code	Explanation
blue-activities	Activities carried out by blue teamers.
blue-activities-IR	Any incident response work done by blue teams.
blue-background	A blue teamer's background.
blue-definition	Definition for a blue team.
blue-issues	Problems with blue teaming like when they lack a skill or do not know something.
blue-recon	Recon/early phases in blue teaming.
blue-resources	Discussions that relate to resources/information blue teams can get access to.
blue-skill	Skills participants think are needed for blue teaming.
blue-structure	Blue team structure.
blue-tools-own	Why a blue teamer would write own/custom tools.
blue-tools-why	Why blue teamers use tools.
blue-when	When to hire a blue team.
red-activity	Activities carried out by red teamers.
red-background	A red teamer's background.
red-definition	Definition for a red team.
red-issues	problems with red teaming like when they lack a skill or do not know something.
red-process	When a participant describes the testing process red teams follow.
red-recon	recon/early phases in red teaming.
red-structure	Red team structure.
vulnerability-severity-criteria	When a participant describes how they define vul severity levels, e.g., using CVSS or any other standard.
mature-vs-immature-bountyprograms	Discussions that relate to how bounty programs are different (big vs small companies, mature vs immature ones, etc)
red-tools-own	Why a red teamer would write own/custom tools.
red-tools-why	Why red teamers use tools.
red-skill	Skills participants think are needed for red teaming.
red-specializations	Discussions that relate to what red teaming companies specialize in, e.g., social engineering, physical security, etc.
red-scoping	Any considerations that relate to defining what's in-scope or out-of-scope for red teaming engagements.
red-when	When to hire a red team.
blue-red-conflict	Conflicts between red teams and blue teams in general.

Table 2: Codebook (cont.)

Code	Explanation
red-blue-conflict-solution	How to fix the red-blue conflicts.
red-relation-other-teams	How a red team relates to other teams, including whether it is operating internally or externally.
purple-activity	Activities carried out by purple teamers.
purple-background	Purple teamers' backgrounds.
purple-definition	Definition of purple teaming.
purple-issues	Issues experienced by purple teams.
purple-structure	Purple team structure.
purple-when	When to hire a purple team.
purple-skill	Skills participants think are needed for purple teaming.
purple-process	When a participant describes the testing process purple teams follow.
red-vs-purple	Differences between red and purple teams.
bugbounty-benefits	Pros of having a bounty program.
bugbounty-communication	Communications between bounty hunters and companies (e.g., when a company invites a hacker to fix an issue or discusses fixes with them).
bountyhunters-process	Testing process as described by bounty hunters.
bountyhunters-recon	Recon methodology as described by bounty hunters.
bountyhunters-tools	Tools bounty hunters use.
bugbounty-background	A bounty hunter's background.
bugbounty-full-time-job	Discussions that relate to doing bug bounties as a full or part time job.
bugbounty-incentives	Any considerations that relate to incentivizing bounty hunters to work for a program.
bugbounty-issues	Problems with bounty programs and considerations that might deter companies from having a program.
bugbounty-legal	Legal issues associated with bounty programs (e.g., safe harbors, CFAA-related discussions).
bugbounty-private	Any discussions that relate to private (invitation-only programs) or hybrid (public and private) approaches to bug bounties.
bugbounty-rogue	Discussions that relate to any misbehaviors by bounty hunters or why companies trust/not trust bounty hunters.
bugbounty-scoping	Any considerations that relate to defining what's in-scope or out-of-scope for bug bounty programs.
bugbounty-skill	Skills participants think are needed for bug bounties/ or discussions that relate to education.
bugbounty-triaging	Triaging bug bounty reports.

Table 3: Codebook (cont.)

Code	Explanation
bugbounty-trust-issue	Programs trusting bug bounty hunters or vice versa.
bugbounty-when	When to create or (not) create a bug bounty program, and the prep work required for that.
bughunters-program-selection	Considerations hunters make when deciding what program to work for.
pentest-process	When a participant describes the testing process pen testers follow.
pentest-skill	Skills participants think are needed for pen testing.
pentest-tools-own	Why a pen tester would write own/custom tools.
pentest-tools-why	Why pen testers use tools.
pentest-when	When to hire a pen testing firm.
pentesting-definition	Definition of pen testing.
pipeline	When participants describe/justify what teams to have at a high level (e.g., QA, red and blue), or the ordering/sequence of teams to plan to have in the long run.
red-vs-pentesting	When a participant compares red teaming to pen testing.
internal-activities	Testing activities by internal teams.
internal-process	When a participant describes the testing process internal testing teams follow.
internal-team-size	Size (number of engineers) in internal teams.
internal-team-structure	When a participant describes structure of internal teams or specializations required in internal teams.
internal-tools-why	Reasons for automation by internal teams, e.g., why do they write custom scripts?, and considerations that relate to their rationale behind using certain tools.
internal-team-when	When to create an internal testing team (e.g., QA).
internal-vs-external	When a participant justifies the rationale behind hiring an internal vs an external team, or any other discussions that relate to outsourcing.
external-scoping	Discussions that relate to scoping external tests, whether with pen testing or red teaming firms.
external-same	When a participant justifies working with the same external firm or switching from one firm to another every now and then (whether it is a pen testing or a red teaming firm).
external-internal communication	Communications between internal and external testers (external consulting firms).
external-criteria	What considerations are taken when selecting an external firm (whether it is a red teaming or a pen testing firm).
upper-management	What teams expect/wish to have from top management people.
vetting-testing-firms	Discussions that relate to how to select an external red teaming or pen testing firm.
third-party-code-vulnerabilities	Discussions that relate to vulnerabilities found in 3rd party code and how to deal with them.
bugbounty-vs-pentesting	Ddiscussions that relate to comparing bug bounties and pen testing activities.
bypassing-fixes	Discussions that relate to a tester's experience bypassing security fixes.
company-area-of-business	When a participant mentions the line of business of the company/organization he/she is working in.

Table 4: Codebook (cont.)

Code	Explanation
clients-misaligned-expectations	Discussions that relate to areas of conflict or misaligned expectations between pen testers/red teamers and their clients.
compliance-vs-security	Discussions that relate to taking compliance-based approaches to security testing.
learning-resources	Any discussions that relate to resources hackers or testers use to learn security testing.
legal	Legal issues for other teams, not bug bounties.
lifecycle	Discussions that relate to SDLC and how security testing is done as a lifecycle.
NDAs	Discussions that relate to non-disclosure agreements.
organization-maturity	How organization maturity affect what teams an organization has.
process-bias	Discussions that relate to idea like hunters are biased towards web vulnerabilities, pen testers are biased towards network vulnerabilities etc.
remediation/patch-management	Fixing/patching vulnerabilities or handling vulnerability reports.
reporting-issues	Considerations concerning vulnerability reports, such as what to include in such reports.
reproducing-vulnerabilities	Any approaches a participant finds effective for making sure that others can reproduce a vulnerability that has been reported.
security-problem-budget	Issues regarding allocating budgets for security.
security-problem-communication	Communication issues within security teams in an organization.
security-problem-communication-solution	Suggestions on how to solve communication problems between teams.
security-problem-culture	General issues in organizational culture that might hinder security.
security-problems-staffing	Issues regarding staffing in security teams (e.g., not being able to find people with required expertise).
security-positive-experiences	When a participant discusses practices that they found effective for improving information sharing, communication, vulnerability discovery, etc.
security-problem-other	Security problems other than communication, staffing and budget.
security-when-reactive	Organizations' reactive approaches to security.
security-problem-solution	Ways to fix security issues in organizations.
organization-maturity	How organization maturity affect what teams an organization has.
small-companies-security	Approaches small companies use/follow to make security testing cost effective.
soft-skills	All soft and people-oriented skills mentioned in the interviews.
teams-communication	Any info sharing or communications issues participants raise during the interviews.
tools-improvement	Discussions that relate to how to improve existing tools, and what tools CISOs would like to have to improve security processes.

C Participants' Demographics

Tables 5 and 6 include detailed demographic information about our participants and whether they held engineering roles, managerial roles or both. We mark a participant as a decider if he/she held a managerial position and a do-er if he/she held a security engineering position.

Table 5: Participants' Demographics

Participant	**Job title**	**Other roles held in the past**	**Years of experience doing security testing**	**Organization size**	**Decider**	**Do-er**
P1	CISO	Programmer, manager, CISO	Less than 1 year	Large	✓	
P2	Bug bounty hunter and part-time security engineer	External pen tester, internal red teamer, bug bounty hunter	1 to 5 years	Small		✓
P3	Security engineer	Bug bounty hunter, pen tester	1 to 5 years	Medium		✓
P4	Lead infrastructure security engineer	Internal pen tester, internal red teamer, blue teamer, bug bounty hunter	1 to 5 years	Medium		✓
P5	Bug bounty hunter	Bug bounty hunter, pen tester	1 to 5 years	Small		✓
P6	Team lead penetration tester	Internal pen tester, external pen tester, internal red teamer, external red teamer, bug bounty hunter	1 to 5 years	Small		✓
P7	Full time developer and runs a pen testing firm	Internal pen tester, external pen tester, bug bounty hunter, developer, security officer	1 to 5 years	Medium	✓	✓
P8	Full time red teamer	Internal pen tester, external pen tester, internal red teamer, external red teamer, bug bounty hunter, vulnerability researcher	6 to 10 years	Small		✓
P9	Security engineer	Pen tester,blue teamer, quality assurance engineer, purple teamer	6 to 10 years	Large		✓
P10	Security manager	Blue teamer, purple teamer, bug bounty hunter, security manager	1 to 5 years	Large	✓	✓
P11	Security engineer	Quality assurance engineer, bug bounty hunter	1 to 5 years	Small		✓
P12	CISO	CISO	More than 10 years	Small	✓	
P13	Security engineer	Internal pen tester, external pen tester, internal red teamer, external red teamer, blue teamer, bug bounty hunter	More than 10 years	Large		✓
P14	Bug bounty hunter	Bug bounty hunter	1 to 5 years	NA		✓
P15	Pen tester	External pen tester, bug bounty hunter	More than 10 years	Small		✓
P16	Pen tester	Internal pen tester, bug bounty hunter	1 to 5 years	Small		✓
P17	Security engineer	Internal pen tester, external pen tester, internal/external red teamer, blue teamer, CISO	More than 10 years	Large	✓	✓
P18	Bug bounty hunter	External pen tester, internal red teamer, bug bounty hunter	1 to 5 years	NA		✓
P19	Security engineer	Internal pen tester, bug bounty hunter	6 to 10 years	Large		✓
P20	Pen tester	Quality assurance engineer, external pen tester, external red teamer	1 to 5 years	Small		✓
P21	Bug bounty hunter	Bug bounty hunter, external pen tester	1 to 5 years	NA		✓
P22	Security engineer	Security engineer, blue teamer	1 to 5 years	Medium		✓
P23	Security engineer	Internal pen tester, external pen tester, internal red teamer, bug bounty hunter	1 to 5 years	Large		✓
P24	Application security analyst	Security engineer, red teamer, pen tester, bug bounty hunter	6 to 10 years	Small		✓

Table 6: Participants' Demographics (cont.)

Participant	**Job title**	**Other roles held in the past**	**Years of experience doing security testing**	**Organization size**	**Decider**	**Do-er**
P25	Security engineer	Security engineer, bug bounty hunter	Less than 1 year	Large		✓
P26	Red teamer	Internal pen tester, red teamer	6 to 10 years	Large		✓
P27	Bug bounty hunter	Internal pen tester, bug bounty hunter	1 to 5 years	Small		✓
P28	Bug bounty hunter	Bug bounty hunter	1 to 5 years	NA		✓
P29	CTO	CTO, external pen tester	1 to 5 years	Small	✓	✓
P30	Security manager	Internal pen tester, external pen tester, internal red teamer, external red teamer, blue teamer, quality assurance engineer, security manager	More than 10 years	Large	✓	✓
P31	Security engineer	Internal pen tester, internal red teamer, blue teamer, purple teamer	1 to 5 years	Large		✓
P32	Security consultant	Pen tester, red teamer, blue teamer, purple teamer	1 to 5 years	Large		✓
P33	Security engineer	Internal pen tester	More than 10 years	Small		✓
P34	Security manager	Internal pen tester, external pen tester, external red teamer, purple teamer, CISO	More than 10 years	Small	✓	✓
P35	Security engineer	Bug bounty hunter, security engineer	1 to 5 years	Large		✓
P36	Pen tester	Internal pen tester, external pen tester, external red teamer,bug bounty hunter	Less than 1 year	Small		✓
P37	CTO	External pen tester, external red teamer, purple teamer, bug bounty hunter	More than 10 years	Small	✓	✓
P38	Director of pen testing firm	Internal pen tester, external pen tester, bug bounty hunter	More than 10 years	Small	✓	✓
P39	Blue teamer	Blue teamer	More than 10 years	Large		✓
P40	Bug bounty hunter	External pen tester, internal red teamer, external red teamer, bug bounty hunter	More than 10 years	NA		✓
P41	Application security lead	Internal pen tester, external pen tester, internal red teamer, external red teamer, purple teamer, bug bounty hunter	More than 10 years	Large		✓
P42	Technical Cybersecurity Architect	Red teamer, blue teamer, purple teamer	More than 10 years	Large		✓
P43	Security manager	Pen tester, blue teamer	More than 10 years	Large	✓	✓
P44	Security manager	Internal pen tester, blue teamer, quality assurance engineer	More than 10 years	Small	✓	✓
P45	CTO	Internal pen tester, external pen tester, internal red teamer, external red teamer, purple teamer	6 to 10 years	Small	✓	✓
P46	Blue teamer	Internal pen. tester, external pen tester, internal red teamer, blue teamer, bug bounty hunter	More than 10 years	Medium		✓
P47	Pen tester	Internal pen tester, external pen tester, internal red teamer, external red teamer	6 to 10 years	Small		✓
P48	CISO	Internal pen tester, external pen tester, blue teamer, purple teamer, CISO	More than 10 years	Medium	✓	✓
P49	CISO	CISO	Less than 1 year	Large	✓	
P50	CISO	Internal pen tester, External pen tester, Internal red teamer, external red teamer, blue teamer, purple teamer	More than 10 years	Small	✓	✓
P51	Secrity manager	Security engineer, security manager	More than 10 years	Large	✓	✓
P52	Security manager	Security manager, blue teamer	6 to 10 years	Large	✓	✓
P53	Security manager	Security manager	More than 10 years	Small	✓	