CHAPTER 20

Understanding Roles and Responsibilities

By Heather Adkins, Cyrus Vesuna, Hunter King, Felix Gröbert, and David Challoner with Susanne Landers, Steven Roddis, Sergey Simakov, Shylaja Nukala, Janet Vong, Douglas Colish, Betsy Beyer, and Paul Blankinship

This chapter addresses the question of who should work on security. We challenge the common myth that security is a topic that only experts should handle. Instead, we argue that *everyone* is responsible for security, though you may need a security specialist in some instances. We also address the role of security experts in a world where security is tightly integrated into the lifecycle of systems, and therefore handled by other types of professionals. Finally, we conclude with a look at some of the specialist options available to support an organization, especially as it grows over time.

As this book emphasizes many times, building systems is a *process*, and the processes for improving security and reliability rely on people. This means that building secure and reliable systems involves tackling two important questions:

- Who is responsible for security and reliability in the organization?
- How are security and reliability efforts integrated into the organization?

The answer to these questions is highly dependent on your organization's objectives and culture (the topic of the next chapter). The following sections lay out some high-level guidance for how to think about these questions, and offer insight into how Google has approached them over time.

On Reliability

Reliability-related roles and responsibilities are covered in other resources, so the majority of this chapter provides comparable insights on the security side. For more on the various ways in which SREs might engage with the reliability needs of an organization, and how that model may evolve over time, see Chapter 32 of the SRE book and Chapters 18, 19, and 20 of the SRE workbook.

Who Is Responsible for Security and Reliability?

Who works on security and reliability in a given organization? We believe that security and reliability should be integrated into the lifecycle of systems; therefore, they're everyone's responsibility. We'd like to challenge the myth that organizations should place the burden for these concerns solely on dedicated experts.

If reliability and security are delegated to an isolated team of people who can't mandate that other teams make security-related changes, the same failures will happen repeatedly. Their task may start to feel Sisyphean—repetitive and unproductive.

We encourage organizations to make reliability and security the responsibility of *everyone*: developers, SREs, security engineers, test engineers, tech leads, managers, project managers, tech writers, executives, and so on. That way, the nonfunctional requirements described in Chapter 4 become a focus for the whole organization throughout a system's entire lifecycle.

A System Security and Reliability Analogy

Modern cars offer a good analogy for the way security and reliability are embedded into system design and delivery. Almost every component of a car incorporates both security and reliability in some way. Seats are designed to handle a crash, a windshield needs to crack safely, and headlights are angled to avoid blinding oncoming traffic. Seat belts must withstand being latched thousands of times. The windshield has to repel all types of weather, and the headlights must always turn on when you need them. A car's digital systems must be similarly hardened. Everyone who plays a part in building the car must do this work—not just safety and reliability experts.

The Roles of Specialists

If everyone is responsible for security and reliability, then you might wonder: what exactly is the role of a security specialist or reliability expert? According to one school of thought, the engineers building a given system should primarily focus on its core functionality. For example, developers might focus on building a set of critical user

journeys for a mobile phone–based app. Complementing the work of the developer team, a security-focused engineer will look at the app from the perspective of an attacker aiming to undermine its safety. A reliability-focused engineer can help understand the dependency chains and, based on these, identify what metrics should be measured that will lead to happy customers and an SLA-compliant system. This division of labor is common in many development environments, but it's important that these types of roles work together rather than in isolation.

To expand on this idea further, depending on the complexity of a system, an organization may need people with specialized experience to make nuanced judgment calls. Since it's not possible to build an absolutely secure system that's resilient against every attack, or a system that is perfectly reliable, advice from experts can help steer development teams. Ideally, this guidance should be integrated into the development lifecycle. This integration can take multiple forms, and security and reliability specialists should work directly with developers or other specialists that consult at each stage of the lifecycle to improve systems.[1] For example, security consultation can happen in multiple stages:

- A *security design review* at the outset of a project to determine how security is integrated
- *Ongoing security audits* to make sure a product is built correctly per security specifications
- *Testing* to see what vulnerabilities an independent person can find

Security and Reliability Risk Evaluation

Nuanced security and reliability advice can be helpful in making judgment calls about risk. For example, suppose developers working on a project haven't had time to build in a desired security protection or think about graceful degradation, but need to launch the product or system soon. A security engineer and an SRE can help the organization understand what might happen if it's launched in its current state. Will adversaries be able to attack a vulnerable system? Will the global system go down if user traffic for a certain country is higher than expected? What is the likelihood that either of these events will happen? Is the organization equipped with temporary mitigations for potential issues that might arise? Experienced security engineers and SREs can offer valuable advice in these situations.

1 The development arc described in Chapter 18 of the SRE workbook demonstrates the value that an experienced SRE offers throughout the entire product lifecycle.

Security experts should be responsible for implementing security-specific technologies that require specialist knowledge. Cryptography is the canonical example: "don't roll your own crypto" is a common industry catchphrase meant to discourage enterprising developers from implementing their own solutions. Cryptography implementations, whether in libraries or hardware, should be left to experts. If your organization needs to provide secure services (such as a web service over HTTPS), use industry-accepted and verified solutions instead of attempting to write your own encryption algorithm. Specialist security knowledge can also be required to implement other types of highly complex security infrastructure, such as custom authentication, authorization, and auditing (AAA) systems, or new secure frameworks to prevent common security vulnerabilities.

Reliability engineers (such as SREs) are best positioned to develop centralized infrastructure and organization-wide automation. Chapter 7 of the SRE book discusses the value and evolution of horizontal solutions, and shows how critical software that enables product development and launches can evolve into a platform.

Finally, specialists in security and reliability can devise best practices, policies, and training tailored to your organization's workflows. These tools should empower developers to adopt best practices and implement effective security and reliability practices. A specialist should aim to build a brain trust of knowledge for the organization by constantly educating themselves on developments in the industry and generating broader awareness (see "Culture of Awareness" on page 476). In creating awareness, a specialist can help the organization become more secure and reliable in an iterative way. For example, Google has SRE- and security-focused educational programs that provide a baseline level of knowledge to all new hires in these specific roles. In addition to making the course material available company-wide, we also offer employees many self-study courses on these topics.

Understanding Security Expertise

Anyone who has tried to hire security professionals into their organization knows that the task can be challenging. If you're not a security specialist yourself, what should you look for when hiring one? Medical professionals provide a good analogy: most have a general understanding of the fundamentals of human health, but many specialize at some point. In the medical field, family doctors or general practitioners are typically responsible for primary care, but more serious conditions may call for a specialist in neurology, cardiology, or some other area. Similarly, all security professionals tend to command a general body of knowledge, but they also tend to specialize in a few specific areas.

Before you hire a security specialist, it's important to know the types of skills your organization will need. If your organization is small—for example, if you're a startup or an open source project—a generalist may cover many of your needs. As your

organization grows and matures, its security challenges may become more complex and require increased specialization. Table 20-1 presents some key milestones in Google's early history that needed corresponding security expertise.

Table 20-1. Security expertise needed at key milestones in Google's history

Company milestones	Expertise needed	Security challenges
Google Search (1998) *Google Search provides users with the ability to find publicly available information.*	General	Search query log data protection Denial-of-service protection Network and system security
Google AdWords (2000) *Google AdWords (Google Ads) enables advertisers to show ads on Google Search and other products.*	General Data security Network security Systems security Application security Compliance and audit Anti-fraud Privacy Denial of service Insider risk	Financial data protection Regulatory compliance Complex web applications Identity Account abuse Fraud and insider abuse
Blogger (2003) *Blogger is a platform that allows users to host their own web pages.*	General Data security Network security Systems security Application security Content abuse Denial of service	Denial of service Platform abuse Complex web applications
Google Mail (Gmail) (2004) *Gmail is Google's free webmail system, with advanced features available via a paid GSuite account.*	General Privacy Data security Network security Systems security Application security Cryptography Anti-spam Anti-abuse Incident response Insider risk Enterprise security	Protecting highly sensitive user content at rest and in transit Threat models involving highly capable external attackers Complex web applications Identity systems Account abuse Email spam and abuse Denial of service Insider abuse Enterprise needs

Certifications and Academia

Some security experts seek to earn certifications in their field of interest. Security-focused industry certifications are offered by institutions worldwide, and can be good indicators of someone's interest in developing relevant skills for their career and their ability to learn key concepts. These certifications typically involve a standardized knowledge-based test. Some certifications require a minimum amount of classroom,

conference, or job experience. Nearly all expire after a certain amount of time, or require certificants to refresh minimum requirements.

These standardized testing mechanisms may not necessarily attest to someone's aptitude for success in a security role at your organization, so we recommend taking a balanced approach to assessing security specialists, considering all of their qualifications in totality: their practical experience, certifications, and personal interest. While certifications may speak to someone's ability to pass exams, we have seen credentialed professionals who've had difficulty applying their knowledge to solving problems. At the same time, early career candidates, or those coming to the field from other specialist roles, may use certifications to upgrade their knowledge quickly. With a keen interest in the field, or practical experience with open source projects (instead of workplace experience), such early career candidates may be able to add value quickly.

Because security experts are increasingly in demand, many industries and universities have been developing and evolving security-focused academic programs. Some institutions offer general security-focused degrees that cover many security domains. Other degree programs concentrate on a specific security domain (which is common for doctoral students), and some offer a blended curriculum that focuses on the overlap between cybersecurity issues and domains such as public policy, law, and privacy. As with certifications, we recommend considering a candidate's academic achievements in the context of their practical experience and your organization's needs.

For example, you might want to bring on an experienced professional as your first security hire, and then hire early career talent once the team is established and can offer mentorship. Alternatively, if your organization is working on a niche technical problem (such as securing self-driving cars), a new PhD graduate with deep knowledge in that specific research area but little work experience might fit the role nicely.

Integrating Security into the Organization

Knowing when to start working on security is more of an art than a science. Opinions on this topic are plentiful and varied. However, it's generally safe to say that the sooner you start thinking about security, the better off you'll be. In more concrete terms, we've observed certain conditions over the years that are likely to trigger organizations (including our own) to start building a security program:

- When an organization begins to handle data of a personal nature, such as logs of sensitive user activity, financial information, health records, or email
- When an organization needs to build highly secure environments or custom technologies, such as custom security features in a web browser

- When regulations require adherence to a standard (such as Sarbanes-Oxley, PCI DSS, or GDPR) or a related audit[2]
- When an organization has contractual requirements with customers, especially around breach notification or minimum security standards
- During or after a compromise or data breach (ideally, before)
- As a reaction to the compromise of a peer operating in the same industry

In general, you'll want to start working on security far before any of these conditions are met, and especially before a data breach! It's far simpler to implement security before, rather than after, such an event. For example, if your company plans to launch a new product that accepts online payments, you may want to consider a specialty vendor for that functionality. Vetting a vendor and ensuring that they have good data handling practices will take time.

Imagine that you launch with a vendor that doesn't integrate the online payment system securely. A data breach could incur regulatory fines, loss of customer trust, and a hit to productivity as your engineers reimplement the system correctly. Many organizations cease to exist after such incidents.[3]

Similarly, imagine that your company is signing a new contract with a partner that has additional data handling requirements. Hypothetically, your legal team may advise you to implement those requirements before signing the contract. What might happen if you delay those extra protections and suffer a breach as a result?

Related questions often arise when considering the cost of a security program and the resources your company can invest in the program: how expensive is implementing security, and can the company afford it? While this chapter can't cover this very complex topic deeply, we'll emphasize two main takeaways.

First off, for security to be effective, it must be carefully balanced with your organization's other requirements. To put this guideline in perspective, we can make Google nearly 100% safe from malicious actors by turning off our datacenters, networks, computing devices, and so on. While doing so would achieve a high level of safety, Google would no longer have customers and would disappear into the annals of failed companies. Availability is a core tenet of security! In order to craft a reasonable

2 The Sarbanes-Oxley Act of 2002 (the Public Company Accounting Reform and Investor Protection Act) sets standards for public US companies regarding their accounting practices, and includes information security topics. The Payment Card Industry Data Security Standard sets minimum guidelines around protecting credit card information; compliance is required for anyone doing payment processing of this kind. The General Data Protection Regulation is an EU regulation concerned with the handling of personal data.

3 A few notable cases of organizations that went out of business or filed for bankruptcy after breaches are Code Spaces and the American Medical Collection Agency.

security strategy, you need to understand what the business requires to operate (and in the case of most companies, what it takes to earn a profit). Find the right balance between the requirements of your business and adequate security controls.

Secondly, security is everyone's responsibility. You can reduce the cost of some security processes by distributing it among the teams affected the most. For example, consider a company that has six products, each staffed with a product team and protected by 20 firewalls. In this scenario, one common approach is to have a central security team maintain the configuration of all 120 firewalls. This setup requires the security team to have extensive knowledge of six different products—a recipe for eventual reliability issues or delays in system changes, all of which can increase the cost of your security program. An alternative approach is to assign responsibility to the security team for operating an automated configuration system that accepts, validates, approves, and pushes firewall changes proposed by the six product teams. This way, each product team can efficiently propose minor changes for review and scale the configuration process. These kinds of optimizations can save time and even improve system reliability by catching errors early without human involvement.

Because security is such an integral part of an organization's lifecycle, nontechnical areas of the organization also need to consider security early on. For example, boards of directors often examine the security and privacy practices of the entities they oversee. Lawsuits in the aftermath of data breaches, such as the shareholder suit against Yahoo! in 2017, are driving this trend.[4] When preparing your roadmap for security, be sure to consider these types of stakeholders in your process.

Finally, it's important to create processes for maintaining a constant understanding of the current issues you need to address, along with their priorities. When treated as a continuous process, security requires an ongoing assessment of the risks the business is facing. In order to iterate defense-in-depth security controls over time, you need to incorporate risk assessment into your software development lifecycle and security practices. The next section discusses some practical strategies for doing so.

Embedding Security Specialists and Security Teams

Over the years, we've seen many companies experiment with where to embed security specialists and security teams inside their organizations. The configurations have ranged from fully embedded security specialists inside product teams (see Chapter 19) to fully centralized security teams. Google's central security team is organizationally configured as a hybrid of both options.

4 In the US, executives and boards of directors are increasingly being held accountable for security in their organizations. The Concord Law School at Purdue University has written a good article on this trend.

Many companies also have different accountability arrangements for decision making. We've seen Chief Information Security Officers (CISOs) and other leadership roles responsible for security report to just about every C-level executive: the CEO, CFO, CIO, or COO, the general counsel for the company, a VP of Engineering, and even the CSO (Chief Security Officer, usually responsible also for physical security). There is no right or wrong configuration, and the choice your organization makes will be highly dependent on what's most effective for your security efforts.

The rest of this section offers some details on configuration options that we've had success with over the years. While we're a big technology company, many of these components also work well in small or medium-sized organizations. However, we imagine this configuration may not work well for a financial company or a public utility, where accountability for security may have different stakeholders and drivers. Your mileage may vary.

Example: Embedding Security at Google

At Google, we first built out a central security organization that operates as a peer to product engineering. The head of this organization is a senior leader within engineering (a VP). This creates a reporting structure in which security is seen as an engineering ally, but also allows the security team sufficient independence to raise issues and resolve disputes without conflicts of interest other leaders may have. This is similar to the way the SRE organization at Google maintains separate reporting chains from product development teams.[5] In this way, we create an open and transparent engagement model that focuses on improvements. Otherwise, you risk having a team with the following characteristics:

- Unable to raise serious issues because launches are overprioritized
- Seen as a blocking gate that needs to be circumvented organizationally via silent launches
- Slowed down by insufficient documentation or code access

Google's central security team relies on standard processes like ticketing systems to interact with the rest of the organization when teams need to request a design review, an access control change, and so on. For a sense of how this workflow functions, see "Google's Smart System for Intake" on page 464.

As Google has grown, it has also become useful to embed a "security champion" within individual product engineering peer groups. The security champion becomes the gateway to facilitate collaboration between the central security team and the product team. When starting out, this role is ideal for senior engineers with good

5 See Chapter 31 in the SRE book.

standing in the organization and an interest or a background in security. These engineers also become the technical leads for product security initiatives. As product teams become more complex, this role is assigned to a senior decider, such as a director or VP—this person can make tough calls (such as balancing launches versus security fixes), acquire resources, and resolve conflicts.

In the security champion model, it's important to establish and agree upon an engagement process and responsibilities. For example, the central team may continue to perform design reviews and audits, set organization-wide security policies and standards, build safe application frameworks (see Chapter 13), and devise common infrastructure such as least privilege methods (see Chapter 5). Distributed security champions are key stakeholders for these activities, and should help decide how these controls will work in their product teams. The security champions also drive the implementation of policies, frameworks, infrastructure, and methods within their respective product teams. This organizational configuration requires a tight communication loop through team charters, cross-functional meetings, mailing lists, chat channels, and so on.

Because of Google and Alphabet's large size, in addition to a central security team and distributed security champions, we also have special decentralized security teams for more complex products. For example, the Android security team sits within the Android engineering organization. Chrome has a similar model (see Chapter 19). This means the Android and Chrome security teams are responsible for the end-to-end security of their respective products, which includes deciding on product-specific standards, frameworks, infrastructure, and methods. These specialized security teams run the product security review process and have special programs to harden the products. For example, the Android security team has worked to harden the media stack and has benefited from an integrated security and engineering approach.

In all of these models, it's important for the security team to be open and approachable. In addition to a security review process, during which developers can receive help from subject matter experts, engineers need timely and consistent feedback on security-related issues throughout the project lifecycle. We address a number of cultural issues around these interactions in Chapter 21.

Google's Smart System for Intake

In many organizations, a ticket queue is the only communication channel to the security team. The "one size fits most" nature of tickets typically results in a lot of back and forth in order to extract relevant information. To help our teams save time, we built a smart system as our ticket queue frontend to automate away as much of the consulting process as possible.

Instead of providing a simple form or forms as input into our queue, we built a system with a dynamic questionnaire. As the user describes their request, the system

automatically asks common security-related questions, and provides warnings and recommendations to educate them about risky decisions. These guiding questions help the user determine whether they are using a memory-safe language, applying a vetted templating system/framework, handling sensitive data, or modifying a critical system. After the user fills out the form, the system is able to identify an explicit problem and create a ticket to automatically route to the correct user or team. Then a security engineer can quickly parse the inherently structured information and relevant data, and help the user.

Since security work constantly evolves and the questionnaire won't cover all use cases, the intake form allows users to bypass sections and choose an "other" option if their request doesn't directly map to a defined workflow. To prevent users from defaulting to the "other" option, we explicitly say that this option is for one-off requests that are time-sensitive.

One key feature of the expert system is its ability to evolve and grow with the organization. If a large number of users skip our questionnaire, we know our expert system needs tweaking. Security engineers periodically examine the sections users most often bypass, and either add new question paths or modify overly burdensome question segments. The goal is to encourage users to focus on the main security questions they need to think about.

Building this system helped us accomplish the following:

- Let the user self-serve, and self-educate themselves in the process.
- Save time for meaningful and productive collaboration in providing a recommendation.
- Dramatically improve the overall speed of understanding the problem and providing a recommendation.
- Greatly increase the overall number of tickets successfully closed per week.
- Improve the quality of the information in tickets.

Special Teams: Blue and Red Teams

Security teams are often tagged using colors to denote their role in securing an organization.[6] All of these color-coded teams work toward the common goal of improving the security posture of the company.

Blue Teams are primarily responsible for assessing and hardening software and infrastructure. They're also responsible for detection, containment, and recovery in the event of a successful attack. Blue Team members can be anyone in an organization

6 This color scheme is derived from the US military.

who works on defending it, including the people who build secure and reliable systems.

Red Teams run offensive security exercises: end-to-end attacks that simulate realistic adversaries. These exercises reveal weaknesses in an organization's defenses and test its ability to detect and defend against attacks.

Typically, Red Teams focus on the following:

A specific goal
: For example, a Red Team might seek to exploit customer account data (or more specifically, to find and exfiltrate to a safe destination some customer account data that is available in your environment). Such exercises are very similar to the way adversaries operate.

Surveillance
: The aim is to determine whether your detection methods can detect reconnaissance by an adversary. Surveillance can also serve as a map for future goal-based engagements.

Targeted attacks
: The aim is to demonstrate the feasibility of exploiting security issues that are supposedly theoretical and very unlikely to be exploited. As a result, you can determine which issues merit building a defense.

Before starting a Red Team program, be sure to obtain buy-in from parts of the organization that might be affected by these exercises, including legal and executives. This is also a good time to define boundaries—for example, Red Teams should not access customer data or disrupt production services, and they should use approximations for data theft and service outages (e.g., by compromising only the data of test accounts). These boundaries need to strike a balance between conducting a realistic exercise and establishing a timing and scope that your partner teams are comfortable with. Of course, your adversaries won't respect these boundaries, so Red Teams should pay extra attention to key areas that are not well protected.

Some Red Teams share their attack plans with the Blue Team, and work very closely with them to get fast and comprehensive insight into the detection situation. This relationship can even be formalized with a Purple Team that bridges the two.[7] This can be useful if you are conducting many exercises and want to move fast, or if you want to distribute Red Team activity among product engineers. This configuration can also inspire the Red Team to look in places it might not otherwise consider. The

7 For more on Purple Teams, see Brotherston, Lee, and Amanda Berlin. 2017. *Defensive Security Handbook: Best Practices for Securing Infrastructure*. Sebastopol, CA: O'Reilly Media.

engineers that design, implement, and maintain systems know the system best, and usually have an instinct for where the weaknesses are.

Detecting Red Teams

If your Red and Blue Teams choose not to share information with each other, be sure to establish a protocol for what to do when the Blue Team detects the Red Team. Imagine this scenario: a Red Team successfully breaches your customer database. The Blue Team detects them and executes emergency response procedures, resulting in notifications to your executives, legal team, and regulators! A good protocol for deescalation after detection will prevent this sort of confusion.

Red Teams are not vulnerability scanning or penetration testing teams. *Vulnerability scanning teams* look for predictable and known weaknesses in software and configurations that can be automatically scanned for. *Penetration testing teams* focus more on finding a large set of vulnerabilities and the testers trying to exploit them. Their scope is narrower, focused on a particular product, infrastructure component, or process. As these teams mostly test prevention aspects and some detection aspects of an organization's security defense, their typical engagement lasts days.

In contrast, Red Team engagements are goal-oriented and typically last weeks. Their goals are specific targets, such as intellectual property or customer data exfiltration. They are broadly scoped and use any means necessary to attain their goals (within safety limits) by traversing product, infrastructure, and internal/external boundaries.

Given time, good Red Teams can attain their goals, often without being detected. Rather than viewing a successful Red Team attack as a judgment of a poor or ineffective business unit, use this information to better understand some of your more complex systems in a blameless way.[8] Use Red Team exercises as an opportunity to better learn how these systems are interconnected and how they share trust boundaries. Red Teams are designed to help bolster threat models and build defenses.

8 Do so by building on a culture of blameless postmortems, as described in Chapter 15 of the SRE book.

Because they don't exactly mirror the behavior of external attackers, Red Team attacks aren't a perfect test of your detection and response capabilities. This is especially true if the Red Team is staffed by internal engineers who have existing knowledge about the systems they're attempting to penetrate.

You also can't feasibly conduct Red Team attacks frequently enough to provide a real-time view of your vulnerability to attacks or statistically significant metrics for your detection and response teams. Red Teams are meant to find the rare edge cases that normal testing cannot. All caveats aside, regularly conducting Red Team exercises is a good way to understand your security posture end to end.

You can also leverage Red Teams to teach the people who design, implement, and maintain systems about the adversarial mindset. Embedding these people directly into the attack team—for example, via a small-scoped project—will give them first-hand insight into how attackers scrutinize a system for possible vulnerabilities and work around defenses. They can inject this knowledge into their team's development process later on.

Engaging with a Red Team helps you better understand your organization's security posture and develop a roadmap for implementing meaningful risk reduction projects. By understanding the implications of your current risk tolerance, you can determine whether you need to make adjustments.

External Researchers

Another way to examine and improve your security posture is to work closely with outside researchers and enthusiasts who find vulnerabilities in your systems. As we mentioned in Chapter 2, this can be a useful way to get feedback about your systems.

Many companies work with outside researchers by establishing *Vulnerability Reward Programs (VRPs)*, also colloquially referred to as *bug bounty programs*. These programs offer rewards in exchange for responsibly disclosing vulnerabilities about your system, which may or may not come in cash form.[9] Google's first VRP, started in 2006, offered a T-shirt and a simple thank you message on our public-facing web page. Through reward programs, you can expand the hunt for security-related bugs outside of your immediate organization and engage with a larger number of security researchers.

Before starting a VRP, it's a good idea to first cover the basics of finding and addressing regular security issues that thorough reviews and basic vulnerability scanning can

9 See Google researcher sirdarckcat's blog post on rewards for a more philosophical outlook.

find. Otherwise, you end up paying external people to find bugs that your own teams could have easily detected. This is not the intended purpose of VRPs. It also has the downside that more than one researcher may report the same issue to you.

Knowing how to set up a bug bounty program requires a little bit of legwork up front. If you choose to run a bug bounty program, you can follow these basic steps:

1. Determine whether your organization is ready for this program.
   a. Scope the areas of your system to target. For example, you might not be able to target corporate systems.
   b. Determine payout levels and set aside funds for payouts.[10]
2. Consider whether you want to run an in-house bug bounty program or hire an organization that specializes in these programs.
3. If running your own, set up a process for bug intake, triage, investigation, validation, follow-up, and fixes. In our experience, you can estimate this process to take approximately 40 hours for each serious issue, excluding fixes.
4. Define a process for making payments. Remember that reporters may be located all over the world, not just in your home country. You will need to work with your legal and finance teams to understand any constraints that may exist on your organization.
5. Launch, learn, and iterate.

Every bug bounty program faces some likely challenges, including the following:

The need to fine-tune the firehose of issues being reported
: Depending on your industry reputation, the attack surface, payout amounts, and the ease of finding bugs, you may be fielding an overwhelming number of reports. Understand up front what level of response will be required from your organization.

Poor report quality
: A bug bounty program can become burdensome if most of your engineers are chasing down basic issues or nonissues. We have found this is especially true for web services, since many users have misconfigured browsers and "find" bugs that aren't actually bugs. Security researchers are less likely to be in this pool, but sometimes it's hard to discern a bug reporter's qualifications up front.

10 For further reading, see sirdarckcat's post about vulnerability pricing.

Language barriers
: A vulnerability researcher may not necessarily report a bug to you in your native language. Tools for language translation can be helpful here, or your organization may have someone who understands the language used by the reporter.

Vulnerability disclosure guidelines
: The rules for disclosing vulnerabilities are not generally agreed upon. Should the researcher go public with what they know, and if so, when? How long should the researcher give your organization to fix the bug? What types of findings will be rewarded, and what types won't? There are many differing opinions about the "right" methods to use here. Here are some suggestions for further reading:

  - The Google security team has written a blog post on responsible disclosure.[11]
  - Project Zero, an internal vulnerability research team at Google, has also written a blog post on data-driven updates to disclosure policy.
  - The Oulu University Secure Programming Group provides a useful collection of vulnerability disclosure publications.
  - The International Standards Organization (ISO) provides recommendations for vulnerability disclosure.

Be prepared to address issues researchers report to you in a timely manner. Also be aware that they may find issues that have been actively exploited by a malicious actor for some period of time, in which case you may also have a security breach to address.

Conclusion

Security and reliability are created by the quality or absence of processes and practices. People are the most important drivers of these processes and practices. Effective employees are able to collaborate across roles, departments, and cultural boundaries. We live in a world where the future is unknowable and our adversaries are unpredictable. At the end of the day, ensuring that everyone in your organization is responsible for security and reliability is the best defense!

11 sirdarckcat has also written a post about vulnerability disclosure.