HANDBOOK

The Security Leader’s Handbook

Contents

Part 1: The First 90 Days

Part 2: You’re Settled in, Now What?

Part 3: Maintaining a Successful Security Program

Appendix

About HackerOne

PART 1

The First 90 Days

You've been told you're the next director of security for your organization. Your role is to start and scale security practices at a high-growth company.

What happens next? Where do you get started?

Your company may have basic security practices in place, such as scanning tools and annual penetration tests. But what is the best foundation for security at scale? What steps will build a strong security program that grows along with your company?

Take our plan and use it to start making progress from day one.

Chapter 1: Getting Your Bearings

Congratulations! You've been made the Director of Security (or similar) of your organization. Taking over such a crucial role can be daunting, and there's probably a lot swirling around your head—at the very least, new and existing projects, upgrades, vulnerabilities, and developing a plan for the coming years.

Veteran security leaders we've talked to suggest starting your tenure by getting your bearings. Take the time to understand what the current state of your organization is and where it needs to go. This will take time, but it's crucial to prepare you to lead your security efforts for years to come.

Understand the Business Reason for Your Appointment

Getting your bearings begins with understanding the circumstances surrounding your appointment. This doesn't mean your professional credentials. What purpose does your position serve within the broader organization?

Our experience working with security leaders at start-ups, Fortune 500 companies like GM, and everything in between reveals 3 primary reasons why organizations appoint a new security leader:

1. Growth

As organizations grow, their security footprint also grows. New startups need security, but are mainly focused on delivering functionality quickly to ensure a product is viable. As they gain users the need for security increases, and the potential harm from a vulnerability is more significant.

To address this, the organization needs robust practices and policies to secure applications, infrastructure, networks, and cloud assets. Achieving this will require cooperation between the security function and other areas of the business—for example, those responsible for IT infrastructure, networking, software development, etc. You'll be the liaison between security and these other functions, so your job will be part captain, part ambassador.

2. Acquiring another company

When a company is acquired, its systems must be thoroughly assessed and integrated safely with yours. Its applications and infrastructure may have vulnerabilities to be managed. It's now your responsibility to properly assess and remediate vulnerabilities and other security issues in newly acquired assets, and safely integrate them into your existing IT environment.

3. Expansion into a new region

When conducting business in new regions you're exposed to new laws and regulations, e.g., GDPR. You must understand any new laws or standards that apply to your organization and build a plan to reach and maintain compliance.

Understanding the reasons behind your appointment plays a large part in deciding where to start and will help you begin to develop a sensible plan based on business-focused objectives. From there, you can move on to understanding what security looks like right now within your organization.

Understand the Baseline of Today's Security

One of the first things you'll need to do as Director of Security is create a baseline for security at your organization. It's important to understand the current state before deciding what the future should be.

Speak to leaders and managers from IT, legal, finance, HR, marketing/PR, and security. They can tell you what threats they see presenting a risk to the organization. This will help you understand the risks your organization faces so you can properly assess them and decide which to tackle first. While it is tempting to focus purely on technological issues, you must develop an understanding of other areas, e.g., business, personnel, financial, operational, reputational, and regulatory risks.

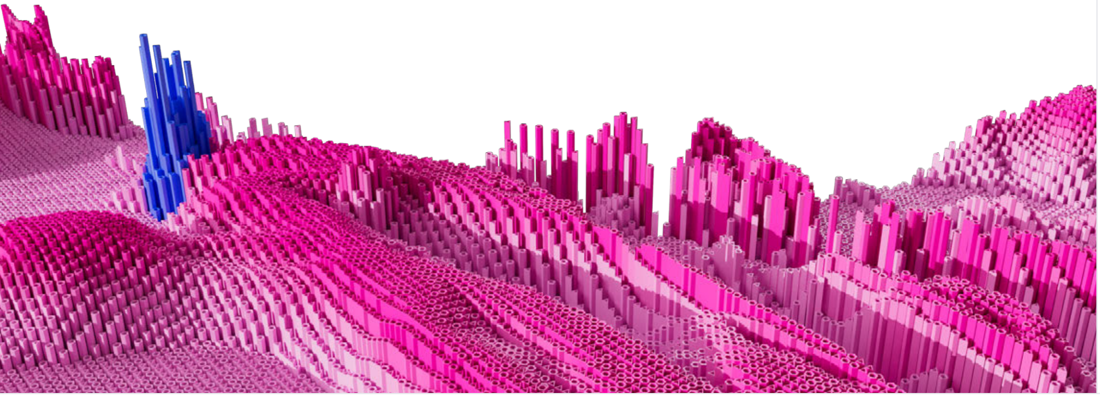
Get a feel for recent incidents or major issues experienced by your organization. These could point to systemic flaws that need to be remediated immediately.

Engaging with other teams provides two benefits

1. You'll begin to understand how your organization handles risk. Your decisions will be based on a thorough understanding of your organization and its various challenges and obligations.

2. You'll position yourself as a partner, not an adversary. You're not there to place blame but to help reduce risk and make the organization safer for employees and customers.

Once you have your bearings and an understanding of the current state of risk management, you're prepared to set the foundation for security moving forward.



Ask each department about the current state of risk management. Are risks being identified and handled properly? How long does it take for risks to be mitigated? Are there any process improvements that would make risk management more efficient for them?

To Be a Business Leader, Understand the Business

In an interview with CIO Talk Network, Rob Hornbuckle, CISO of Allegiant, explained:

"Once you have a full comprehension of the business, you can talk to [other departments] not just as a security person but as a fellow officer, about how their pieces affect the organization. Being a security leader brings you to the leadership table, but what you do when you get there is dependent on your capabilities, your knowledge and business understanding, and how well you establish those trusted relationships."

Rob Hornbuckle
CISO, Allegiant

Chapter 2: Setting the Foundation for Security

At this point, you've met with several departments to understand the risks facing your organization. You've created a picture of current risk management practices. This is important for a good foundation.

But don't forget that security cannot happen without other departments onboard. For example, the relationship between security and developers is foundational to effective application security.

Development teams must understand you're there to help them find solutions. You aren't the "code cops" coming to break down their door. You're not trying to get in the way of delivering functionality. Instead, you're there to help them to find solutions to fundamental security problems. You're working with them—in an equal partnership—to create solutions that meet the needs of all stakeholders.

Approach relationships carefully—take time to assess their health, and determine any changes required to strengthen them. Coming from a place of mutual trust lays a solid foundation. It may take time if your organization has a history of conflict between security and other departments, but it's worth the effort.

Understand Security from a User's Standpoint

A sound security foundation must include an understanding of the end user's perspective—whether that's an internal employee or a paying customer.

Good hackers get familiar with a product and try to put themselves in the position of a user. They try to break the system by doing unexpected things. They ask themselves: "How can a user make the system do something it shouldn't?" You should do the same.

Start by walking through the employee onboarding process from start to finish. Pay attention to the processes used to create accounts, provision hardware, and create passwords. Don't search for vulnerabilities, but pay attention to inefficiencies or glaring problem areas.

After an internal look, walk through a customer's experience of your products. Start with marketing and move through account access, entering credit card information, or however your process flow takes people from awareness through to becoming a paying customer.

Understanding these flows through the eyes of an end user can inform improvements to your security strategy moving forward. **You can't secure a system you don't understand**, and taking the time to build that understanding now will pay dividends later.

Understanding Threats To Your Systems

The National Institute of Standards and Technology (NIST) defines threat modeling as:

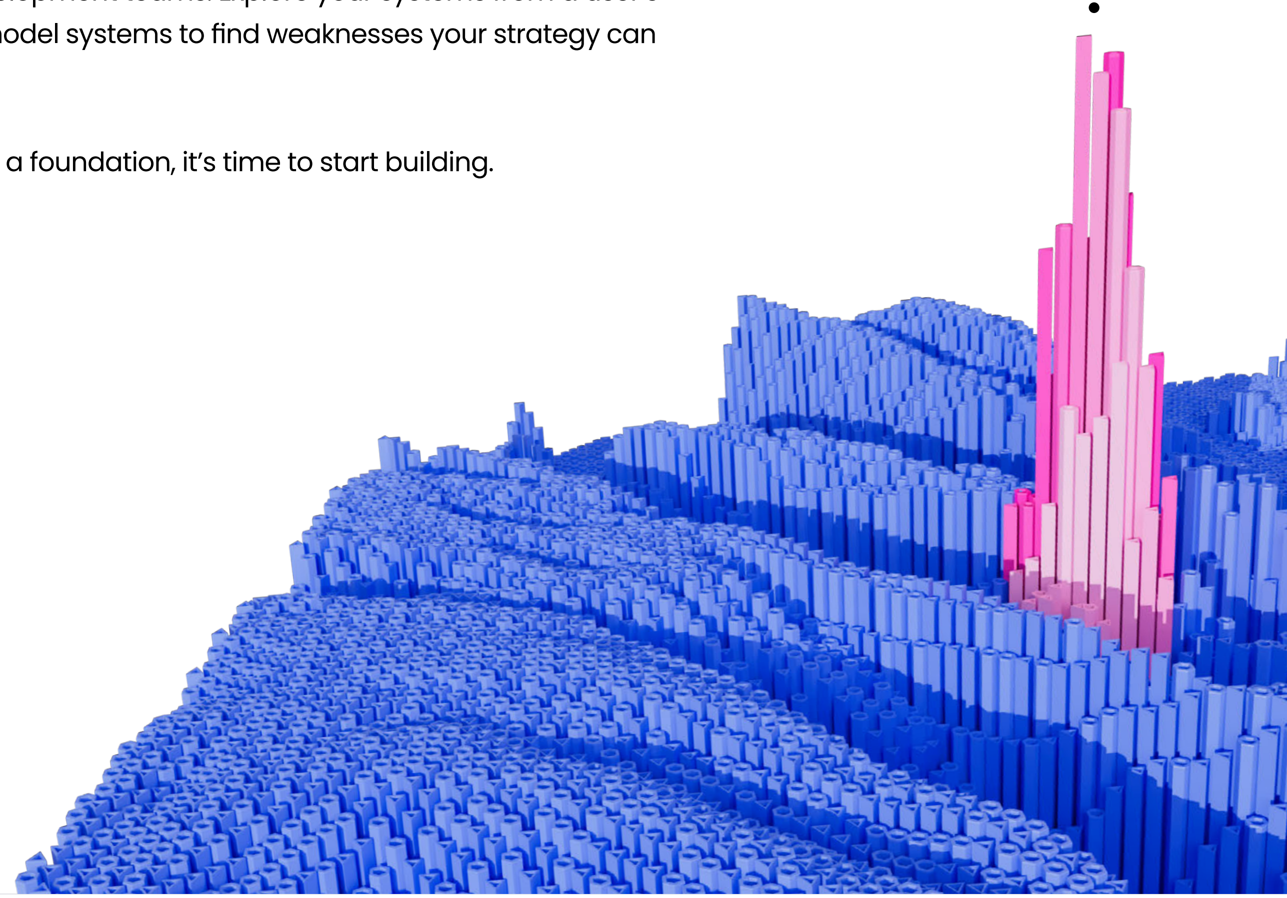
“A form of risk assessment that models aspects of the attack and defense sides of a logical entity, such as a piece of data, an application, a host, a system, or an environment.”

Put more simply, it’s a structured process of reviewing a system or asset to identify potential threats, attacks, and vulnerabilities that could affect its operation. Once uncovered, threats are recorded, risk assessed, and prioritized for remediation. When this process occurs early enough in the development life cycle, you’ll be able to fix major problems before they are released to production—and long before an external attacker can exploit them.

The results of a threat modeling exercise can change your plan. Use them to focus your security budget and penetration testing efforts. Setting a secure foundation helps you remain stable as you begin to implement your security strategy.

Build trust with development teams. Explore your systems from a user’s viewpoint. Threat model systems to find weaknesses your strategy can help resolve.

Once you have laid a foundation, it’s time to start building.



Chapter 3: Prioritize and Execute

At this point, you've identified:

1. What projects are currently in progress.
2. Which new projects are needed to improve security.
3. Systemic and specific issues to be resolved (based on threat modeling results).

Your task now is to decide what to do first. There is no silver bullet to effective planning—but there are ways to break your plan up into stages and arrange them to maximize benefit to the organization.

Congratulations! You've made it through your first 90 days. Here's what you've accomplished:

- You've established a baseline of the risks and threats facing your organization.
- You've learned your systems inside and out, from both a technical and user-focused perspective.
- You've decided what projects are most important to complete, and you're ready to execute.

Leading a security organization is more than planning. You must now decide on a mode of operation. How will you operate your business day-to-day? What principles will guide your decisions?

Prioritization Checklist

□ Understand the reasons for existing practices. Are they needed? Is that need based on data? You may need to shut down practices that are no longer important or performing poorly.

□ Identify new projects/practices needed to address identified threats and risks, support upcoming business initiatives, or respond to new external demands (e.g., regulatory).

□ Identify projects that are immediately necessary for the organization and prioritize them according to importance and available resources.

□ Break the upcoming year into quarters and plan which projects will fall where. For the first quarter, pick one or two projects that must be done to address immediate business needs.

□ For each following quarter, continue to plan in projects according to their importance and available resources.

Planning in this manner ensures your team can focus on those projects that will best support the organization, e.g., by reducing unacceptable risks, addressing imminent threats, or supporting business initiatives. Remember to focus on 1-2 projects at a time rather than spreading resources thinly across multiple projects—this will enable you to score faster "wins" and demonstrate value.

PART 2

You’re Settled in, Now What?

You’ve got your bearings, built strategic relationships with other departments, and understood what you need to do next. Now comes the fun part: executing your plan.

Let’s look at how to manage the business of security on a daily basis.

Chapter 4: Don’t Eliminate Risk, Manage it

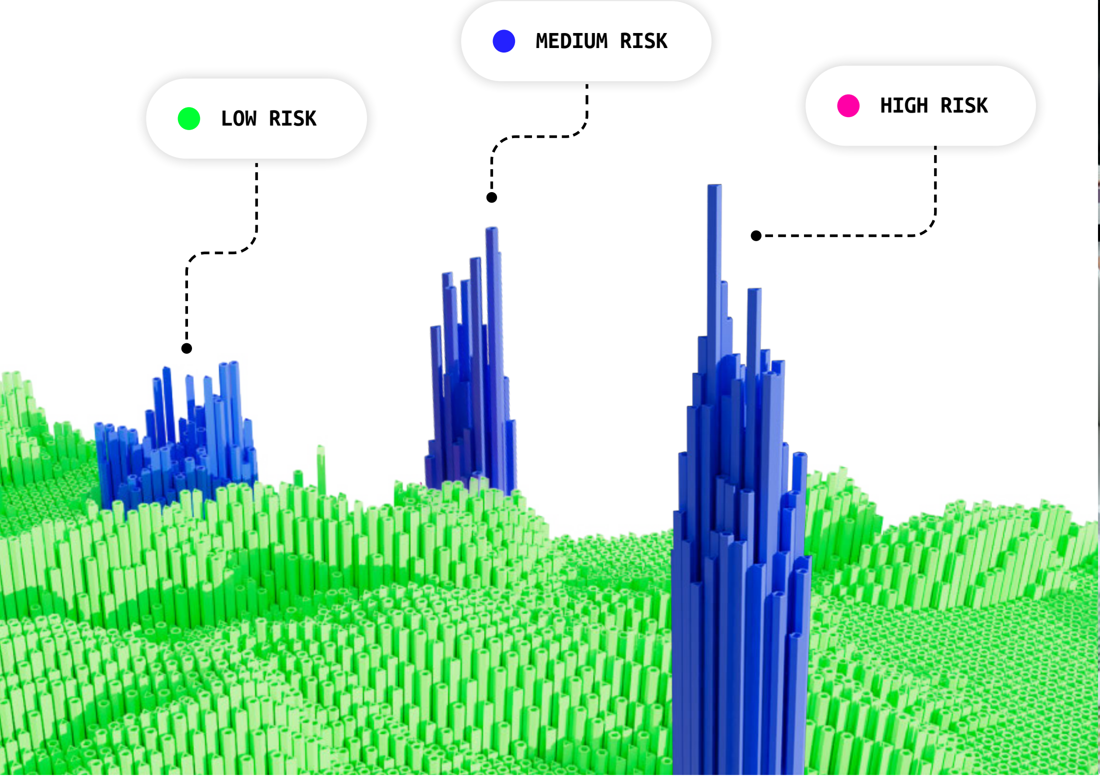
Although we’ve talked about security projects, there’s an important point to understand. Planning projects based on identified threats can have a detrimental effect. A project has a beginning and an end—once it’s finished, you move on to the next project. Risk management doesn’t work that way.

In most cases, you can’t eliminate risks. You can only manage them.

Risk management begins with understanding your assets—where they are, what they’re worth, and who owns them. A vulnerability in an internal MS Access database used by 20 employees is not as important as a vulnerability in a publicly exposed API.

Any identified risk should go through a cost/benefit analysis. The possible exposure is compared against the cost to mitigate the risk. If a risk could cost the organization $10,000 but requires a $50,000 investment to fix, it is a low priority. It simply doesn’t make sense to fix it—at least for now. Risks like this will be accepted by the organization and recorded for reassessment at a later date.

Risk ratings are a necessary part of any risk management strategy. But what rating system works best to help determine priority?



Simple rating systems help the relationship between security and development teams. Developers need to understand what needs to be fixed, and what doesn't. Introducing intermediary levels (e.g., 'Medium,' or 'High') or scales from 1-10 can help, but only as long as there are clear guidelines for what constitutes a NOW problem vs. something that can be scheduled for later. Without this, vulnerabilities classified as (e.g.,) 'Medium' are likely to sit in limbo because it's unclear how important they are.

Risks rated highly should undergo root cause analysis to find out what led to such a critical risk and how to prevent the same thing happening again. Over time, emergency, "drop everything" vulnerabilities should begin to disappear as secure development and automated testing practices become the norm.

Keep Risk Rating Systems Simple

Andrew Dunbar, VP of Security Engineering and IT at Shopify, stated his preference for simplicity in a webinar on **effective application security testing**. He makes determining potential exposure "as simple as it can be to achieve the right outcomes." Don't over-complicate your risk rating system to the point where ratings don't mean anything. Dunbar explains that Shopify has two ratings, high or low. High ratings need to be fixed right away. Low risks can be fixed during the next sprint or two.

Find a rating system that works for your organization, being sure to remove ambiguity over what does and doesn't need to be addressed immediately—and what happens to lower severity vulnerabilities.

shopify

Patch Management

The code you write might be bug-free, but what about the code your code depends on?

Dependencies are inevitable when building complex software systems. Your business operations likely depend on a mixture of in-house applications, open source components, and third-party vendor applications. All require good patch management to stay secure.

The code you write will often depend on open source libraries and frameworks. When vulnerabilities are found within these frameworks, all applications that use them are also vulnerable. It's essential to update libraries and frameworks quickly so your applications aren't exploitable.

Similarly, your organization can also be placed at risk by vulnerabilities in third-party vendor products. These products typically undergo the same testing as yours, and occasionally vulnerabilities will be found that put your users at risk. Patch management processes ensure patches released by third-party vendors don't sit for months before being applied in your environment.

Once a vulnerability is announced to the public, attackers will try to exploit it. This is true of vulnerabilities in dependencies and third party software—the source isn't important, it's the impact you should be concerned about. The longer you wait to patch your systems, the longer attackers have to find your application and take advantage.

Patch management processes don't have to be complicated. Whenever possible, work within the confines of your developers' daily workflow.

For example, if an open source library needs to be patched, create a pull request within the repo of the application so all developers will see it. Explain in the pull request what you're updating and that all they have to do is merge and their job is done. You'll not only help your application stay secure, but you'll also generate goodwill with development teams.

Chapter 5: The Wide World of Security Assessments

Security assessments are a necessary part of a security leader's role. In smaller organizations, you may directly meet with outside partners for these assessments. In larger organizations, there may be a team in charge of handling these relationships, but you'll still have to sign off on the process and results.

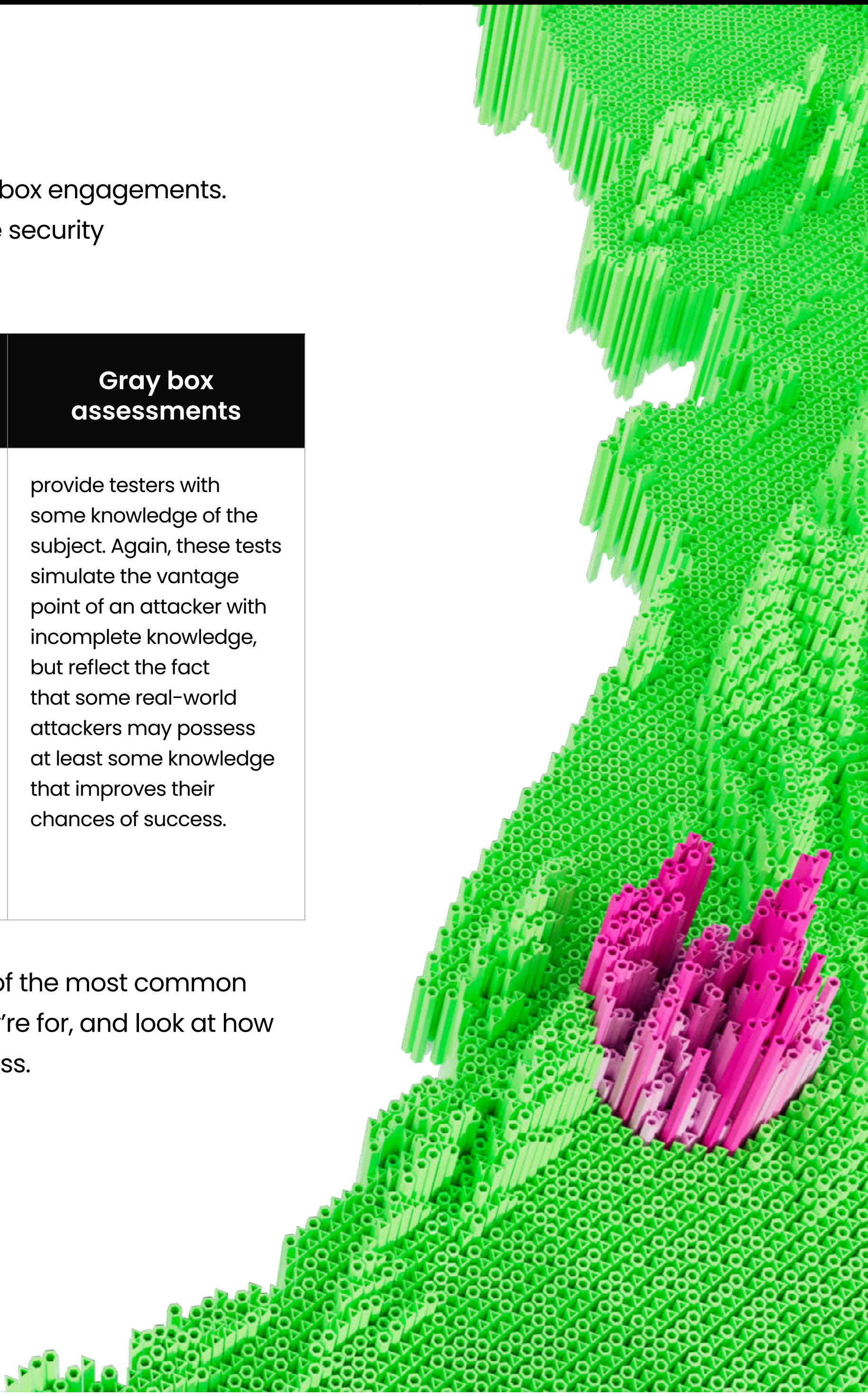
Before jumping into scheduling security assessments for your applications, take the time to understand the specific purpose of each type of assessment.

Black-Box and White-Box Testing

Security assessments can be white box or black box engagements. Understanding the difference will help you to use security assessments effectively.

White box assessments	Black box assessments	Gray box assessments
allow testers to see into the inner workings of an application, system, or IT environment. The tester has access to source code, system diagrams, internal logic, relevant documentation, etc., allowing them to uncover a broad range of technical issues.	require testers to work with no knowledge of how the subject works. These tests simulate the vantage point of an attacker who has to learn by exploring, and are designed to illustrate how a real-world attacker may be able to successfully compromise a system.	provide testers with some knowledge of the subject. Again, these tests simulate the vantage point of an attacker with incomplete knowledge, but reflect the fact that some real-world attackers may possess at least some knowledge that improves their chances of success.

With that out of the way, we'll now look at some of the most common types of security assessments, explain what they're for, and look at how they can be combined for maximum effectiveness.



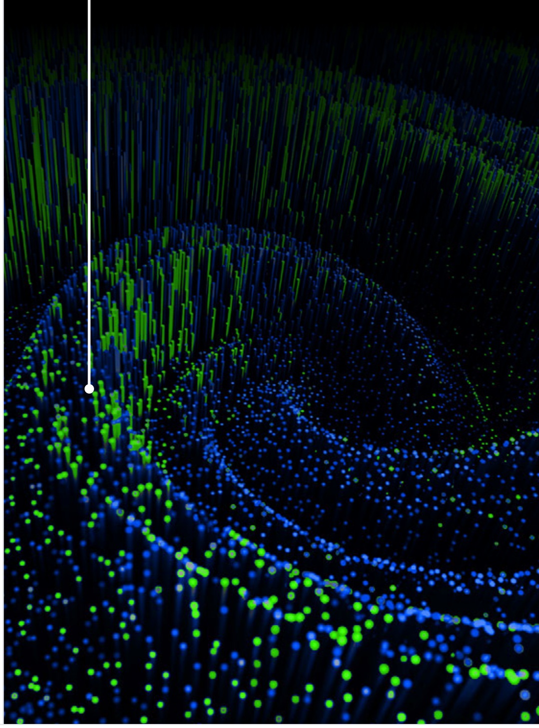
Security Assessments

Security assessment is a catch-all term for any human-led testing engagement that aims to uncover vulnerabilities in an IT environment or asset. A security assessment can have a wide range of different objectives—anything from providing broad, low-level testing coverage across a wide attack surface (e.g., a cloud environment) to highly focused and intensive testing of a specific product element (e.g., an API).

Historically, the most common form of security assessment was a traditional penetration test. These engagements were mainly used to satisfy compliance requirements (e.g., for PCI-DSS) and to provide intensive testing prior to the release of a new product, version, or feature. While these engagements had their place, they often delayed IT initiatives due to scheduling difficulties and slow results reporting.

Today, security assessments are far more flexible, with many security providers offering bespoke testing engagements to meet the customer's specific needs. The introduction of ethical hackers into the security assessment ecosystem has yielded even more flexibility, enabling organizations to schedule assessments more quickly and easily, and even leverage continuous testing programs.

Red teams continually test applications and other IT assets to find vulnerabilities.



Red Team Services

A red team is a permanent team used to improve the security posture of a company. Red teams are not a one-time assessment. They continually test applications and other IT assets to find vulnerabilities. Red teams focus on real-world tactics, and are made up of highly trained and experienced professionals who think like attackers.

Red teams serve a unique purpose. They aren't just finding vulnerabilities and reporting them—they exist to make corporate defenders (A.K.A. the 'blue team') better. The result is an improved ability to detect and contain attacks as they happen, instead of finding evidence long after the damage is done.

Audit

Audits are necessary for most organizations, particularly those in highly-regulated industries. An audit is not a true security assessment, as it is not designed to uncover vulnerabilities. It measures how well your systems adhere to a standard, e.g., PCI-DSS, NIST 800-53, or HIPAA.

A standard may state that you must use strong encryption to protect customer data. It may have a list of acceptable encryption algorithms, and you may use any of them. That's as far as the audit goes. On the other hand, a penetration test could discover that you store encryption keys alongside the data in the database, allowing an attacker to easily decrypt the data.

You can be compliant with one or more standards and be insecure at the same time. Audits don't verify security beyond what's required to satisfy the requirements of a specific framework. This is an important distinction.

Organizations with good security practices are very likely to be compliant. But **compliance, while necessary, should never be confused with security.**

Juggling numerous security assessments, penetration tests, and audits is no easy feat—you can do it when you understand the purpose behind each assessment and when to use it.

Chapter 6: Securing the Shifting Sands

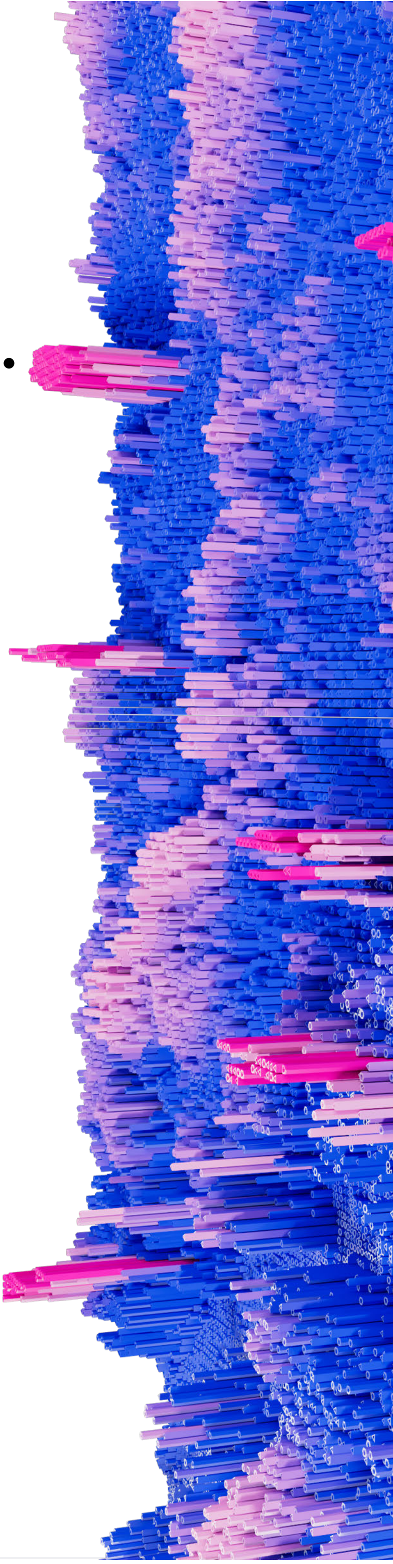
Any experienced security leader will tell you that continuous learning is a requirement of the job. The ever-changing risk landscape exposes your company to new and dangerous risks every day. Now we'll look at some general principles that will help you keep ahead of these risks.

Keep an Eye on Third Parties and Vendors

Third parties are a requirement in today's connected world. There's nothing wrong with using third parties to perform tasks your business wasn't created to do. Why create your own Human Resources system or shopping cart functionality if that's not what brings you revenue? Using third parties, however, creates risk. There are three ways risk increases when using third parties:

1. The risk of data being misused by a third party.
2. The risk of a third party's poor security practices leaking your data without your knowledge.
3. The increased attack surface if the third party application contains vulnerabilities.

In today's environment, you're only as secure as your **weakest vendor**. Vet your vendors carefully and make sure you're comfortable with their security policies before signing a contract with them. Once you hand over your data, you're placing your organization at their mercy.



Your Attack Surface Changes

Companies are increasingly adopting DevOps practices, which encourage development teams to put code out quickly and get fast feedback wherever possible.

Don't sacrifice security in exchange for "better, faster, cheaper." Fast-moving development environments increase the risk of services and applications being deployed without the security team's knowledge. This is especially true of cloud environments, where developers can create virtual machines and containers in seconds and deploy code to them at any time.

Strong policies are needed so administrators know what they are allowed to do. Policies preventing the abuse of cloud resources are a good idea, but you have to enforce them. AWS, Azure, and Google Cloud aren't going to police your admins and developers for you.

The good news is automation can be used to help enforce policies. For example, AWS Lambda can be used to **scan file uploads to S3 buckets** in AWS. Policies can prevent developers from creating new virtual machines using their accounts. A good rule of thumb is to use a build pipeline to build any infrastructure your applications need, preventing humans from deciding how to build and deploy VMs and containers and minimizing opportunities for human error.

And it's not just about architecture. The security team should be made aware of all new assets—hardware and software—and what their purpose is.

This can be daunting in a microservice environment but is absolutely necessary. You must know when new services are created and released, understand what cloud service accounts exist, and have a clear process to create new ones so they can be monitored. You can't be purely reactive—proactively look at what is currently running in your environment and investigate anything unexpected.

The key to maintaining the security of your systems is to reduce the element of surprise. Build processes that enable security teams to stay up-to-date with new assets. These assets must be tested and deployed according to well enforced policies. Hold your vendors to the same standards you'd hold yourself—don't trust your data with anyone until you have verified their suitability.

Verifying that third parties are adequately protecting your data while staying in control of your changing environment can be a challenge. However, successfully tackling this challenge is possible and enables a mature security program that will be invaluable as your company grows.

● PART 3

Maintaining a Successful Security Program

As Director of Security, it's your responsibility to create an environment that encourages security, making the day-to-day measures we discussed in Part 2 easier.

How do you create programs that deliver security at DevOps speeds? How do you stay ahead of coding errors that can cause significant harm? Do you share what you've learned or keep it secret?

Chapter 7: The Art of Continuous Security

“Continuous security” is a strange phrase. Nothing is 100% secure. No one silver bullet can keep all systems everywhere impenetrable at all times. But that’s not the goal of continuous security.

Continuous security is a defined process that allows you to know what is happening in your environment and react to it quickly. It uses smart automation to make security the default. It makes security an intrinsic part of applications without stopping development teams from delivering quickly.

The use of the word “art” in the chapter title is deliberate. Security in a DevOps environment is often more an art than a science. There are concrete aspects, such as metrics to measure test coverage or policies to prevent rogue servers or buckets.

But how much test coverage is enough? 70%? 80%? And who should have authority to create servers—all administrators or a select few?

These are decisions that have to be made. You can get all the advice you like, but the final decision is yours. You make it and you have to live with it.

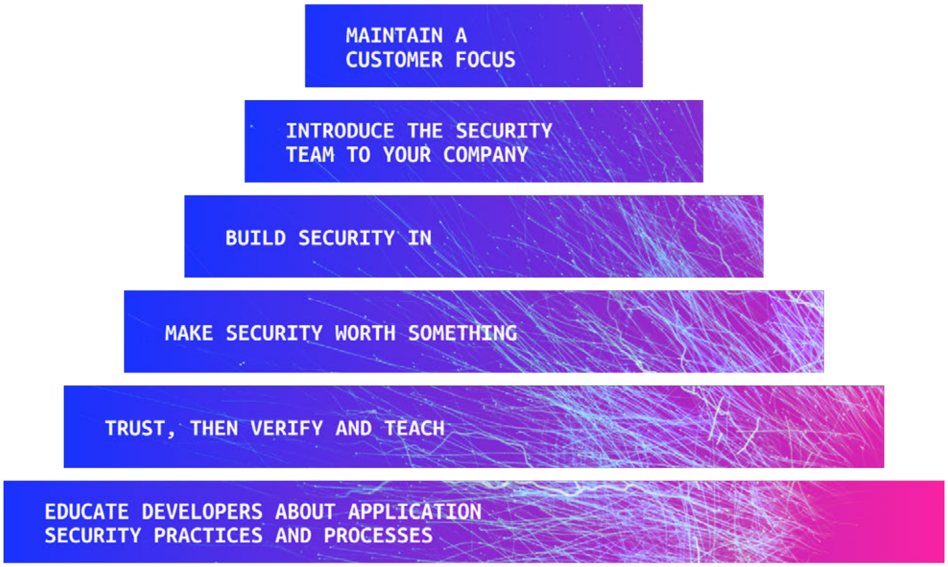
The best guideline to use is your customers. Your goal should be to build software your customers will trust—what will that take? Often, vanity metrics or minimum thresholds only deliver minimum security. Being trustworthy takes much more than just meeting the minimum required standard.

Build a Culture of Security

Culture is the operating environment of an organization, including its values, mission, and attitude and those of its employees. Security has often been a background process, like scanning for vulnerabilities or performing a vulnerability assessment before deploying to production. That's not enough for continuous security.

Achieving a Culture of Security

Let's look at some key steps to building a culture of security



MAINTAIN A CUSTOMER FOCUS

INTRODUCE THE SECURITY TEAM TO YOUR COMPANY

BUILD SECURITY IN

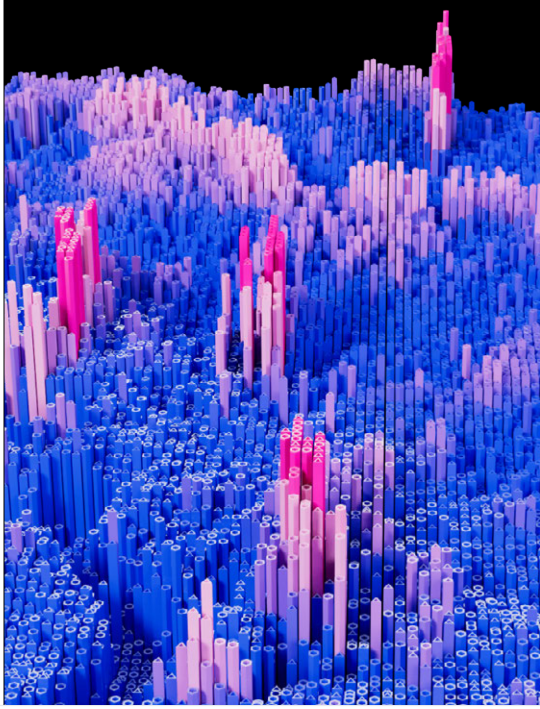
MAKE SECURITY WORTH SOMETHING

TRUST, THEN VERIFY AND TEACH

EDUCATE DEVELOPERS ABOUT APPLICATION SECURITY PRACTICES AND PROCESSES

Your developers should understand application security principles and be trained on what security processes exist and why. Allow them to spend time with the security team, learning what to look for and how applications look through a security practitioner's eyes. Similarly, the security team should spend time with developers so they can understand how poor security practices obstruct the DevOps pipeline—and avoid building security practices with those characteristics.

Give developers the freedom to experiment. Trust that they want to do the right thing, then verify. When mistakes happen—they will—solve them without placing blame, and then fix your practices and systems so the same mistake can't happen again.



An excellent way to show trust while educating developers was discussed in a BSides San Francisco conference talk by Arkadiy Tetelman. His favorite way to build a security culture is to send out "fireside bug bounty" write-ups to developers and other interested parties when a legitimate vulnerability is found via bug bounty. Each communication outlines what the vulnerability was, how to exploit it, who fixed it and how, and how to eliminate that entire class of vulnerabilities across the system.

Fireside bug bounties encourage developers to think about security and be helpful when security teams make requests for fixes or updates.

Make security worth something. Give $200 to the developer who reports a strange VM running in the cloud, or fixes a nasty vulnerability without the security team having to ask. Reward the marketing employee that reports a phishing email, even when no tests are running.

Make security easy by building it in as much as possible. Common security features such as authentication and authorization should be built into reusable development frameworks. Build servers with automated scripts based on a known secure template. A developer should have to work hard to build an insecure system, because everything is set up to ensure the security of code and IT assets.

Introduce the entire company to what your security team does and why it's important. Fun events like security expos are a chance to demonstrate what attackers can do if they succeed in breaching the company. Show a day in the life of a security engineer or incident response engineer. Tell the security team's story, entertain your visitors. If it's memorable, you'll have less friction when you need to introduce new policies or standards, because other functions will already understand the stakes.

Above all, make your customer the focus. Build your culture around delivering the best service to your customers—not just keeping the lights on, but **becoming trustworthy stewards of their data.**

Building a culture takes time, but is well worth the effort. Security has become everyone's job. Make it easy. Make it fun. Make it worth something.

So far, we've focused on planning and directing internal resources. Today, many security leaders recognize the value of opening certain security activities to external experts. In addition to the assessments described earlier, bug bounties have grown increasingly popular in recent years.

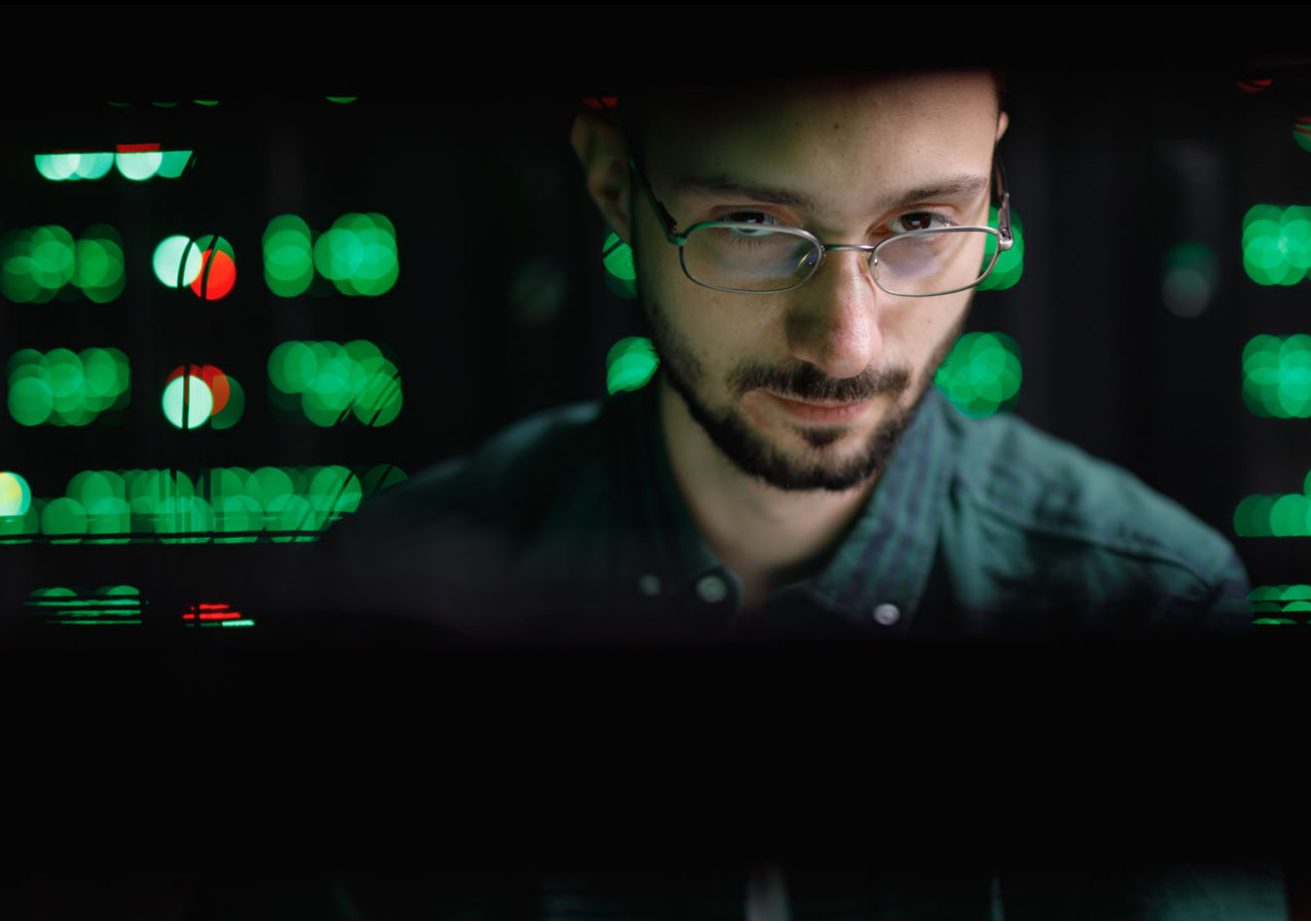
When should your organization consider setting up a bug bounty program? And how can you integrate it with existing security activities? Let's take a look.

Chapter 8: Bug Bounties: When and How?

A bug bounty program takes advantage of the global hacker community to uncover vulnerabilities you don't have the resources or skills to find internally. Hackers submit bugs and are rewarded by you based on their severity and impact.

Bug bounty programs provide access to more security researchers than you could ever afford to hire, allowing you to scale your security testing program more easily—and with a much greater scope—than hiring. This can be a crucial leg-up for fast-growing organizations, where security teams are typically a fraction of the size of development teams, and struggle to keep up with the growth of code bases.

While not a silver bullet, bug bounty is a strong addition to the security program at your company. A bug bounty program can help you achieve continuous security by providing continuous testing by thousands of security researchers with all manner of specialisms and testing skills.



How To Build Toward a Bug Bounty Program

Like any large initiative, it's best to start small with bug bounty and scale up your program as you learn to manage and mature it. There are concrete steps to quickly take you from bounty newbie to master.

A vulnerability disclosure policy (VDP) is the first step you should take before creating a bug bounty program. A VDP formalizes the process of submitting vulnerabilities by providing guidelines on what assets hackers can test, how to submit vulnerabilities, and how you will handle submissions. Your VDP should include language promising not to prosecute hackers as long as they stick to the guidelines.

VDPs are essential to establish guidelines and make hackers feel comfortable submitting findings. Once a VDP is in place, you'll need a mechanism to accept vulnerabilities—typically an email address using the form: **"security@<companyname>.com."**

What Security Leaders Say About Bug Bounties

"Coinbase loves bug bounties. Bounties de-criminalize the actions of good-faith security researchers while still forbidding malicious hacking. Bounties help grow the next generation of security talent. We love bounties so much that we've decided to expand our bounty payouts."

Philip Martin,
Director of Security
Coinbase

"We see the bug bounty program as a permanent part of our security program. To lose this program would be an enormous setback. There's no question it's a runaway success. There isn't a single person in the company who thinks the bounty program was not successful."

Neil Matatall,
Security Engineer
Github

5 Critical Components in Every VDP

Promise
Demonstrate a clear, good faith commitment to customers and other stakeholders potentially impacted by security vulnerabilities.

Scope
Indicate which assets, products, and vulnerability types are covered.

“Safe Harbor”
Assures that reporters of good faith will not be unduly penalized.

Process
The steps a hacker should use to report vulnerabilities.

Preferences
A living document that sets expectations on how reports will be evaluated.

Once your VDP is established, there’s another intermediary step you can take to prepare your teams and stakeholders ahead of launching a public bug bounty program.

Hacker-powered assessments and penetration tests are a great way to begin your journey working with ethical hackers. These engagements are time-bound, and enlist the support of a small group of hackers to achieve specific goals—for example, security testing a new product ahead of launch or meeting specific compliance requirements.

Once your teams are accustomed to working alongside expert hackers via your VDP and hacker-powered testing engagements, you’re ready to expand.

> “Our HackerOne bug bounty program is one part of our enhanced security strategy. It helps us identify systemic issues, which we can then work to resolve. This can mean anything from taking a different approach to building our technology to having more robust review processes.”

Joe Xavier
VP of Engineering
Grammarly

The next step is implementing a private bug bounty program. Instead of opening your program to the entire hacker community, you personally invite a smaller number of hackers to participate.

Private bug bounty programs provide a manageable number of vulnerabilities, helping you uncover and resolve issues in your processes and prepare your teams to communicate effectively with hackers. These programs measurably improve the security of your assets and IT infrastructure without placing undue stress on your personnel or processes while they are still adapting to this new way of working.

Once you have submission, triage, and remediation processes working like a well-oiled machine, you can open your program up to the wider community.

Public bug bounty programs produce more vulnerability reports because they maximize your access to the global hacker community's unparalleled range of testing skills. A public program also provides you with excellent PR opportunities by demonstrating your commitment to continuous security.

Bug bounty programs have helped many organizations supplement and scale their security teams in ways they never thought possible. Take the time to evaluate your current processes and see where a bug bounty program may fit.

Beyond Bug Bounty

Finding vulnerabilities is only half the battle. The other is keeping track of assets. Most security teams struggle to understand and monitor their entire attack surface—and this is particularly true at fast-growing organizations. This is another area where the global hacker community can help. While searching for vulnerabilities, hackers frequently come across previously-unknown digital assets. Since these assets aren't known to the security team, they are often insecure and pose a significant risk. By reporting these assets, hackers can help your organization stay on top of its attack surface, substantially reducing cyber risk. To find out more, visit our **Attack Surface Management** page.

Chapter 9: Share Your Knowledge

Security has long been associated with secrecy. The mantra has been: if no one knows how something works, they can't attack it. If you share your protective strategies, you're opening the organization to attack.

However, the security industry is changing. Secrecy is no longer the name of the game. Instead, openness and collaboration between security professionals is becoming the norm. When security leaders share their knowledge, it helps the entire security community become better.

The Internet doesn't have to maintain an "everyone for themselves" attitude. The Internet becomes safer as a whole when leaders work together and share information. If your security program develops into a world-class operation, share your learning points with the wider security community. If every organization has a world-class security team, all of us will be that much safer.

Keep track of what you've learned, and look for opportunities to share it with others. Professional organizations such as ISC2 and SANS exist to help share security knowledge. Become an instructor or contribute research reports and case studies.

Conferences also present a great opportunity to share knowledge. Apply to speak at a security conference so you can share what you've learned in a relaxed setting with hundreds of people. If they apply what you've taught them, you've had a positive impact on the security industry as a whole.

The top security leaders lead not only their organization but others as well. Collaboration and sharing is the path to a more secure future. Take advantage of opportunities to teach and learn from others. Together, we'll make the Internet a safer place for everyone.

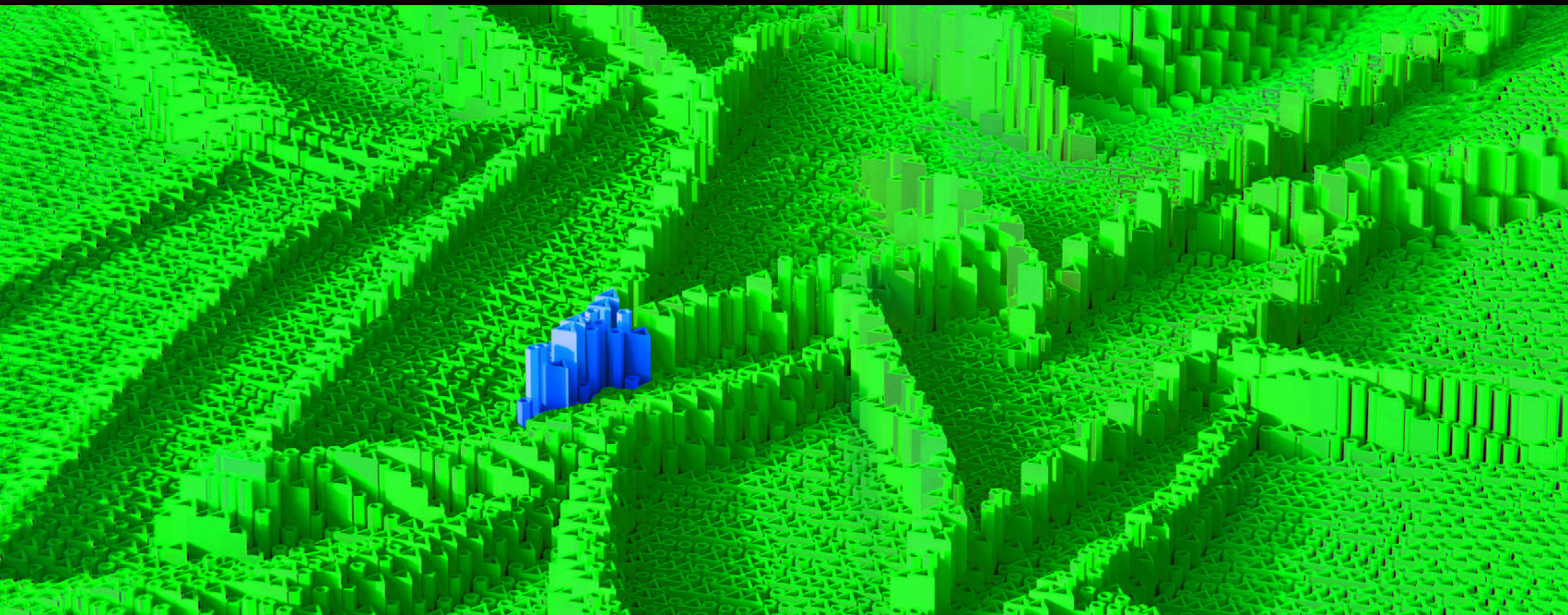
Appendix

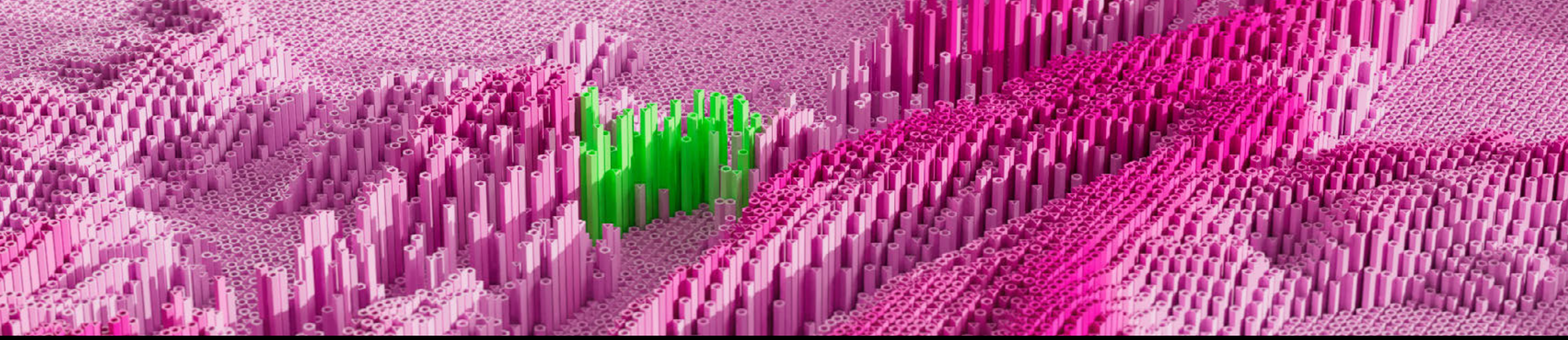
Application Security Checklist

- □ Determine your organization’s current baseline state of security
- □ Understand security from a user’s standpoint
- □ Threat model your applications and infrastructure to identify risks
- □ Don’t try to eliminate risk, manage it
- □ Establish clear patch management procedures
- □ Use security assessments to find vulnerabilities from different angles
    - □ Threat assessments to review the entire system for vulnerabilities
    - □ Penetration tests to test one specific piece of your system once hardened
    - □ Create a red team to constantly attack your applications with the mind of an attacker
- □ Create security assessment policies for third-parties and rigorously vet your vendors
- □ Create processes to monitor newly created services and applications
- □ Monitor AWS (and other cloud) accounts and assets for rogue VMs and services
- □ Create “trustworthy” thresholds for test coverage and other metrics that create software worthy of customers’ trust
- □ Promote a culture of security:
    - □ Teach application security practices and processes to developers and other teams
    - □ Trust their intentions, then verify their actions and use mistakes as teaching opportunities
    - □ Make security worth something within the organization
    - □ Build security in wherever possible
    - □ Introduce the company to your security team
    - □ Maintain a customer focus
- □ Build toward a bug bounty program:
    - □ Create a VDP
    - □ Perform a hacker-powered penetration test
    - □ Create a private bug bounty
    - □ Create a public bug bounty program
- □ Share your knowledge through professional organizations and conferences

About HackerOne

HackerOne pinpoints the most critical security flaws across an organization’s attack surface with continual adversarial testing to outmatch cybercriminals. HackerOne’s Attack Resistance Platform blends the security expertise of ethical hackers with asset discovery, continuous assessment, and process enhancement to reduce threat exposure and empower organizations to transform their businesses with confidence. Customers include Citrix, Coinbase, Costa Coffee, General Motors, GitHub, Goldman Sachs, Google, Hyatt, Microsoft, PayPal, Singapore’s Ministry of Defense, Slack, the U.S. Department of Defense, and Yahoo. In 2021, HackerOne was named a ‘brand that matters’ by Fast Company.





HackerOne has vetted hackers for organizations including:

gm
Lufthansa
ZEBRA
zoom
Spotify
citrix
PayPal
Uber
HYATT
U.S. Department of Defense
Google
reddit
Nintendo
Adobe
A.S. Watson Group
sumo logic
yahoo!
priceline
shopify
slack
yelp
salesforce
TOYOTA

hackerone

With over 2,000 customer programs, more companies trust HackerOne than any other vendor

Contact Us