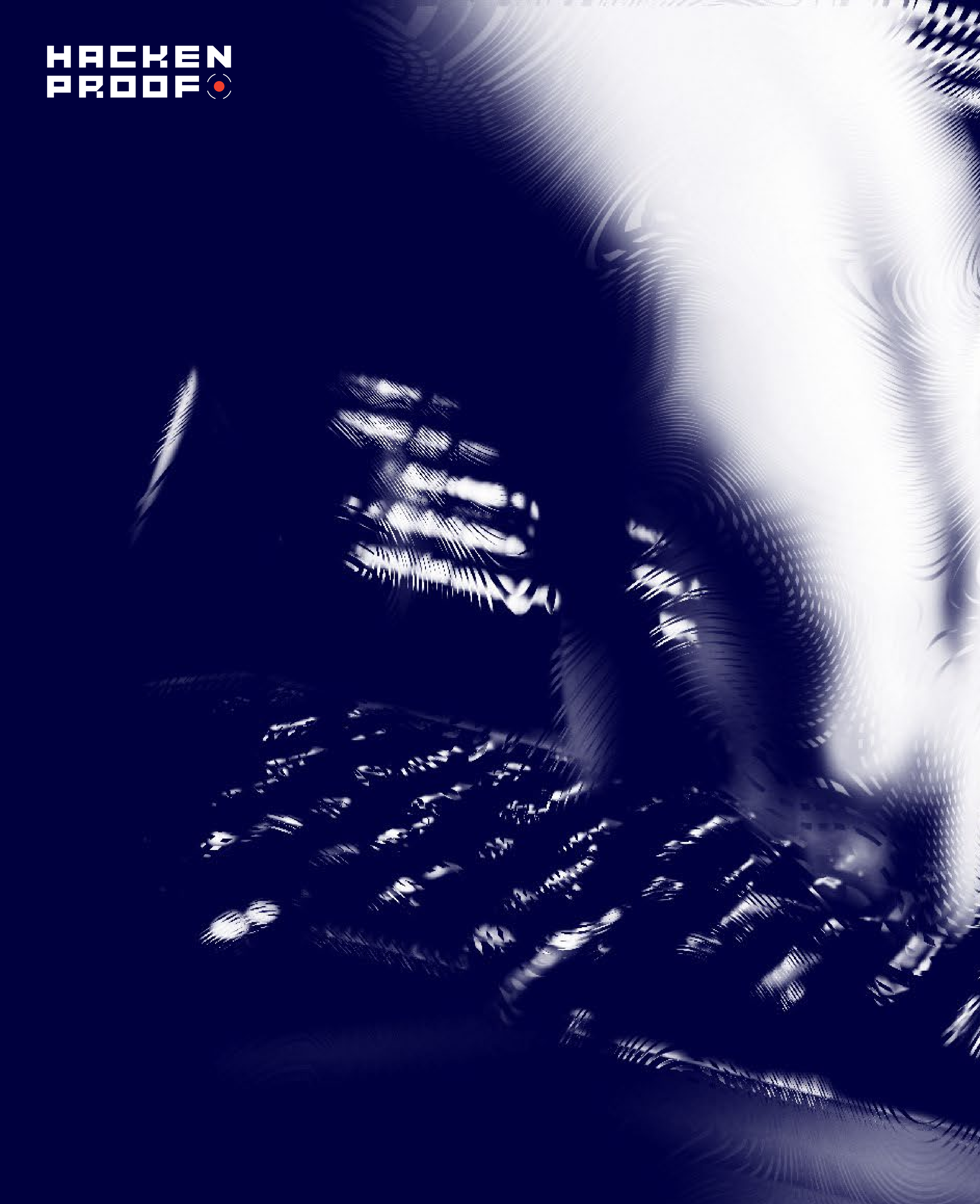
HACKEN
PROOF



Hackenproof's Guide to Bug Bounty

Table of Contents

Table of Contents

Intro

The data loss rates across industries put pressure on companies to rethink their security measures: the current methods appear to be simply untenable. Data protection demands being proactive and getting ahead of the curve.

According to a recent PWC CEO Survey Report, more than 40% of CEOs are worried about the cyberthreats.

And they have good reason to be worried: a recent report compiled by the Center for Strategic and International Studies, states that nearly $600 bn has been lost to cybercrime in 2018 alone.

As a result, global spending on cybersecurity has risen by 17% in the past 2 years and is projected to be more than $96 billion in 2018.

The hacking business is huge. The statistics reveal the magnitude and dynamics:

- According to Industry Week, the average cost of cyber attacks increased from $4.9 million to $7.5 million during the period of 2017-2018.
- The global business landscape faced a 350% ransomware attack increase, a 250% growth of business email compromise, and spear-phishing attacks affected 70% of companies.
- Technology, retail, and government are the three industries that suffer the most — about 95% of cyber attack records.
- As per Accenture, the average cost of the data breach for a business is $2.4 million.
- 43% of data breaches occur to small business causing $80,000 annual loss on average. Most small businesses never recover after a cyber attack (Smallbiztrends).
- Over 60% of digital companies will fall the victims of cyber attacks by 2020, and the average cost will exceed $150 million (Cybintsolutions).

Bug bounty programs are gaining momentum as one of the most prominent preventive tools in the context of a data breach. They provide companies with a convenient way to access a crowd of cybersecurity experts with various backgrounds without the need to actually paying for an army of cybersecurity experts.

In this guide, we will explain how bug bounty platforms work and what is the lifecycle of a bug bounty program. We will bust the myths about bug bounty and showcase a real-life success case of one of our clients.

1. What is Bug Bounty

To get a better idea on what is a bug bounty, first, we need to understand the concept of vulnerability disclosure

Vulnerability disclosure is the practice of reporting security flaws in computer software or hardware. Vulnerabilities may be disclosed directly to the parties responsible for the flawed systems by independent security researchers or by other involved parties.

Bug Bounty is the type of responsible vulnerability disclosure by which individuals can receive recognition and compensation for reporting bugs, especially those pertaining to exploits and security vulnerabilities.

When you launch a bug bounty program, you are basically telling the public that you're willing to pay a set amount of money to anyone who manages to report a security vulnerability to you. Despite how counter-intuitive it may sound to have this kind of policy, bug bounties offer a certain number of advantages over traditional penetration testing:

- Participants in the bounty are paid once a vulnerability is found, creating an incentive to do a thorough sweep of all the software. Penetration testing doesn't present these incentives since team members are paid regardless of how thorough they are.
- Many bug bounty participants are skilled full-time professionals who participate in several different hunts at the same time.
- Bounties give thousands of skilled hackers the opportunity to test their mettle, providing an incredible number of perspectives. Penetration testing teams tend to be restricted in size. Regardless of their skill, their perspective is limited.
- Companies with huge "attack surfaces" (i.e. software that is very prone to breaches) can uncover bugs that were previously left out by their own teams.

Despite all of the above advantages, bug bounty isn't a replacement for penetration testing. Penetration testing is a normal part of SDLC that's usually done before a product is released to the public. It involves a team of individuals, either outsourced or in-house, that attempt to "hack" the software or system that the company wants to release. They then report all vulnerabilities found on the product, allowing developers to fix these problems before they become nuisances later on.

During penetration testing, the team typically follows a set procedure to uncover all possible vulnerabilities. This may involve using techniques that hackers typically use to infiltrate systems and software. What you end up with is a comprehensive list of critical areas in your software that most hackers would be able to subvert. A combination of periodic penetration tests and an active bug bounty program are the best solution to ensure that an organization has a diverse pool of testers and continuous coverage.

1. What is Bug Bounty

Many companies have a mindset that building an “impenetrable wall” around their digital assets that will save them. The reality, however, is different. No matter how great the wall is – sooner or later hackers will find a weak spot in it and exploit it.

Technology is evolving all the time and your defense has to keep up the pace. The right mindset if you don’t want to suffer a security incident – is to constantly keep testing your “wall”, find vulnerabilities and fix them, before criminals can exploit them.

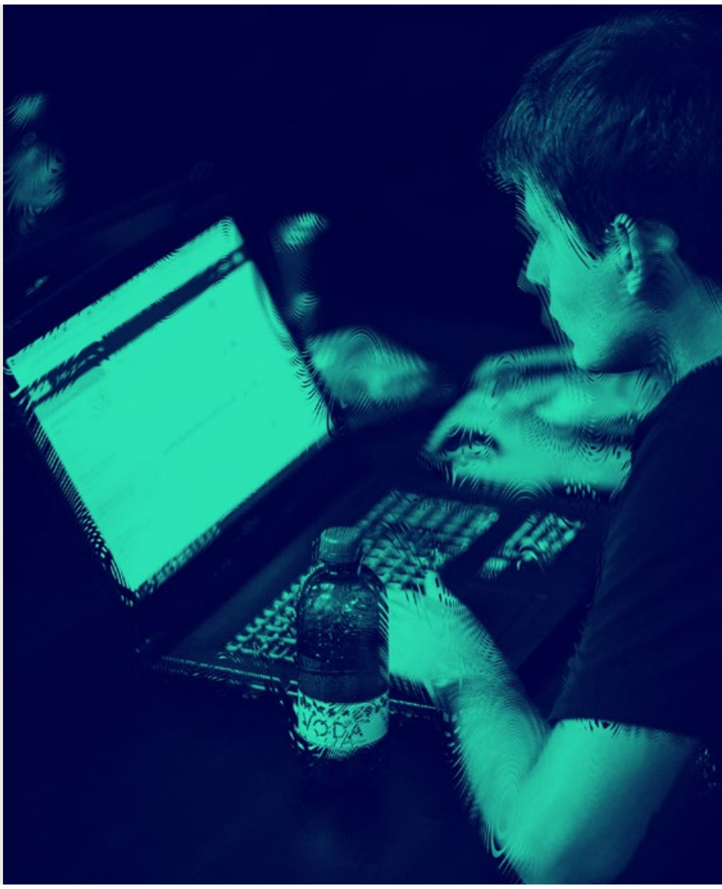
2. How does Bug Bounty fit into SDLC

A lot of companies follow common development processes when creating software. But these processes do little support for making secure software. They typically identify security defects in the verification phase if identify them at all. Firstly, fixing bugs on this phase is too expensive. Also, it is absurdly, if it is possible to foresee them.

A better practice is to integrate security from the beginning, from the planning phase. This helps find defects close to the time they're made.
With the traditional SDLC model, organizations would hold all security activities until the end of the software development lifecycle. In the past several years, many companies have switched their traditional ways to new security practices throughout the SDLC.

One of them is the Bug bounty program for your products. How can bug bounties help and secure your SDLC? Here you can see only some benefits of Bug bounty programs:

- Bug bounties assist in finding areas in the zone of attack of the highest risk
- By identifying unknown vulnerabilities, bug bounties actually help to inform on the first periods of the SDLC is the security was well designed.
- Bug bounty program results can also be used for developing training programs and support secure-coding best practices
- The pay-for-results model and diverse testing pool combine to improve upon vulnerability scanning which only discovers known issues and penetration testing results which are limited in perception and scale



Bug bounty model is cost-effectiveness and more resultative model. Public or private, continuous or short-term, the bug bounty model scales the benefits of traditional manual testing methods and goes far beyond automated testing methods to deliver real-world security assessment in real time.

By integrating bug bounty in the Security Development Process we could come to the process where a bug bounty program is running from the first release of the application.

3. How to launch a bug bounty program

Do you want to have a great volume of actionable, unique reports at a fraction of the penetration testing? But you don't know where to start with it? Let’s start from the finish. Imagine tons of good formulated security reports in order in your project.

Suppose you are receiving very useful bug submissions that your prior efforts never found, your engineering team is now able to secure your systems faster and cheaper than ever before, and you just launched a competitive bounty challenge for your top hackers. Can you manage it by yourself? Can anybody create such much reports by oneself? How could it possible?

Just run a Bug Bounty program. Sounds difficult, but we will help you to understand that it could be easy, fast and efficient.

Let's break it down step by step.

Step 1 - Prepare

Not surprising, it is the most important part in the launching of any bug bounty program. If an organization is well prepared, then they are in a perfect position to receive ongoing value out of their program. Those who aren’t can end up getting burned.

This phase focuses on laying out the groundwork: developing processes and finding the platform that best fits your requirements.

Step 1.1 Choose a type of bug bounty

Responsible Disclosure

Use our platform to receive vulnerability reports discovered by third-party hackers, free of charge.

By using the Responsible Disclosure service you will ensure that security reports end up in front of your security team while minimizing the chances that vulnerabilities are disclosed through unsuitable and unsecured channels. You’ll have a page with clear rules on how to disclosure vulnerabilities in your product and your contacts.

Bug Bounty Program

A bug bounty is a improvement of responsible disclosure with stimulation hackers by reward to look for vulnerabilities in your product.

There are several different ways to run a bug bounty program. It can be run as a public or private program, as a time-bound program, as a responsible disclosure program or a live hacking event. To feel the difference between highlights of them - see the table for comparison of them.

3. How to launch a bug bounty program

Public Programs

Everyone on the platform can participate. This is a great solution for large and mature products.

A public program gets more interest, and therefore, more vulnerability report submissions. They also come with the benefit of marketing an organization's information security program, indicating to customers and partners that security is a priority. However, the signal-to-noise ratio is lower than average, making it more time intensive and costly.

Private programs

Private programs are fully confidential and are available only to a selected number of hackers. They offer our clients the opportunity to tap into the power of crowdsourced security testing – a vast number of testers with rich skill sets and perspectives for focused testing in an invite-only program.

We help the client to hand-pick a limited number of proven researchers with skills and background that are a good match for their needs. The program is completely anonymous. This is a good solution before starting a big public bug bounty program and is also used for pre-release and early products.

Time-bound bug bounty

Time Bound. Time-bound bounties are short-term programs. They are generally private, invitation-only programs that last two to four weeks. Time-bound programs are effective options for limited budget or short-term needs. For example, if there is a new release with new features in an application that needs to be tested, an invitation-only, time-bound bug bounty can get set up quickly. Combine that with an incentive structure that awards the specific functionality on which the researchers should focus and that new functionally will get a quick security assessment. Time-bound bounties are also a good way for an organization to try the bug bounty concept

Live Hacking Event

Host a bug bounty event custom-tailored to your needs, to allow top hackers from all over the world to secure your product together with your in-house security team. Invite hackers with the necessary skills and get more vulnerabilities in the shortest time.

Type/Feature	Responsible Disclosure	Public Bounty	Private Bounty	Time Bound Bounty	Live Hacking
Incentive	Recognition	Monetary	Monetary	Recognition	Recognition
Access	Public	Public	Private	Public/Public	Public
Scope	Full Coverage	Full Coverage	Full/Partial Coverage	Specific Target	Full Coverage
Duration	Continuous	Continuous	Continuous	1-2 months	1-3 days

3. How to launch a bug bounty program

Step 1.2 Define Scope

The bug bounty program needs to have points where test should be done and things what to test. A scope should be clear, leaving nothing open to interpretation. Every scope should have completeness and clearness.

Step 1.2.1 Scope and focus area

The scope helps prevent frustrating researchers by making sure they do not expect to get a reward for a vulnerability discovered on an otherwise in scope target. You can add some special documentation in scope too. It will be a good manner from your side.

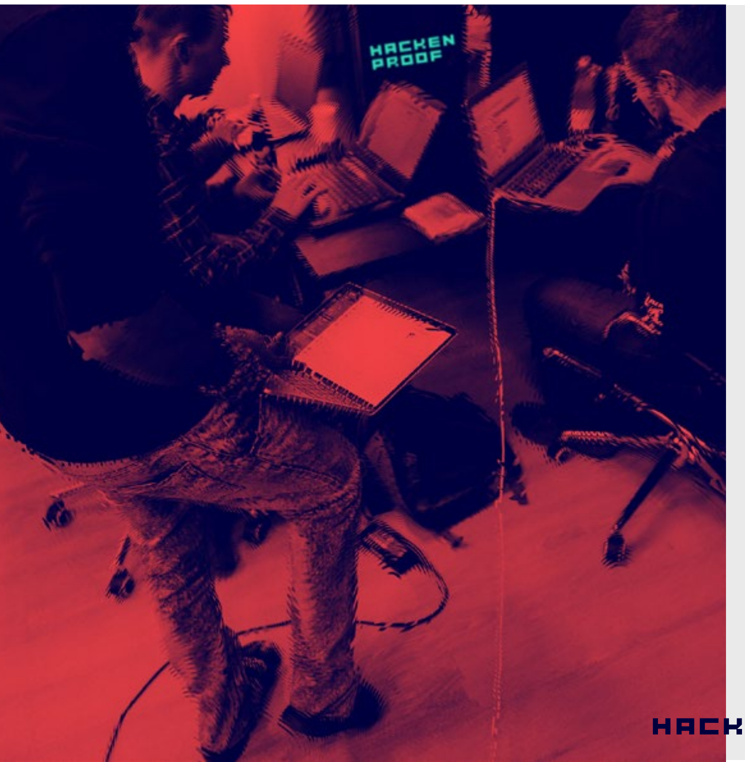
Guide researchers in the right direction by accurately articulating any exclusions. It is also essential to explicitly call out what is not in scope. The most common example are third-party services are out of scope. Or possibly, infrastructure can't handle scan. Security team can start small because can't handle all the issues, so sometimes out of scope can be changed with time.

Nice to have good formulated focus area. Focus areas may include specific bug types, particular functionality, new features, or something needs more attention. Companies also create a list of vulnerabilities that they don't want hackers to be working on. Out of scope can include vulnerabilities that don't pose a security risk to the client or is already known.

Step 1.2.2 Environment for testing

Your environments should be ready for a load of bug bounty hunters. Make sure your production and staging environments can stand up to testing.

For many of the reasons, you may also decide to set up a staging environment for researchers to test. There are some benefits to make separate staging for bug hunters:



- If your environment happens to fall over, no production clients or users are affected.
- Researchers will not contact with personal information or sensitive user data.
- You don't have to wonder if traffic is malicious, or if it's just researchers performing testing.
- Researcher traffic won't interfere with website metrics collected by other departments.
- On staging, there is the latest version of the application.

3. How to launch a bug bounty program

But sometimes there is more reasonable to run you bug bounty program on the production environment. Should you allow researchers to test on your production systems or is it better to use a publicly available test environment?

Especially if you have limited resources, to retest after staging environment or you're testing a non-web application, including open source applications or mobile applications or devices, it may be difficult to support and maintain a staging environment to test against.

Here are advantages of testing on production:

- Testing on production allows you to practice your incident response, for example testing traffic will emulate black hackers and allow you to verify your monitoring and blocking controls.
- If a vulnerability is reported you know it works in production so you will have less false positives.
- Setting up a staging environment could require significant effort to set up, maintain and keep in sync with production.
- Improves awareness towards security and performance within engineering teams.

But there may occur some problems during testing on production

- Researchers will test on a real system with real customers and real customer data. If current controls cannot prevent researchers from affecting customers and their data this may be an issue
- A researcher could publicly disclose an issue without permission or steal sensitive data
- Researchers will test from all over the world. This makes it near impossible to force rules such as "testing during business hours only". So it's important to plan for 24/7 testing

Step 1.3 Set Rewards

Step 1.3.1 Choose an incentive

The incentives associated with a bug bounty do not need to be purely financial. Other options include giving the researcher public recognition or sending them a physical
item like a t-shirt. The list below discusses possible types of incentives:

- Monetary. The most common motivator! The saying “You get what you pay for” certainly rings true for BBP’s as well.
- Swag. As simple as t-shirts or stickers to things like free air miles. A meaningful physical item can be a source of pride for a security researcher. It can be a physical manifestation of bragging rights. It also serves as marketing for an organization’s security program.
- Recognition — Finding bugs improves researchers ranking and score, giving them access to elusive private programs and recognition

Step 1.3.2 Bounty Table

How much does a bug cost? When considering the bug value, take into consideration security maturity and bug severity.

High-value submissions. Amount of submissions you are going to get is difficult to know. Know the upper end of your budget, keep a buffer and then curate your payout ranges to allow you to pay at least a few very high-value submissions before going back to your finance team. Costs of spent by internal teams are also usually unknown but can turn out to be significant. Unless you have dedicated engineers working on the program, expect to re-prioritize internal goals mid-way.

Lower reward ranges. It is also possible to start with lower reward ranges and increase them over time. Lower reward ranges can bring initial success, however, the reward range is what allows organizations to compete for talent within the market – and for sustained success, we suggest starting with the above ranges.

Apart from your company’s brand, the payout range (both lower and upper bounds) has a significant impact on the quantity and quality of submissions you are going to receive. There is good evidence to suggest that a high lowermost bound is much more impactful for a number of submissions than a high upper bound, whereas a high upper bound can incentivize researchers to spend longer durations testing specific areas of your product or testing specific class of vulnerabilities. Not all researchers care about money, but many do.

Starting amount depends on the bug bounty program and on the scope of it. Higher bounties result in more attention, especially from higher ranked hackers with polished skill sets. But do not hurry. Firstly rate the scope of vulnerabilities, then correct your bounties.

The simplest and the best way to approach this is to set up a bounty table. It illustrates how much you are willing to pay for various bugs and helps set expectations for hackers. Also, it gives you and your team a guideline to ensure fair and consistent reward amounts.

Typically you want to pay out based on the severity of the issue identified. HackenProof provides CVSS (Common Vulnerability Scoring System) scoring on the platform to assist with this. Both hackers and teams can use CVSS to calculate a severity, or simply pick one from Low, Medium, High, or Critical.

The severity of a bug. To select the specific payout, however, we need a severity for the vulnerability. The severity of a bug is important to the reward process because higher severity issues deserve higher rewards – they require more time, effort and skill to identify. To obtain severity, we must evaluate the vulnerability for technical and business impact.

Severity can be critical, high, medium and low. Here you can find some tips on how to estimate the severity of a vulnerability.

Info/Severity	Critical	High	Medium	Low
Impact	Root-level compromise of infrastructure, privilege escalation or financial theft	Affect security of the product and operations, get access to confidential information	Get access and affect several user accounts	Affect singular accounts and difficult to exploit
Vulnerabilities	• RCE • Authentication Bypass • SQLi	• Stored XSS • SSRF • IDOR	• Reflected XSS • CSRF • XXE	• Debug Info • Rate Limiting • Cryptographic Issues
Median Bounty	1,500$	900$	300$	100$
Competitive Bounty	5,000$	3,000$	700$	300$
Top Bounty	10,000$	6,000$	1,500$	500$

3. How to launch a bug bounty program

Step 1.3.3 Bounty Handling

You defined the payment for a vulnerability. But how do you actually pay out? You can do it by your financial department who is handling this, but it's an operational overhead that you definitely don't want to have. If you decided to reward hackers with swag, make sure that you have allocated resources for this process.

HackenProof provides a service for making the paying process painless and fast. You can set up a prepaid balance to pay bounties and we will do all pays for you. You'll only choose the sum to pay.

Step 1.4 Establish Triage

Triage is a process of validating submitted vulnerability reports to check if it's reproducible, unique, within the scope and poses a security risk. You can do it on your own. Or you can entrust this thing to the HackenProof team.

How to triage process:

1. Check if the report in scope. If it's not in scope or is an obvious non-issue, mark the report as Not Applicable and explain your rationale to the hacker as to how you came to this decision
2. If it's in scope, check, is the report is reproducible. If it isn't reproducible, you can mark the report as Needs More Info, explain you were unable to reproduce the issue, and ask them to provide more details as needed
3. If it is in scope and reproducible - it is a valid issue
4. If it's a valid issue and the risk is low you can mark the report as Informative. Particularly for public programs, note that if you're confident it's a non-issue, it's best practice to publicly disclose the report. If future reports of a similar nature come in, you can point to the previous report to indicate your rationale for not considering it to be an issue
5. If it is worth fixing and will improve your security posture, mark the report as Triaged. Thank the hacker for reporting the issue

3. How to launch a bug bounty program

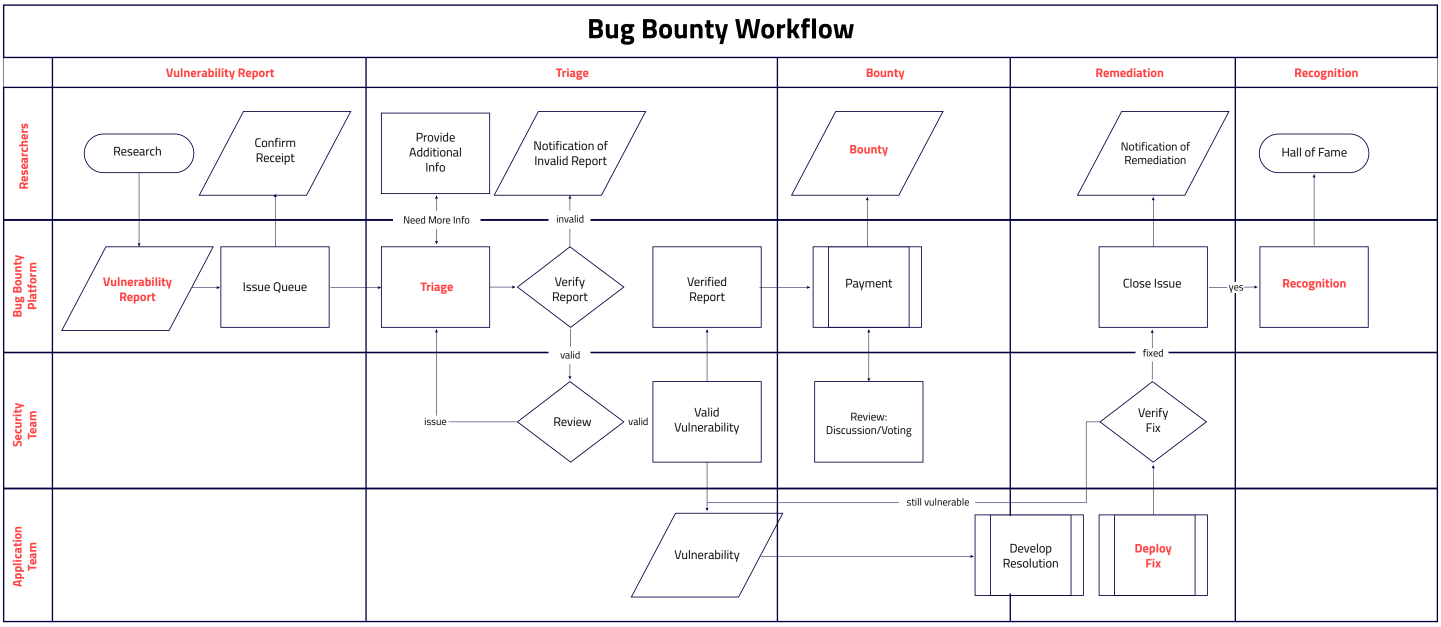
Pay attention to is this the first report of this issue? If not, mark the report as a Duplicate. You can invite the hacker to view the previously submitted report to help transparently demonstrate that they were not the first to find it.

Every bug bounty platform has a team of in-house cybersecurity specialists called "Triage Team". The job of the triage team is to verify the bugs reported by researchers and determine the severity level of a bug for the client. If your team doesn't have the time and resources to take on the flow of incoming bug reports yourself, HackenProof can help handle them for you. We'll handle all communication with hackers, reproduce reported bugs, and escalate only valid bugs to your team.

According to vulnerability management standard ISO/IEC 30111:2013 process and organizational structure should be set up for support vulnerabilities investigation. Before launching you need to set up the workflow for the vulnerability reports. You should lean on processes in your company.

Remember, that when the vulnerability is discovered, fixing it is not just a matter of applying a quick patch to solve the immediate problem. You also need to do a root cause analysis. Root cause analysis is indeed a widely-accepted security best practice. The vulnerability is still high while you didn't get to the root of the problem. Even after the fix. Also, asking researchers to recheck your fix is a good idea.

Here is the example schema of such process, you can modify it and add some steps or change some arrows.



Step 1.5 Craft the Policy

The Bug bounty Policy describes the rules of engagement for researchers that are going to be working on a bug bounty program. It's a company's responsibility, with the help from a bug bounty platform's staff, to write a clear policy, and researchers' responsibility to get accustomed to it before getting started on a program.

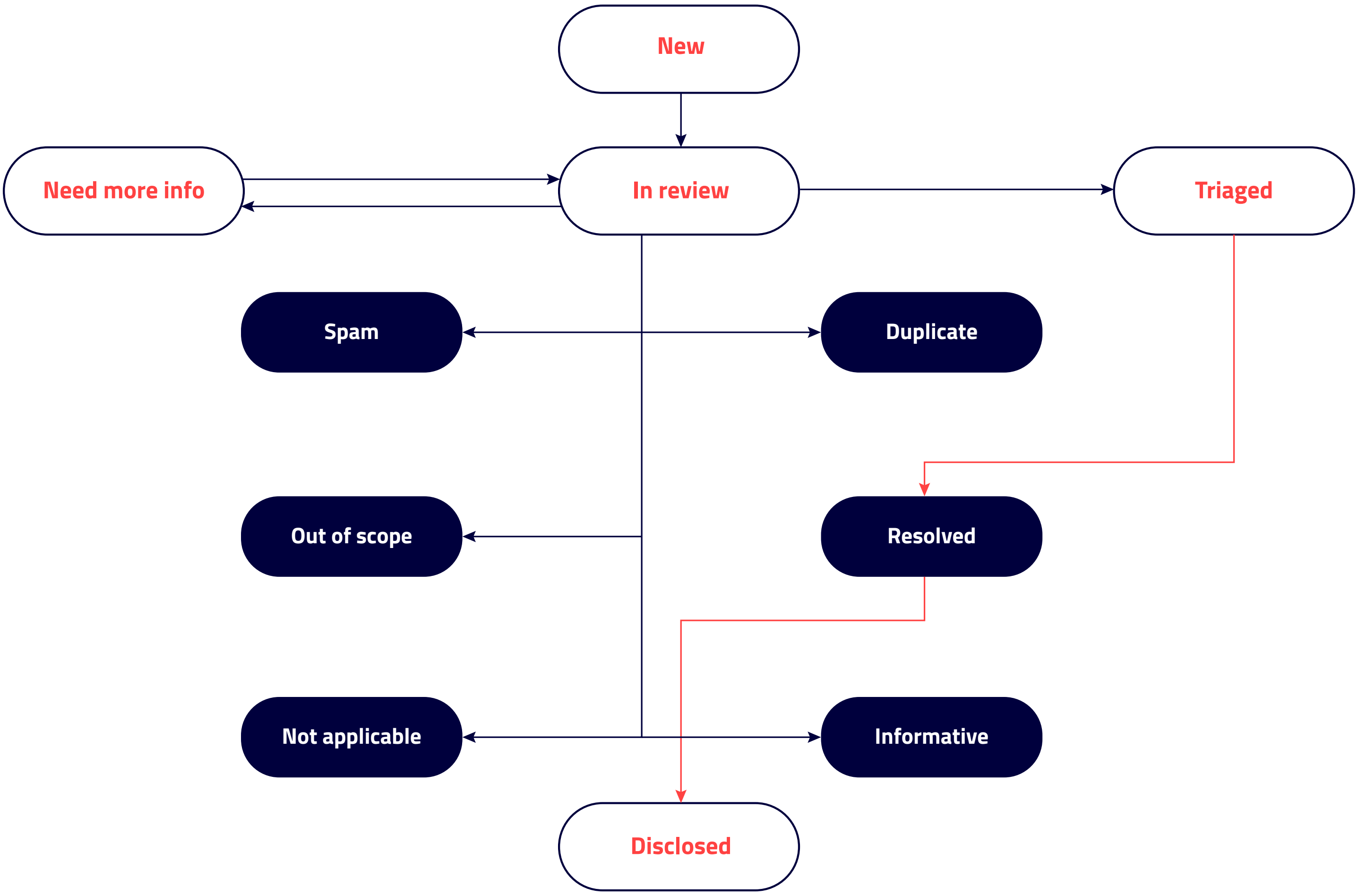
Transparency between hackers and security teams is a way to a successful bug bounty program. The "front door" for hackers to any bug bounty program is the security page, which commonly contains your disclosure policy, rules of engagement, scope, and other important information. With good configured policy page you garner more participation from top hackers. Also, the security page sets clear expectations among hackers around bounty pricing and timelines. And with good formulated security page you'll get reports with correct scope and vulnerabilities you care most about.

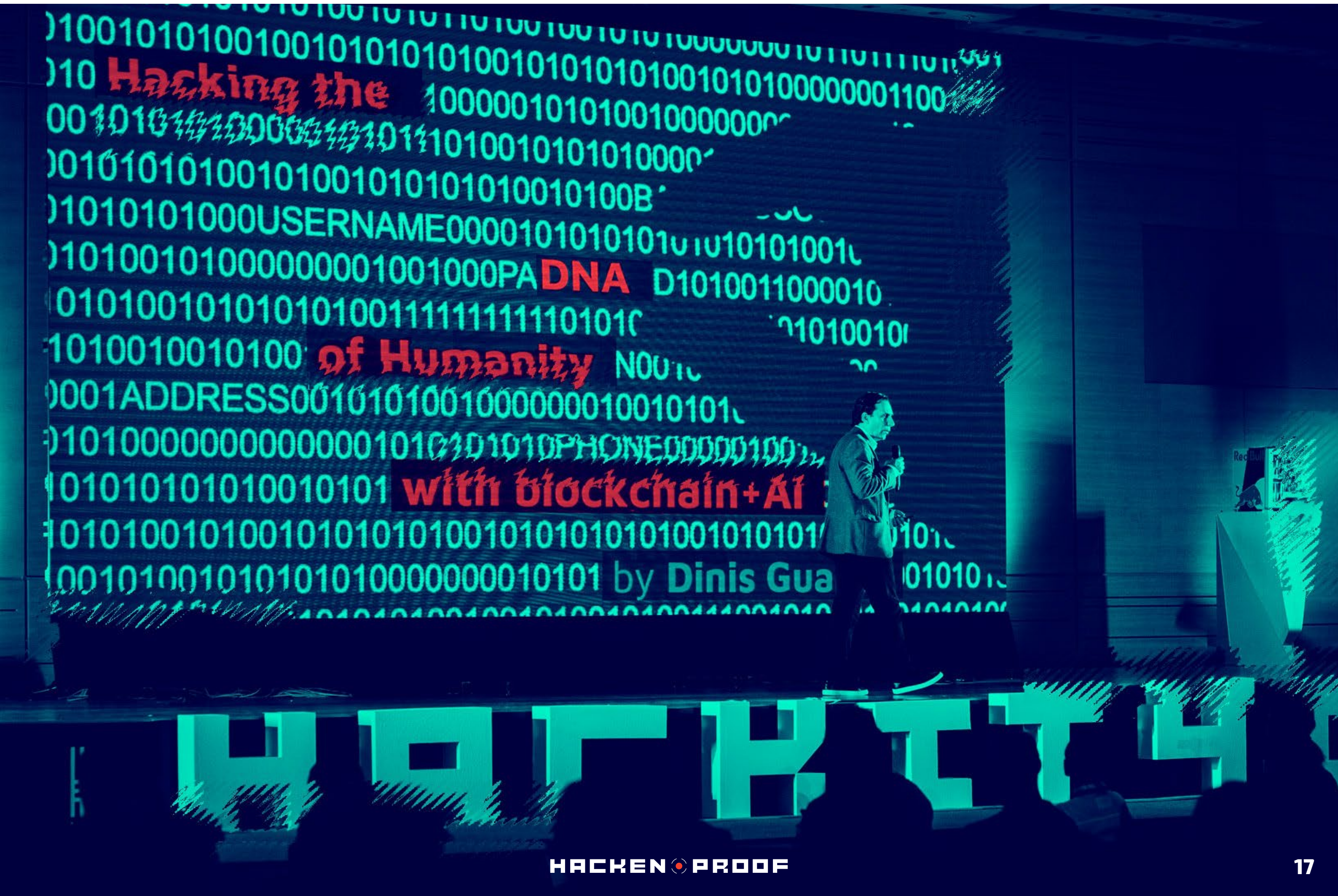
Structure of a bug bounty brief may vary, but it usually contains the following points:

1. About - a short description of a company that is hosting a bug bounty program. This provides a bit of context to researchers that are going to work on this particular bug bounty program
2. Scope - simply states what resources must be tested by researchers. "Where" should researchers be looking for bugs
3. Focus - deals with "what" researchers should be looking for. This may include specific bug types, functionality, features, some technical details - submission preferences and prioritizations etc. Companies provide as much documentation as possible, in order to assist hackers in working on a program efficiently
4. Out of Scope - companies also create a list of vulnerabilities that they don't want hackers to be working on. These usually include vulnerabilities that don't pose a security risk to the client or known vulnerabilities
5. Rewards - usually, companies themselves determine the pricing level for different vulnerabilities types, but often bug bounty platform's staff advise companies on the compensation level, in order to make bug bounty program attractive to the researchers
6. Rules of a Bug Bounty - describes in detail what researchers can and cannot do when working on this particular bug bounty program and what disclosure guidelines they should follow
7. Service Level Agreement - details how the company communicates and pays researchers during the bug bounty program
8. Safe harbor - is a provision of a statute or a regulation that specifies that certain conduct will be deemed not to violate a given rule

As you can see, bug bounty policy is a complex document and is an integral part in a life cycle of a bug bounty program. If companies get this part wrong – it's highly likely they will fail to have a successful bug bounty program.

3. How to launch a bug bounty program





3. How to launch a bug bounty program

Step 1.6 Build the Internal Process

Before starting a bug bounty program you need to be sure that your internal teams are prepared for it as well and resources are allocated. Running a bug bounty program means that your team will perform a lot of different tasks starting from attracting hackers, triaging reports, communicating with hackers, paying for bugs and coordinate the whole process. Tracing vulnerabilities back to their root causes will provide invaluable information for your team to investigate what went wrong and decide how to prevent it in the future.

During the bug bounty way you need support from the following teams:

1. The security team for bug bounty coordination, vulnerability reports validation, and communication with researchers

2. Engineering team for vulnerability fixing and product maintenance. The development team should be notified that you have additional channel getting vulnerabilities except you well-established internal process to be ready to start working on fixing vulnerabilities. It's better to plan before how you are going to deal with untypical loads with your DevOps team as well

3. The legal team for checking the terms and conditions of cooperation with third parties and evaluating legal risks. You should take in mind the protection of your company's interests and use appropriate legal language for this. Ask your lawyers review bug bounty policy and make sure that friendly hackers won't be sued for responsibly disclosing vulnerabilities

4. PR team for attracting talented researchers to the program. It's nice to make a press release to make a public statement that you are serious about the security of your customer's data and launching an exclusive program for attracting talented white hat hackers to prove it

5. Finance team for payments to third parties all over the world. This process can be organized by yourself and you will pay every hacker separately that can cause unnecessary operational overhead or make a deposit and pay through bug bounty provider

Step 1.7 Select a Provider

The bug bounty program can be run on your own. It means that you need to engage hackers, deal with managing a ton of reports, processing payments alone. For a lot of companies, it's easier to use a bug bounty provider who ensures that all processes work smoothly. It saves you a lot of time, allows you to get started quickly, focus on exploring really cool bugs, not painstaking work and don't make all mistakes along your bug bounty way.

3. How to launch a bug bounty program

If you want to run a bug bounty program by your own from the very beginning you need to think about the following points:

- How to find and engage researchers to start looking for vulnerabilities on your site?
- Who will process, validate and prioritize incoming reports?
- How should researchers submit their findings in a structured way?
- How are you going to reward researchers?

Bug bounty provider takes it over and provides a tool and guidance to run a program. It usually includes:

- A pool of specialists who can instantly start hacking and you don't need to spend a lot on marketing and gather researchers bit by bit
- A security team who will carefully triage reports, speaks the same language as hackers and consult you along this way
- A payment handling mechanism to ensure timely rewards all over the world
- An easy in use tool with a ticketing system inside for announcing a program, keep rules updated and submitting bugs in a structured form
- Different bug bounty engagement models to fit your needs

3. How to launch a bug bounty program

Step 2 - Launch

Launching your bug bounty program looks like a usual release. It is not something incredible. Sure, it's not a good idea to start in on Friday evening. But at all, don't worry. The foundations and preparations have been set. You have your well-crafted policy, agreed upon SLAs, locked down your budget, and you're ready to start your launch process!

Step 2.1 Start Small

It's important to take it easy, at least at first. In the first day, expect two (maximum) serious, non-duplicate vulnerability reports. The average customer targets finding ten bugs in the first two weeks – you can target more if you like.

Beginning with a much more manageable amount of hackers contributing to your program means you can test your processes.

Step 2.2: Analyze

Bug bounty programs are a good thing for finding vulnerabilities. But even better thing is the data that comes out of it. Data with time will help you move from fixing individual problems towards root causes of systemic issues and overarching improvements to your security program.

Collect different statistical data and it will help you to analyze and make conclusions about your bug bounty program. Specifically, regarding the character of vulnerabilities, the severity of bugs, vulnerability types. You'll know where is your budget spent.

HackenProof provides quite a bit of data around your program in your dashboard, such as:

- Reports Overview
- Payment statistics
- Bounty Payouts
- Severity average
- Time management
- Top hackers

Use this data and make your own statistic. With this analysis, you can change some parameters and improve your bug bounty program in the future.

Step 2.3: Exchange Feedback

Developing a good relationship with hackers will keep top talented hackers coming back again and again. You did a good job, wrote a clear bug bounty policy. But sometimes hackers make mistakes, and it’s important to provide feedback when this happens.

When reports are coming in hot, hackers hitting the right targets and finding the types of bugs you want to see. Do you need some feedback to researchers? Yes, now is the time to ask for and listen to any feedback from hackers participating in your program.

It’s important for you to provide feedback to hackers as well! Every security team (yours included) has their own preferences for how bugs are reported, as well as which properties and bug types they care about most.

In an ideal case, every report will be valid, with clear reproduction and precise steps. But not in the real world. Nothing ever goes that swimmingly.

A lot of things can go wrong - for example, you need more info to reproduce the report or you feel the bounty amount is too big or mistakes in triage process, whatever. It can happen anytime. Even if you have the clearest rules page among all bug bounty programs. In this case, only tact, diplomacy and patience will help you. Be quiet and calm. And try to solve the problem through negotiations.

3. How to launch a bug bounty program

Step 3 - Refine

Once the bug bounty program has begun, white hat hackers are testing the software and report bugs they find. Researchers write up a bug report explaining in detail how to exploit a vulnerability and submit it via the platform's website.

You are running your bug bounty program, you are getting vulnerability reports, your security process becomes. And what's now? Don't stop!

Step 3.1: Scale

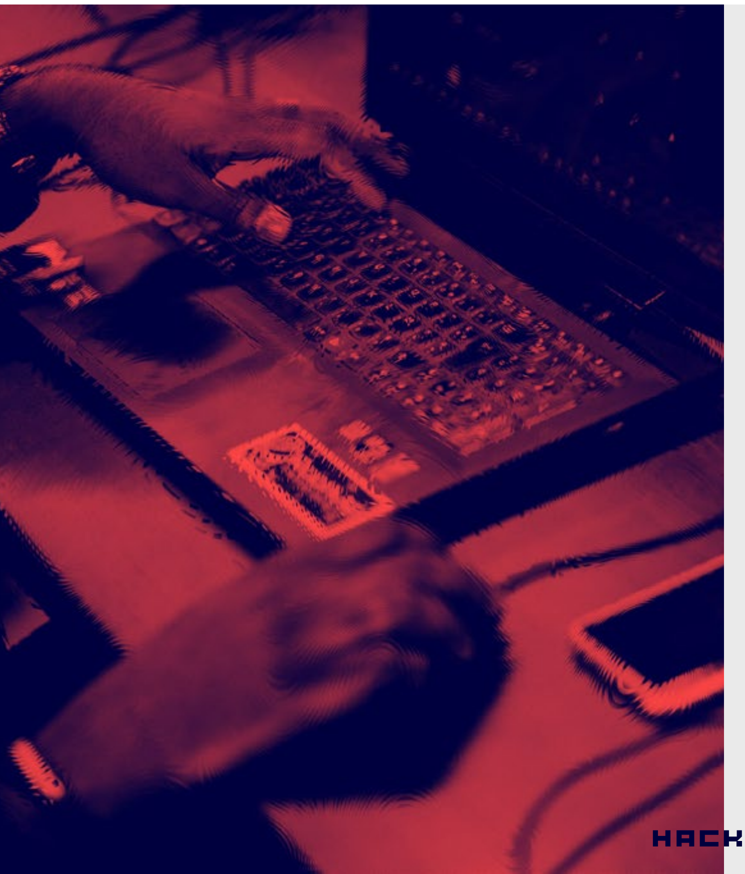
If you're in the groove of triaging reports, filing bugs internally,etc., and it seems like volume is steady (or even starting to tamper down), now is the time to turn up the volume.

There are three primary ways to incentivize more participation and increase the volume of reports:

1. Juice the bounty amounts
2. Increase the scope of your program
3. Add more hackers to your program

More hackers + more scope + increased bounties = bigger, worse bugs which means more value. It might seem counterintuitive, but you want vulnerabilities.

Over time, you'll want to increase your scope to cover just about anything. The key things to consider as you expand your scope are:



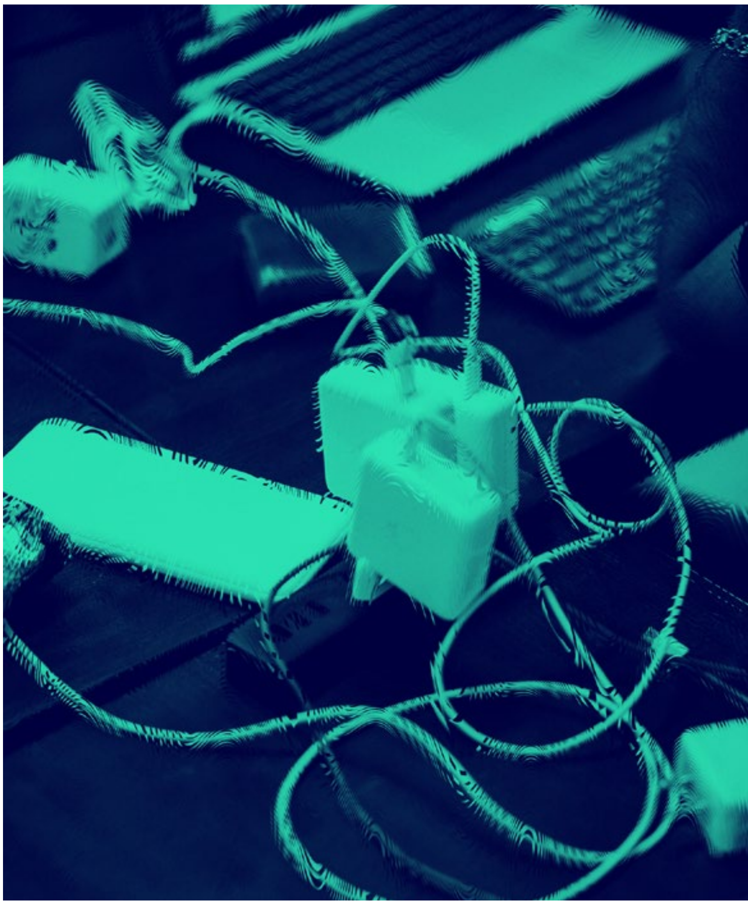
- Can I handle the growth load?
- What types of bugs do I want more?
- As you add sites to your in-scope attack surface (one at a time), you can iterate through the previous processes you went through when first launching your program.
- Your bug bounty program should be treated as a dynamic program that will need adjustments and tweaks over time. As you introduce a new scope, illustrate in your security page what sorts of issues you're hoping to find. Consider what you can do to give hackers more information to better aim their efforts at uncovering juicy bugs

3. How to launch a bug bounty program

To whatever extent you haven't already, you'll want to automate as much as possible. As a way to solve the problems is templates. If you manage large volumes of tickets and find yourself making the same statements over and over - you can do some templates for different cases.

Recruit hackers that understand your business and tech and are willing to invest the time to test your logic and applications.

Reward bonuses if you want to focus researchers attention into yours products. For example, if the business really cares about vulnerabilities that allow access to certain types of sensitive data then increase the rewards for these issues. Remember, the best results will come from systematically manual testing. The better your Program, the more repeat engagement you'll see. Researchers will get a feel for your product and maybe they will even know it better than you do.



Step 3.2: Improve

Remember, security improves when bugs are fixed, not when they are found. To improve existing process do next steps.

- Add automation
- Speak with your team
- Collaborate with hackers
- There's a friendly hacker waiting for news on remediation of the bug. Don't be afraid to give them occasional updates, especially if you think you're at risk of slipping out of your remediation timeline SLA.

4. Myths and facts

Even though bug bounty programs have been invented two decades ago, it’s only in recent years they have become more or less well-known solutions that companies employ to fend off different hacker attacks. There are a lot of bug bounty myths regarding the whole process because it is pretty new for many companies. Here is the debunking of the most common bug bounty myths:

Myth # 1: Only large tech companies run bug bounty programs

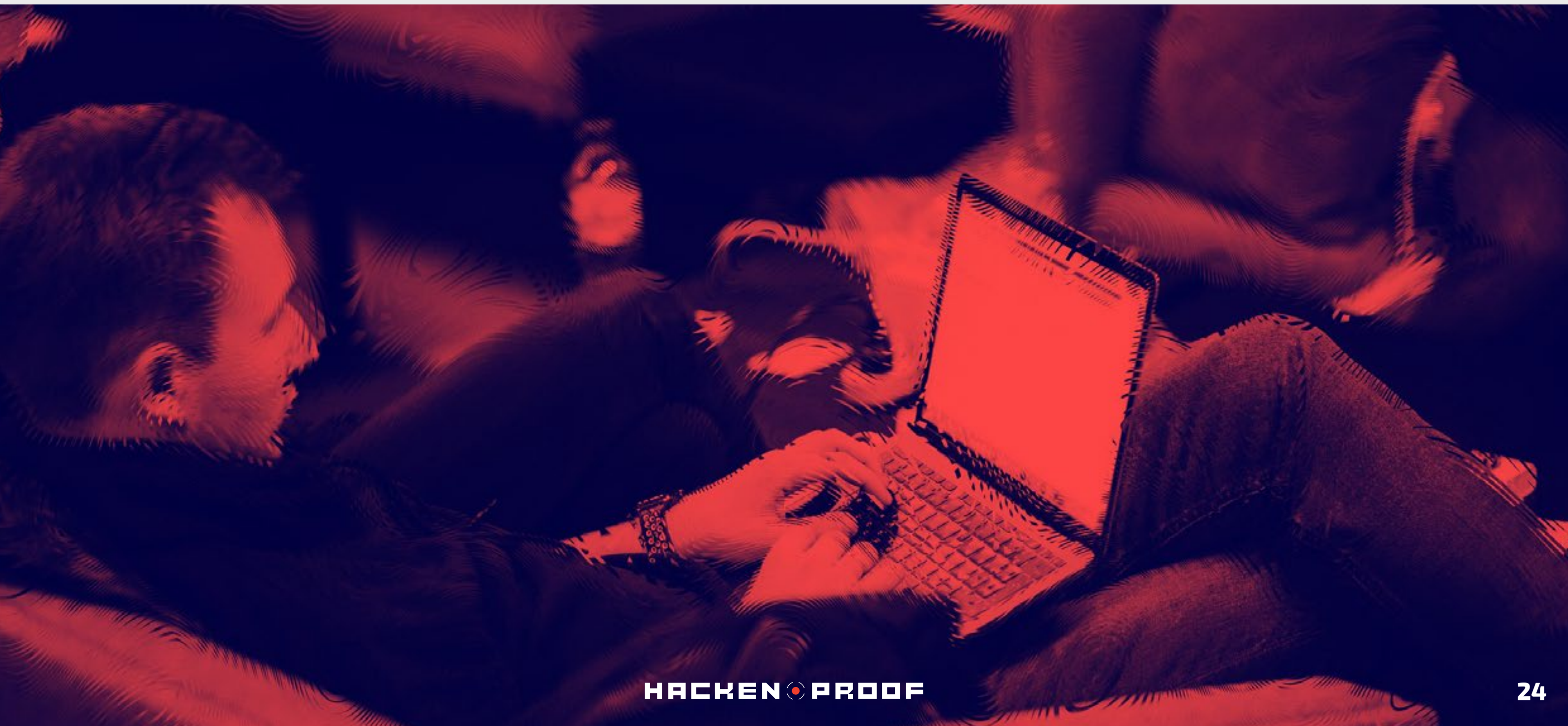
That used to be a correct statement in the past, for a simple reason - only large companies had both the media presence and qualified employees to successfully conduct bug bounty programs. With the rise of bug bounty platforms, such as HackenProof, that’s not the case anymore. Nowadays bug bounty platforms help almost any kind of businesses launch and manage successful bug bounty programs.

Bug bounty programs provide an opportunity to level the cybersecurity playing field. We are ready to help you to combat constant external threats which can appear in your work by arming you with a complex of strong measures and expertise. The bug bounty program is necessary not only for technology companies but sometimes even more necessary for non-technical ones. It's really so because some security measures are forgotten to be done.

The market for bug bounty programs is growing at an ever faster pace. Just look at Gartner's latest research that says that more than 50% of large corporates will employ crowdsourced security testing by 2022.

As products and companies grow, it becomes more and more difficult to maintain an adequate level of security. At scale, bug bounty programs become more and more cost efficient compared to traditional penetration testing.

Companies from all branches of industries regardless of their sizes and types can take this advantage now. The bug bounty model has evolved to be effective and flexible.



4. Myths and facts

Myth # 2: Hackers can't be trusted

This is a quite common misconception among the business community. We hear it all the time: "How can you guarantee that cybersecurity researchers won't take vulnerabilities they find and sell them on the black market?" Quite a reasonable concern. There are some main points companies should bear in mind when it comes to white hat hackers:

1. White hat hackers are public figures. Being public is "part of the game"

We've interviewed a lot of white hat hackers during our work and we constantly ask them the question "why have you chosen a white hat hacker path?"

Their responses are always the same "I don't want to go to jail". Researchers genuinely enjoy what they do on a daily basis, they don't want that to stop. The overwhelming majority states that they don't do it for the money, but because they enjoy looking for vulnerabilities in software products. To prove the point - please, check out the Responsible Disclosure Policy of SproutSocial that specifically says "no compensation will be awarded for bugs found". Yet, there is still a long list of researchers who have submitted vulnerabilities.

2. Selling vulnerabilities on the black market in most cases doesn't make any sense

A black market is a hostile place, where people get scammed all the time, so selling anything on it is very easy and safe.

Also, the black market is not interested in either low or medium vulnerabilities. Selling them on the black market would be close to impossible. At the same time, companies are prepared to pay top dollar for critical vulnerabilities on their bug bounty programs. Bounties for Remote Code Execution can easily cost tens of thousands of dollars.

So, to sum up, selling vulnerabilities to companies via bug bounty programs is easy, legal and can make researchers reasonable money.

3. Legal bug hunting means you can gradually build a reputation

Another big advantage of being a white hat is that one can continuously build up his reputation as time goes by. With every vulnerability found, white hat hacker gains reputation points, as well as monetary rewards. This information is usually included in their CVs as a demonstration of their skills. Bug Bounty platforms feature leaderboards, where cybersecurity researchers compete with each other. Bug Bounty platforms award top researchers with custom merchandise. After a certain amount of time, successful researchers become influencers and are being asked to give talks at conferences and are being invited to participate in bug bounty hackathons across the globe. And this sort of fame is more important for white hackers than one-time profit from the sale vulnerability.

4. Background checks

When dealing with clients that want an extra layer of confidence, we offer private bug bounty programs. We hand-pick researchers that we've verified personally and we can also conduct background checks, upon client's request.

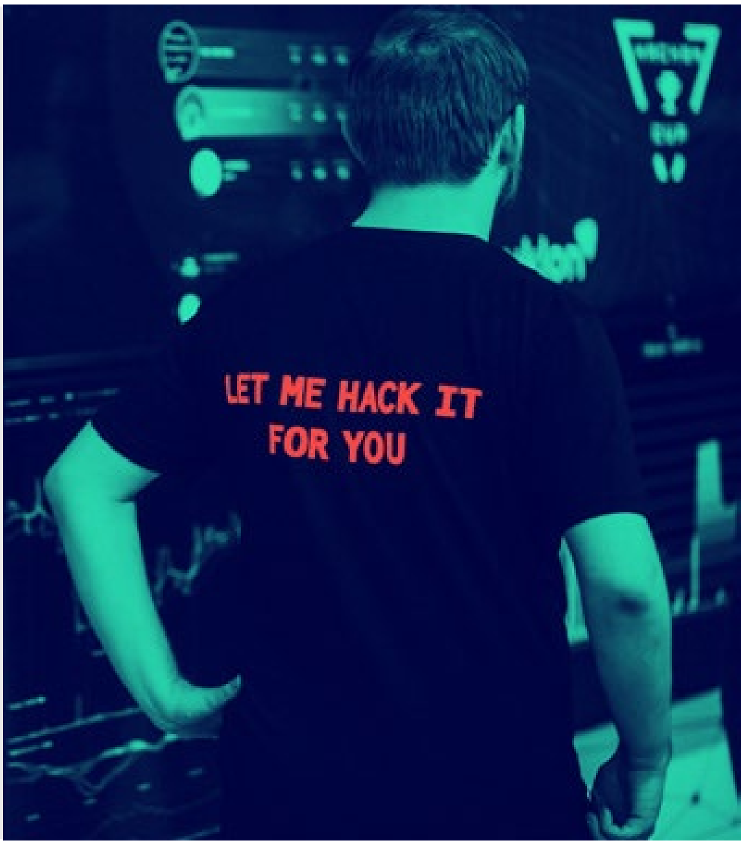
4. Myths and facts

Myth # 3: Some important data can be stolen during a bug bounty program

Hacker in bug bounty platforms is white hat hacker. And for them is more important to get the bounty or the reputation, not to do some inappropriate things.

If one researcher will find some vulnerability and don't report it, the other researcher will find and report it. You'll be able to protect yourself and prevent the attack by leveraging crowd-sourced approach with many eyes involved.

To find vulnerability means that hacker will show the way to stole something important, but will not do this. In reality, incidents when a researcher goes rogue extremely rare, and we actively work to prevent them and ban those who have inappropriate behavior and break bug bounty rules.



Myth # 4: Bug bounty program is a very risky action

The main thing, that the risk of being vulnerable is bigger than the risks associated with running a bug bounty program.

When you are giving permissions for researchers to find holes in your application you protect yourself in the future. You are giving your organization more knowledge and control. And as a result, you reduce and minimize all risks.

Also, you should choose a good partner to run a bug bounty program with. This action will make the risk lower. Because in the community all members follow a set of rules and partner controls that everybody follows these rules.

The bug bounty program is not a risky action at all. It is another type of security testing your application.

Myth # 5: Bug bounty programs don't provide results

A bug bounty is an addition to traditional safety methods. This myth is easy to bust by looking at the numbers. Let's start with the big companies that everyone is familiar with. A recent report says:

- Google has paid security researchers over $15 million for bug bounties
- Facebook has received 17,800 submissions from researchers in 2018 alone and awarded over $1.1M. Bounties paid since 2011 exceed $7,5 million

Both Facebook and Google wouldn't have spent their time on bug bounty programs if they didn't yield results.

Myth # 6: Bug bounty programs are too expensive and hard to budget compared to penetration testing

It's important to look at the pricing policy of a bug bounty program, compared to penetration testing:

1. The client is in control of the budget at all times

Companies can easily put a "limit" on the bug bounty budget if they wish to do so. That way, a company can be certain that payments to researchers won't go "out of control". Bug bounty helps to control the budget and provide the best way to manage it.
There will no such problem as an irrelevant waste of budget at all. Because before launching bug bounty program bug bounty platform does scan your product for simple security vulnerabilities so-called "low hanging fruits" and remove all of them out of scope. That's why all vulnerabilities founded by hackers will not too simple and easy to catch.

2. During a bug bounty program, companies pay only for verified vulnerabilities

During penetration testing, companies pay for the procedure itself, regardless of how many vulnerabilities are found during the process. Bug bounty programs, however, pay bounties to white hat hackers only for verified vulnerabilities.

Myth # 7: Bug bounties are hard to run and manage. It is too long to wait for results

During a bug bounty program, companies usually prefer to "outsource" all the daily management process to a specialized team. By doing so, companies don't have to distract their in-house technical team. Here's how the whole process works when launching a managed bug bounty program:

1. Bug bounty policy is being published on a bug bounty platform's website and white hat hackers start looking for vulnerabilities within the scope of a program

2. White hat hackers find and report vulnerabilities through a website of a bug bounty platform

3. A triage team of a bug bounty platform verifies all vulnerabilities that are being sent by researchers and prepare reports for a client. Reports contain a description of a vulnerability and a detailed instruction of what needs to be done in order to fix a problem.

Managed bug bounty programs save companies a ton of time by taking on daily communications with white hat hackers that report vulnerabilities. The larger the company's digital footprint, the more time can be saved by a managed bug bounty program.

According to the statistic, the first few critical bugs are in the first week after the launch of the program. Results will not take long to wait.

Myth # 8: Bug bounty programs attract additional attention from hackers

The product is tested every day, you even don't know about it. You've got no reports and no information regarding your security status. When your product is in production it is better to be informed regarding your vulnerabilities before it will be exploited by cybercriminals.

There are some types of bug bounty models. One of them is the public bug bounty program. Everyone on the platform can participate. If you want to start with small steps before goes public, the other option is a private bug bounty program. Private program means that it is fully confidential and are available only to a selected number of hackers. Private programs are limited to trusted and vetted researchers, giving organizations the power to better control the scope of what is tested, as well as how it's tested. It offers clients the opportunity to tap into the power of crowdsourced security testing – a vast number of testers with rich skill sets and perspectives for focused testing in an invite-only program.

5. Customer Story: Reducing Risk by Using Crowdsourced Security Testing

To illustrate how bug bounty programs help our clients supercharge their security, we'll take a look at how the security team at Kuna, one of the largest crypto exchanges in Eastern Europe, have been operating their public bug bounty program with HackenProof for over 12 months.

Product type: Managed Bug Bounty
Start Date: 21/01/2018
Reports submitted: 84
Vulnerabilities resolved: 24
Hackers participated: 38

How Hackers Stole $1B From Cryptocurrency Exchanges In 2018

Due to the sharp rise in popularity of cryptocurrencies (i.e. high demand to trade crypto), we saw a sharp rise in the number of exchanges. From the point of view of cybercriminals, this was a perfect storm - you have entrepreneurs wanted to exploit the market opportunity, creating a large number of crypto exchanges, that are trading in assets which aren't controlled or governed by any institution. This meant that security was not the main priority for founders who were building exchanges. Hackers quickly recognized the lucrative opportunity to earn easy money by exploiting vulnerabilities in crypto wallet software and servers. It's no surprise that around $1.1 billion worth of cryptocurrency was stolen in 2018 alone.

The security team at Kuna recognized early the need to get in front of potential security issues.

Let's hear directly from Roman Cherednik, CTO at KUNA Exchange:

HP How did you make the decision to start a bug bounty program?

> We operate a platform that is extremely sensitive to vulnerabilities. Sometimes even a small mistake in implementation may cost a lot. So Bug Bounty program is an extra measure for us that improves our security by leveraging the community of white hackers.

HP Why did Kuna choose to host a managed bug bounty program with HackenProof? Why not just self-host and manage it by yourselves?

> We tried. It appeared to be quite a time-consuming task to engage the engineering team to process and manage the Bug Bounty inbox. So we decided to use BugBounty-as-a-Service to delegate this task to HackenProof and let professionals process, triage and validate the incoming reports. They forward us the only requests that we need to focus on. As a bonus, the issues are now unified and go with precise reproduction steps using the terminology of our platform that we use ourselves.

5. Customer Story: Reducing Risk by Using Crowdsourced Security Testing

How has launching a bug bounty program impacted Kuna's cybersecurity strategy?

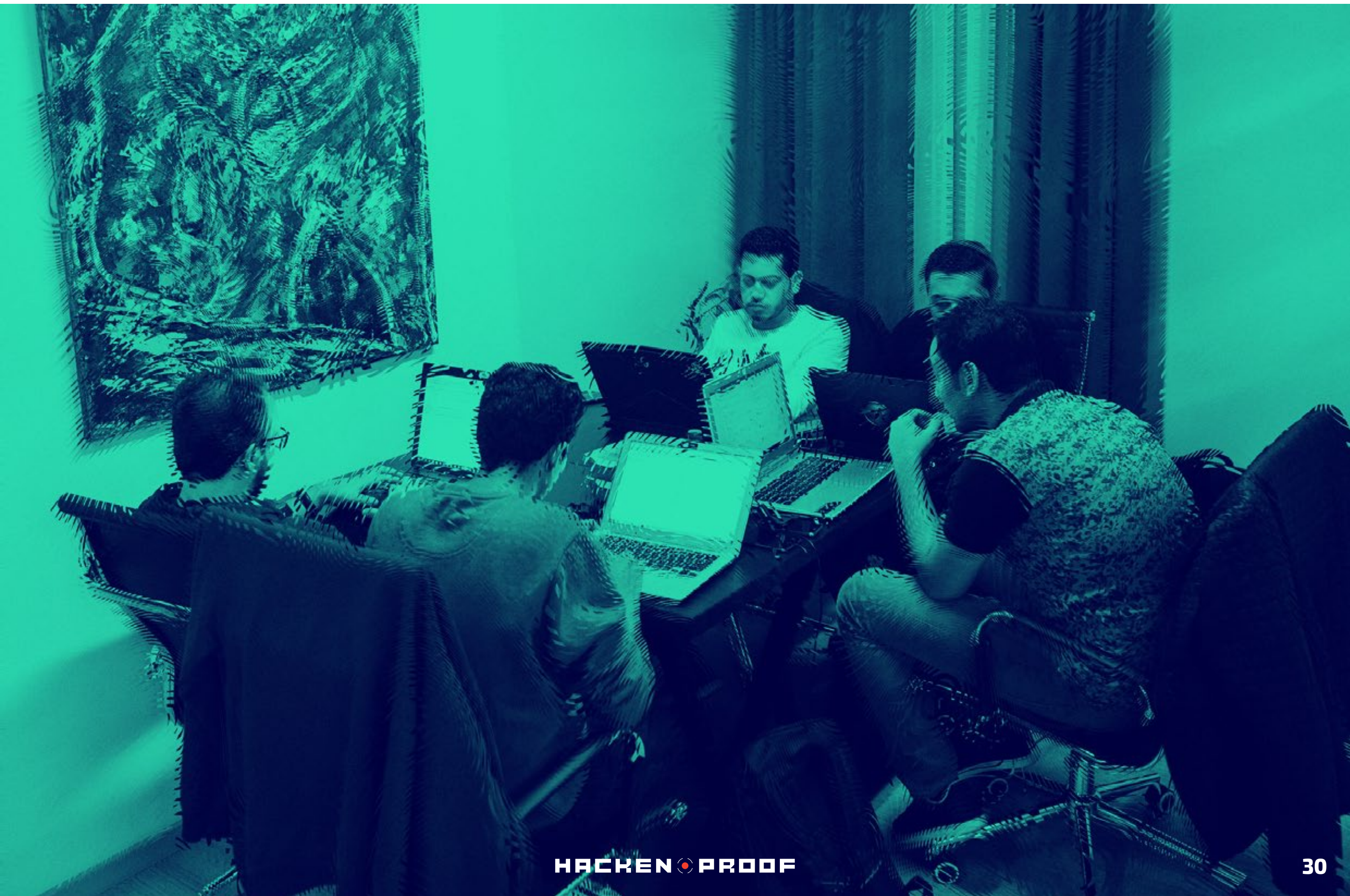
We have already fixed multiple issues that white hackers reported to us and we expect this collaboration to expand further-going as we continuously update our platform with new integrations and features.

Any highlights of hacker interactions so far?

Probably, the ones that ask for money upfront :)

What advice would you give other organizations on launching their bug bounty program?

Just do it and it will pay off.

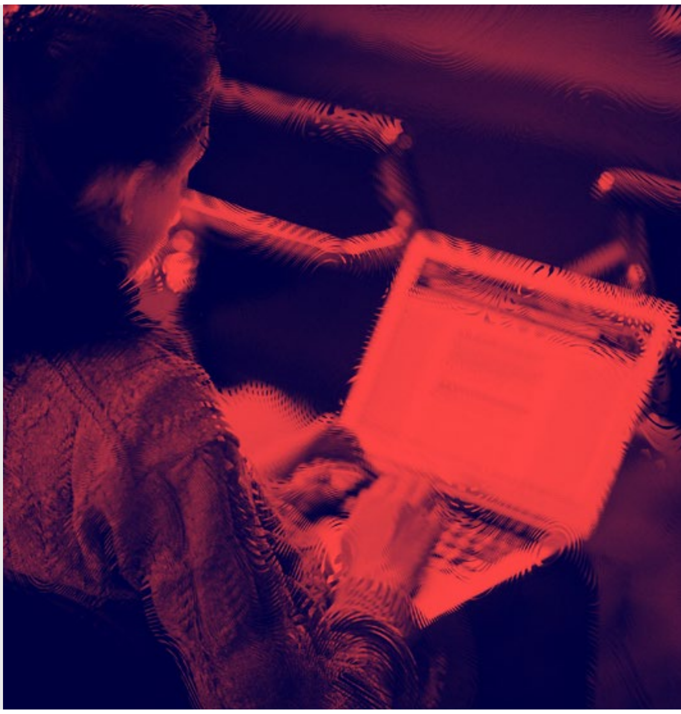


6. Augmenting traditional cybersecurity solutions with Bug Bounties

Our digital era dictates the rules. Every year cyber attacks become more advanced and cybercriminals find new ways to breach systems and steal a company's data, funds or trade secrets. Nowadays, due to cybersecurity vulnerabilities, companies may lose hundreds of millions of dollars as well as suffer a significant blow to their reputation. While corporations are aware of the problem and increase budgets for cybersecurity, conventional methods that were effective in the early 2000s aren't going to be good enough today. In order to keep up with the constantly increasing number of advanced cyber threats, companies have to adapt and recognize that cybersecurity is a continuous problem that requires constant diligence in order to suppress.

Traditionally, one of the most influential and effective ways to provide security feedback in the development lifecycle is via penetration testing. During penetration testing, penetration testers manually test the application using real-world attack scenarios resulting in high fidelity, actual findings of not only how the application is vulnerable but what the impact of that security flaw is. Because it is so effective, it has been adopted by various compliance regimes. PCI DSS, for example, requires that in scope applications are pentested "at least annually and after any significant infrastructure or application upgrade or modification".

In a waterfall model, where there is a series of security touchpoints along the software development lifecycle, penetration testing is done towards the end right before an application is deployed to production. It is a deployment gate that cannot be passed until pentesting is complete. If a major security flaw is discovered, the deployment is delayed until the code can be fixed. A manual process that usually takes a week or two and has some chance of delaying an application deployment, even more, is the antithesis of devops. In a world where production code deployments are highly automated weekly, daily or even multiple times per day activities, this kind of manual security testing can no longer be a gate.



Application development efforts can modernize this manual security testing control by adding bug bounties and using them to augment the penetration testing program. Bug bounties are not a replacement for penetration tests. However, because of their ongoing nature, bug bounties provide continuous feedback. Pentesters have a limited amount of time for each engagement, typically only a week or two. During that time, they try to cover the breadth of an application's entire attack surface often limiting their depth of coverage. Combining penetration tests with a bug bounty program enables them to be focused on high risk, new, or recently changed application functionality. Penetration testers have the benefit of bug bounty findings as a starting point for where to take a closer look.

6. Augmenting traditional cybersecurity solutions with Bug Bounties

The combination of bug bounties and penetration testing allows for a decoupling of pentesting as a traditional gate to application deployments, while maintaining a balance of high application attack surface coverage and continuous security feedback to developers. Adding a bug bounty program helps information security by providing continuous feedback on real-world attack scenarios by real-world hackers to developers as releases to production occur.

Nothing in this world can guarantee absolute security, but bug bounty programs can help significantly reduce the risk of a cybersecurity incident by utilizing a crowdsourced security approach.

We are always happy to talk to companies that want to learn how a crowdsourced security approach can help their businesses increase the security of their products.

GET IN TOUCH

HACKEN
PROOF

Contacts

info@hackenproof.com
hackenproof.com

Headquarters
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond,
Eesti, Kesklinna, Estonia

R&D Office
Parus Business Centre
01601, Kyiv, Ukraine
2A Mechnykova St, 12th floor