bugcrowd

Ultimate Guide to

Offensive Security

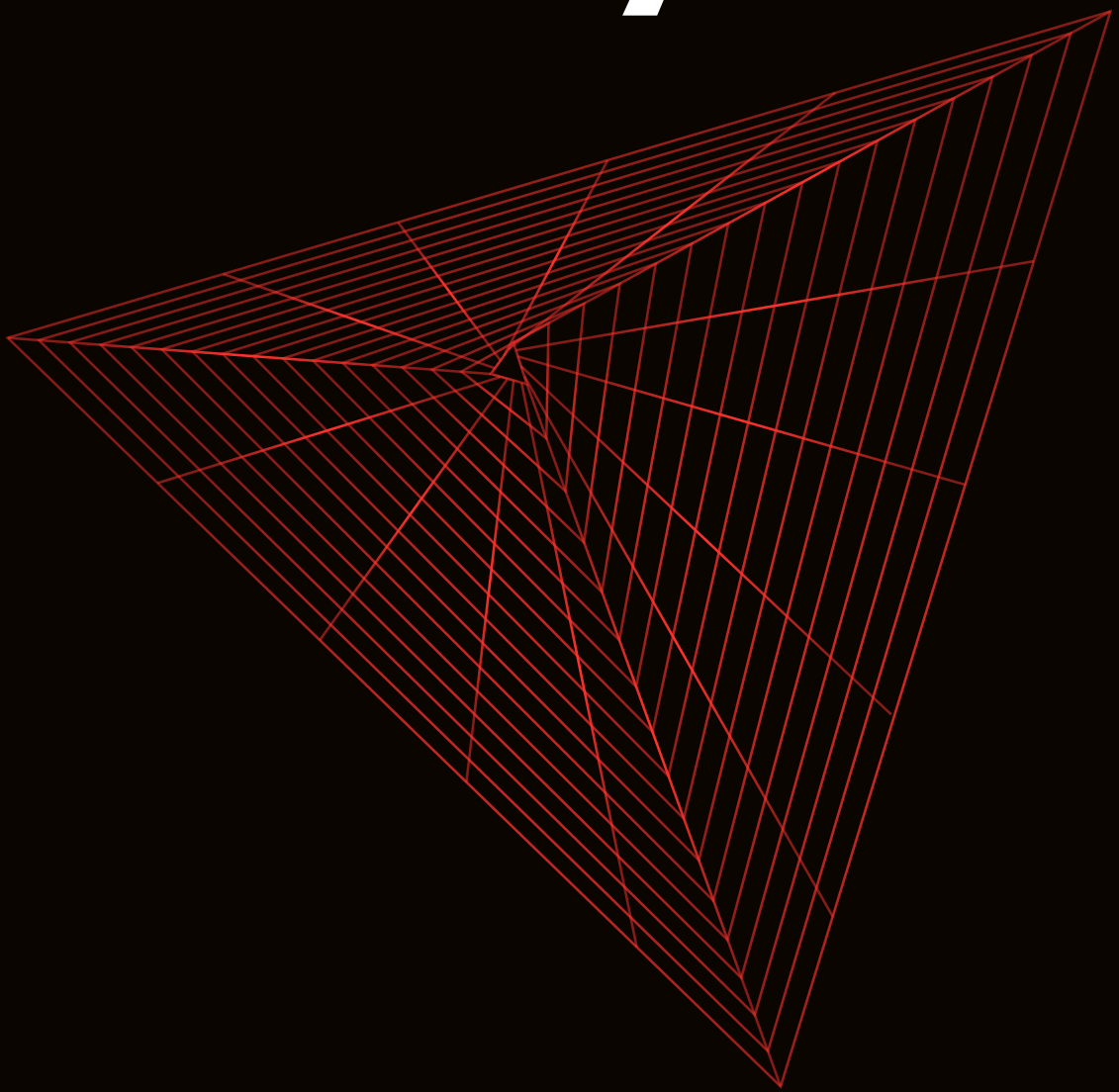


Table of Contents

Introduction

In December 2023, threat actors found two zero-days in Ivanti's VPN products. Given that Ivanti products are widely popular, their extensive usage meant that **threat actors were able to access the systems** of many US corporations and federal agencies.

At least 2,100 Ivanti appliances were hacked with these two zero-days. But that wasn't the end. Ivanti deployed patches in late January, only to quickly follow up with the disclosure of new zero-days. A few days later, agencies warned that threat actors had also found ways around the original patches.

Unfortunately, exploits just like the ones Ivanti dealt with are becoming more and more common. In 2023, 26,447 vulnerabilities were found. The number of vulnerabilities found has been growing year over year—roughly 25,000 vulnerabilities were found in 2022 compared to 20,000 found in 2021. Additionally, 84% of hackers believe that there are more vulnerabilities now than at the start of the pandemic. On top of the disclosure of sensitive data, these vulnerabilities cost companies exorbitant amounts of money. In 2023, a data breach cost, on average, $4.4 million. It may also take months to find and patch a zero-day. But most importantly, user data and customer trust may be irrevocably lost as the result of a breach.

Security measures like firewalls and scanners, which are parts of defensive security, may help protect against known exploits, but they are ineffective against the constant barrage of new exploits. Offensive security measures are the only methods that help companies find and patch new exploits.

As such, offensive security is a necessary component of a robust security strategy. Without it, a company can't stay ahead of threat actors.

In this guide, we cover the basics of offensive security: what it is, why it's important, and how you can get started with limited time, resources, and money.

What is offensive security?

Offensive security experts use threat actors' methods on your systems. These experts **simulate attacks** to identify hidden exploits. With knowledge of exploits and attack vectors, you can then patch these exploits before threat actors get to them.

Another way to think about offensive security is to compare it to other familiar security practices, specifically defensive security. Defensive security focuses on building robust defenses to prevent and ward off attacks. In contrast, offensive security is about discovering what attacks and exploits are possible.

A successful security strategy isn't an either/or—it should contain **both offensive and defensive methods**. Offensive security helps discover new exploits. Once those exploits are found, defensive security helps address the gaps.

Offensive security vs. reactionary security

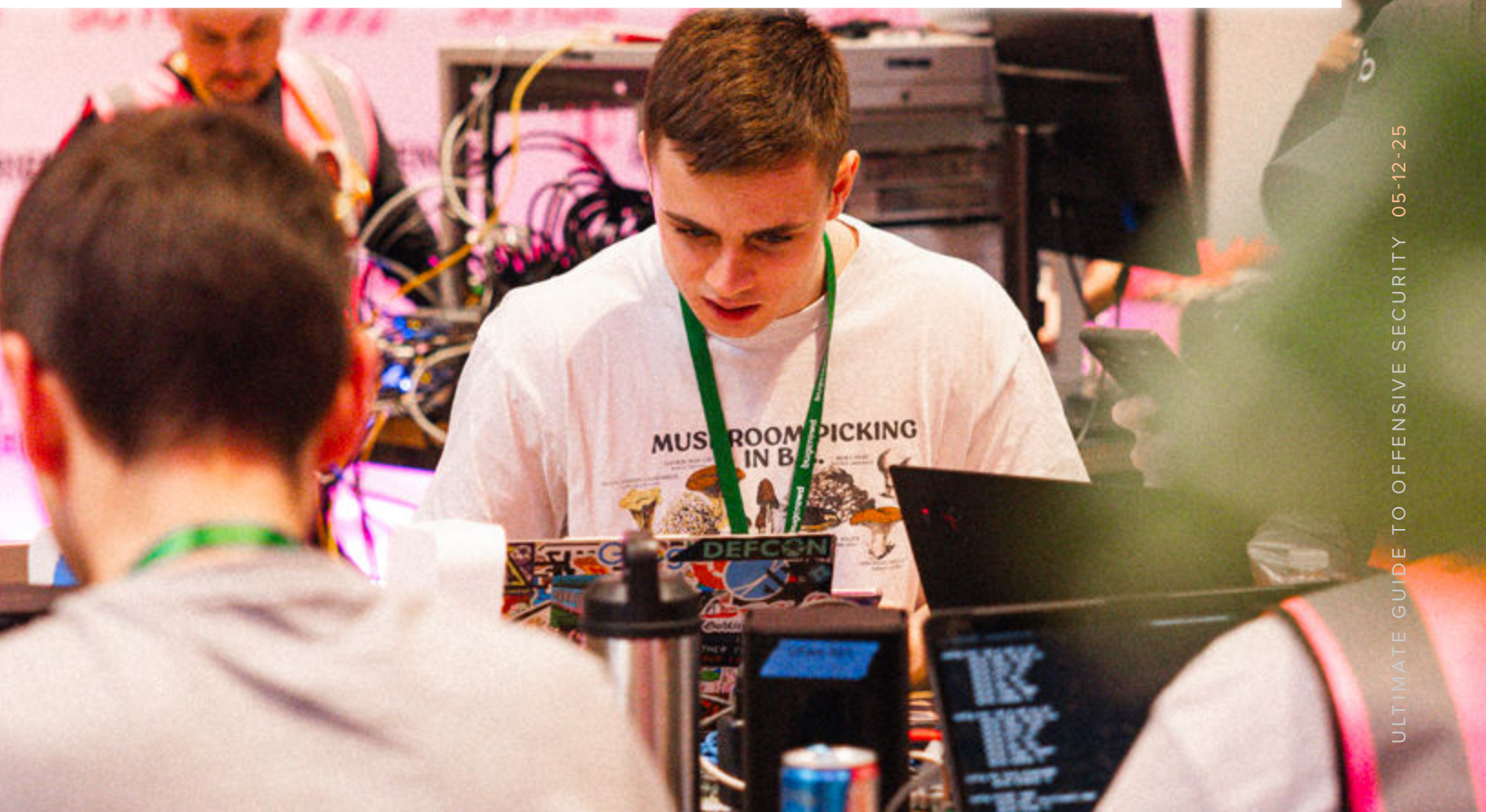
Offensive security is a form of proactive security. In other words, it is a constant process of identifying and fixing exploits before attackers can take advantage of them. Its complement is reactionary security. This means getting attacked and then patching the exploit that made the attack possible.

Both types of security are necessary. In an ideal world, you would be able to find and fix all exploits before threat actors get to them. In reality though, there will always be a new exploit waiting in the wings, and sometimes, threat actors will find it before you do. So, having quick incident response and disaster recovery plans, which are vital components of reactionary security, are essential to mitigate the fallout.

How do you know if you're pursuing offensive or defensive security?

We put together a handy table of the common offensive and defensive security practices below:

Offensive security	Defensive security
Pen testing	Vulnerability scanning
Red/purple teaming	Antivirus software
Bug bounty engagements	Firewalls
Vulnerability disclosure programs	Intrusion detection systems
	Incident response tools



Why pursue offensive security?

Offensive security is the only method that allows companies to identify new exploits before threat actors do. In security, unknown exploits can be the biggest cause for concern, primarily because most companies won't have the defenses in place for these kinds of attacks.

In contrast, while known attacks (like spear phishing) are worrisome, there are best practices to minimize the likelihood and fallout from these attacks.

Unfortunately, there are always new findings being added to the exploits list. This is in large part because security is a **cat-and-mouse game**. Companies patch their vulnerabilities and beef up their security. Subsequently, threat actors account for this and try new tactics, techniques, and procedures (TTPs) to find other exploits. Companies then patch up the new exploit before the cycle continues endlessly. Accordingly, companies can never rest on their laurels.

However, by employing offensive security, the cat-and-mouse game can actually become an advantage to be leveraged by security experts. Hackers can be employed to probe systems with the TTPs that threat actors use.

Offensive security is also **technology-agnostic**. Each offensive security test may be limited to probing only one specific technology, but the overall offensive security process transfers to new technologies immediately. Defenses for cloud exploits don't transfer at all to AI exploits, but having a pen testing process allows you to swap in an AI security practitioner for a cloud security practitioner with minimal effort.

Lastly, offensive security **improves with scale**. The chances that any singular offensive security test will reveal serious exploits are low. But, by working with different experts, all with their niche methods, the chances increase significantly. In contrast, scale has a more limited effect on defensive security. Increasing the number of people working on a firewall might only increase its effectiveness by small amounts.

Why don't more companies engage in offensive security then?

Effective offensive security has upfront costs. You first need to set up a consistent testing process. You then need to train people within your company to run tests on your systems or make the business case to onboard vendors to support an offensive security strategy. Keep in mind that if you add a new technology or vendor, more internal ongoing training would be required.

Because of the associated cost, the finding of new exploits is seldom prioritized over patching the laundry list of existing exploits.

Crowdsourced security solves these problems. Bugcrowd's platform makes it easy to find the exact experts you need to carry out specific tests, run the tests, and convert the findings into actionable steps for your systems. It handles the process, enabling you to scale offensive security with ease.

We dive more into this later in the guide.

How does offensive security fit into your strategy?

Offensive security needs to be part of a **larger risk management strategy,** with defensive security playing a critical part. It doesn't matter if you find every exploit possible if you don't actually patch those exploits in your systems.

When exploits are inevitably found and abused by threat actors, you will also need reactionary security measures to both minimize damage and implement mitigations for the future.

In the larger security scheme, offensive security works best as a **complement** to defensive security. Offensive security tests identify weak points and vulnerabilities in your systems. Additionally, they also reveal the exact methods used to take advantage of the vulnerabilities.

With this information, you can set up defensive measures against these specific attack vectors.

An ideal security process would be a **loop.** In this loop, the newly beefed up system (the output of the defensive security phase) can be fed as the input to the offensive security phase to both measure the new defenses' effectiveness and to find new vulnerabilities. The cycle can then repeat.

In the next section, we delve into some frameworks to help you get into the offensive security mindset.

How does offensive security fit into your strategy?

Offensive security frameworks: Thinking like a hacker

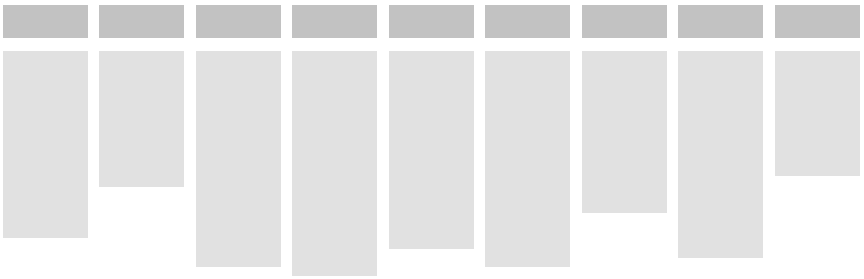
The most important part of offensive security is the **mindset** it engenders—it encourages experts to **think like a threat actor.**

There are commonly implemented frameworks that detail the most widely used phases and techniques of attack. Once you identify which framework(s) are the best fits for your organization's use cases, you can choose specific tools (which we'll discuss later) for implementing your offensive security practice.

Below, we cover three of the most common frameworks.

MITRE—ATT&CK

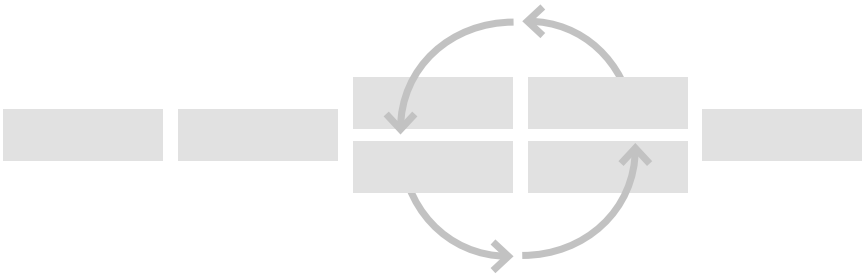
LINK



MITRE's ATT&CK framework features the comprehensive breakdown of threat actors' tactics and techniques. There are three versions of the framework, one each for enterprise, mobile, and industrial control systems. There are 14 different tactics (e.g., reconnaissance, resource development, and credential access) in the enterprise version. Each tactic lists many techniques threat actors may use to execute that tactic. The framework then describes each technique, lists sub-techniques, and details ways to both identify and mitigate the use of the technique. ATT&CK is very comprehensive and is also updated consistently. Learn more about getting started with this framework here.

MANDIANT—TARGETED ATTACK LIFECYCLE

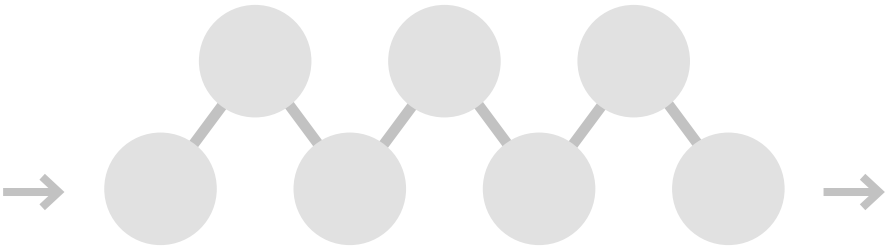
LINK



Mandiant's Targeted Attack Lifecycle outlines the general steps threat actors take to attack systems. First, threat actors do reconnaissance of a system to identify weak points. Then, they exploit a weak point in an "initial compromise" of the system, often in the form of phishing. Next, they establish a foothold (e.g., through a backdoor) so they can access the system repeatedly. This leads to a loop of reconnaissance and gaining access to more resources in both the initial and adjacent systems. Finally, the threat actor steals data or resources, completing the attack.

LOCKHEED MARTIN—CYBER KILL CHAIN

LINK



Lockheed Martin's Cyber Kill Chain framework breaks an attack down into seven steps. The framework views each step as a vital link in the overall chain of attack; breaking just one link breaks the whole attack. The seven steps are as follows:

1. **Reconnaissance:** Find weak points in a system.
2. **Weaponization:** Create a payload to take advantage of a weak point.
3. **Delivery:** Send the payload to the victim (through the web, USB, email, etc.).
4. **Exploitation:** Use the payload to exploit the weak point and run code on the victim's system.
5. **Installation:** Install malware on the victim's system.
6. **Command & control:** Create a channel to remotely exploit the system.
7. **Actions on objectives:** Fulfill the original goal of the attack (e.g., data exfiltration or remote code execution).

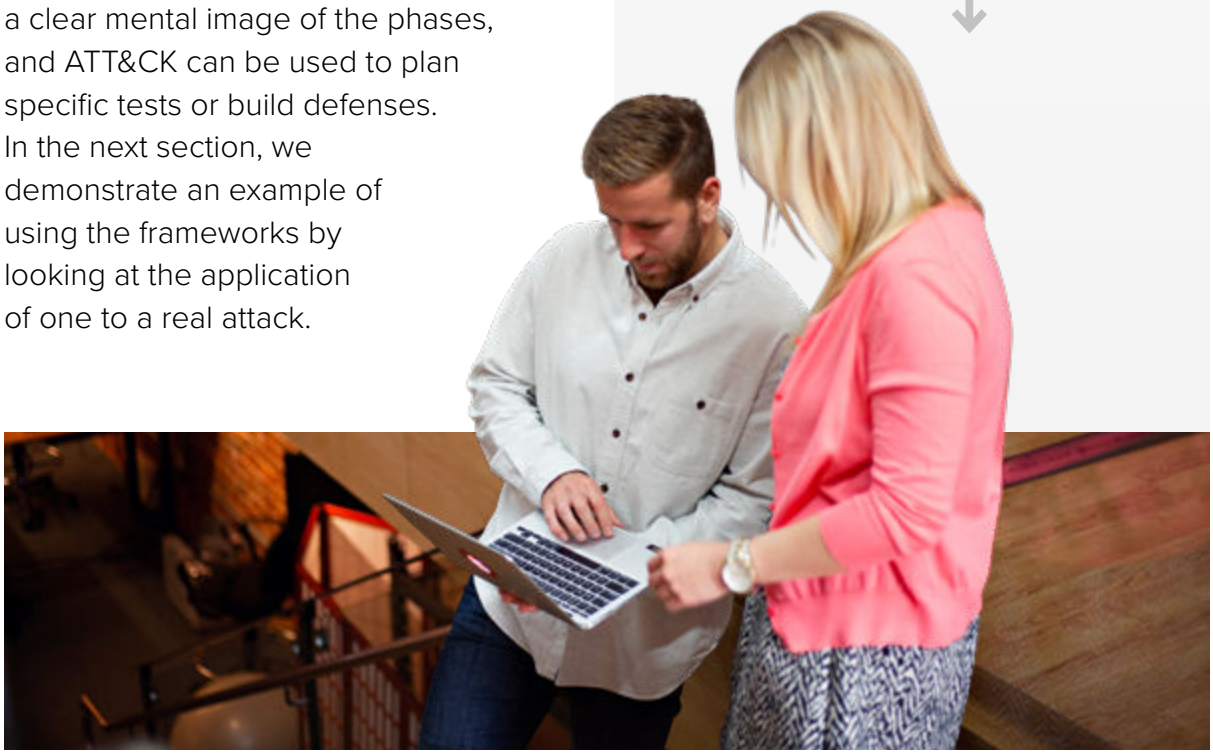
How to use a framework

Adopting a framework helps your security team **ease into offensive security.** Frameworks can function as checklists to begin with, evolving into processes over time.

You may want to incorporate **multiple frameworks** into your strategy. For instance, Cyber Kill Chain offers a simple framework for thinking about threat actors, but it doesn't reveal how threat actors execute each phase.

ATT&CK details everything you would want to know about tactics (and techniques), but any given attack may only use a few of the listed tactics.

A combination of both would work well; Cyber Kill Chain would enable a clear mental image of the phases, and ATT&CK can be used to plan specific tests or build defenses. In the next section, we demonstrate an example of using the frameworks by looking at the application of one to a real attack.



Yahoo hack: Exploring the threat actor's mindset

In 2014, two threat actors, Aleksey Belan and Karim Baratov, **hacked Yahoo and exfiltrated data** on up to 500 million user accounts.

Let's analyze the attack using the MITRE ATT&CK framework.

Reconnaissance

Belan and Baratov researched the Yahoo organization to identify employees to target. The exact method they used is unknown, but they specifically targeted mid-level employees who had access to Yahoo's technical resources.

Weaponization

The threat actors designed spear phishing emails that included a malicious link. The link probably installed malware (though the exact malware is unknown) on the victim's computer that would reveal their Yahoo credentials. Belan and Baratov may have crafted multiple emails, all targeted at specific Yahoo employees.

Delivery

Belan sent the email to their targeted victim, after which the victim clicked on the link. From there, the malware gave them the credentials they needed to begin the attack in earnest. Of course, other layers to their scheme were involved, such as how their emails managed to bypass spam filters or why the malware wasn't detected.

Execution

With the employee's Yahoo credentials in hand, Belan was able to access Yahoo servers and start looking for privileged information.

Persistence

Belan and Baratov set up a backdoor (exact method not divulged) so they could access the servers at any time, even if their stolen credentials expired.

Discovery

Belan started exploring the resources available to him in Yahoo's servers. He found Yahoo's User Database, which contained contact information (including recovery emails) on user accounts. The database also had a cryptographic "nonce," which could be used to generate access tokens for user accounts.

Credential access

Belan and Baratov first identified the accounts they wanted to infiltrate from the User Database. Then they used the nonces to generate access tokens for those accounts, which ultimately gave them access to the accounts of Russian government workers and journalists and US government workers, among others.

Exfiltration

Belan and Baratov copied data from within targeted users' accounts and data from Yahoo's User Database, extracting these data out of the system. In total, 500 million accounts were breached.

The threat actors in this attack were **very precise;** out of the 500 million user accounts they could have accessed, they generated access tokens for only 6,500.

Furthermore, Yahoo only suspected 26 accounts had been hacked when they identified the breach. It took another two years for Yahoo and the FBI to realize the full extent of the breach.

Offensive security techniques and tools

Now that we understand the general offensive security methodology, let's discuss how offensive security practices are actually carried out.

First, we'll cover the common methods used, and then we'll go over useful offensive security tools. To use these methods and tools, **specialized knowledge** is required, so their implementation generally relies on collaborations with ethical hackers. When setting up offensive security practices, there are common tools that greatly assist the work of hackers.

Common methods

Offensive security practices at their core involve thinking like a threat actor. Below, we cover the main methods that actually put this mindset into practice.

Red teaming

Red teaming is an exercise where an offensive security team (the red team) covertly simulates an attack on a company's systems, looking for any weaknesses to gain access and simulate impact. The open-scope and covert nature of red teaming lets it emulate threat actors, who similarly would strike without warning and would gain access by any means necessary.

The internal team responsible for preventing, identifying, and responding to breaches (the blue team) responds to the red team attack as if it were a real intrusion.

In most cases, the blue team doesn't know about the red team's activities. A variant called purple teaming combines the blue and red team into one: the red team calls out weaknesses as it spots them and the blue team can monitor and respond immediately.

The end result of red (or purple) teaming is deep knowledge about system weaknesses. This knowledge can be used to prioritize roadmaps, upgrade defenses, and uplevel teams.

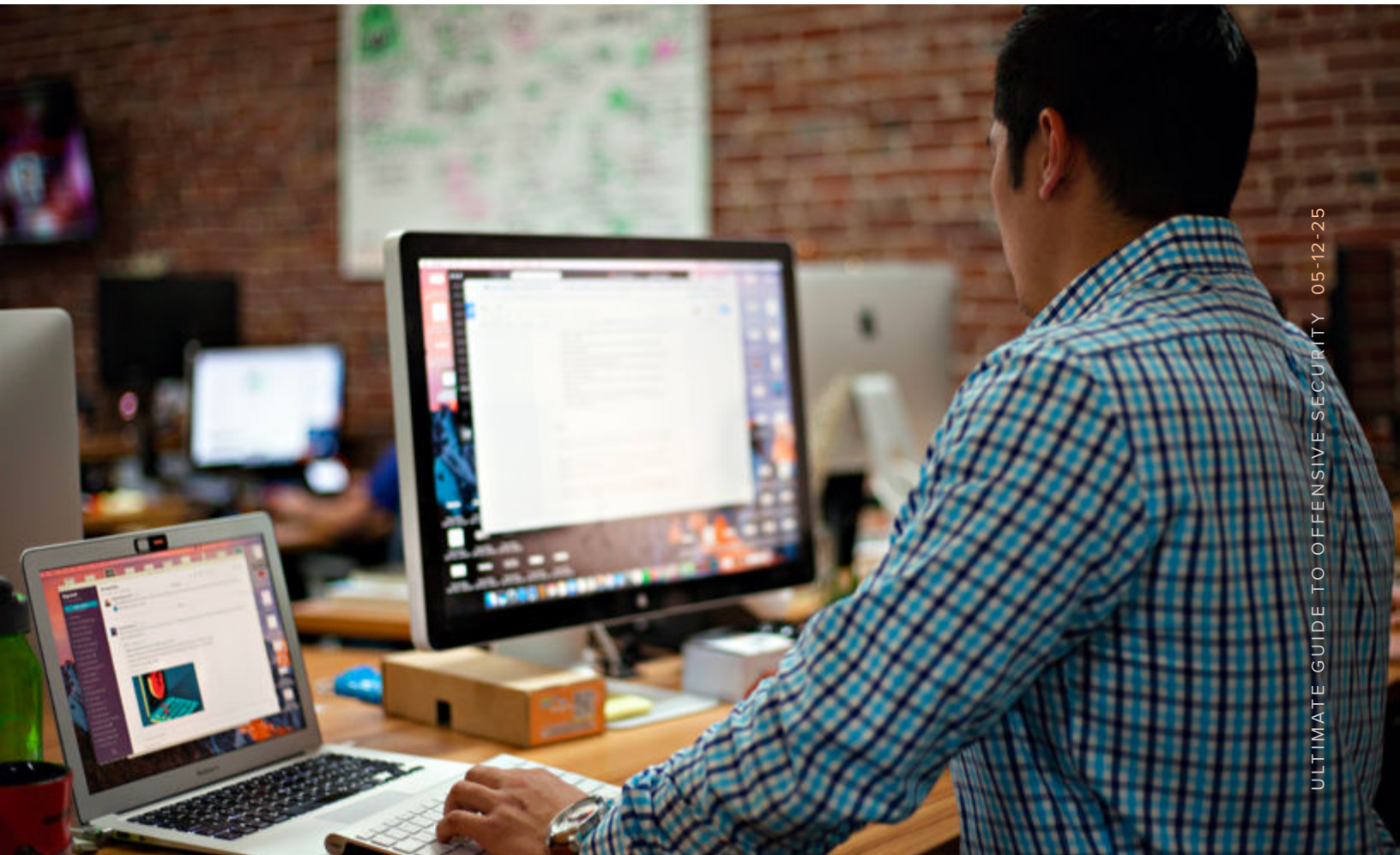
CROWDSOURCED RED TEAMING

Red teaming recently joined the ranks of crowdsourced offensive security methods. Previously, red teams were internal teams or teams from consultancies. Now, the global hacker talent of the Crowd enables a new delivery method: red teaming as a service via crowdsourced red teams.

Crowdsourced red teams bring the same benefits of other crowdsourced offensive security methods: flexibility to your systems, security posture, and needs. Companies can work with one crowdsourced red team for an extended period of time, or they can work with multiple red teams in a setup similar to a bug bounty (they can also set up a model in between these two).

Combined with the usual power of red teams—close emulation of threat actors—crowdsourced red teams can help companies identify detection and response gaps and improve incident response without toiling over finding the right red team.

Crowdsourced red teaming can be the right move for companies (over VDPs, bug bounties, and pen testing) if they are looking for the most thorough test of their systems and if they want to increase the maturity of their security posture as much as possible.



Crowdsourced testing

Crowdsourced testing is the process of contracting expert hackers to test your systems. Its main benefits are **breadth and scalability**.

You can hire experts in many different security specialties, all with their own TTPs. By doing so, you're more likely to find exploits in your systems.

This information will then make your systems more resistant to a wide breadth of methods. Crowdsourced testing also scales well; it requires fewer resources and much less time, as it is **significantly less costly** to pay hackers per finding than hiring a full-time employee.

Additionally, the incentive structure of crowdsourced testing prioritizes speed and critical vulnerabilities. The first hacker to find a specific vulnerability earns the reward associated with a vulnerability. On top of that, P1 vulnerabilities pay out more than P2s, and so on. In the realm of crowdsourced testing, there are three main offensive security methods: vulnerability disclosure programs (VDPs), bug bounties, and penetration testing. We'll discuss each in turn.

VULNERABILITY DISCLOSURE PROGRAMS

VDPs are a secure way to engage external hackers in identifying vulnerabilities in a company's systems. Companies set up their VDPs to make it safe for external hackers to report to the companies any vulnerabilities they find. Companies can then report these findings to other companies and fix the underlying vulnerability as well. VDPs signal to both threat actors and a company's customers that the company takes security seriously and therefore will be more difficult (although never impossible) to exploit.

VDPs are a **low-pressure** way to get started with offensive security because for companies, the process is mostly passive. Once a company has done the upfront work of setting up the VDP, hackers may find vulnerabilities of their own volition, with no contracting needed.

BUG BOUNTIES

Bug bounties are similar to VDPs but go one step further—they offer monetary rewards to hackers who find vulnerabilities. The first hacker to discover a vulnerability receives a bounty, and different vulnerability levels are associated with different reward amounts.

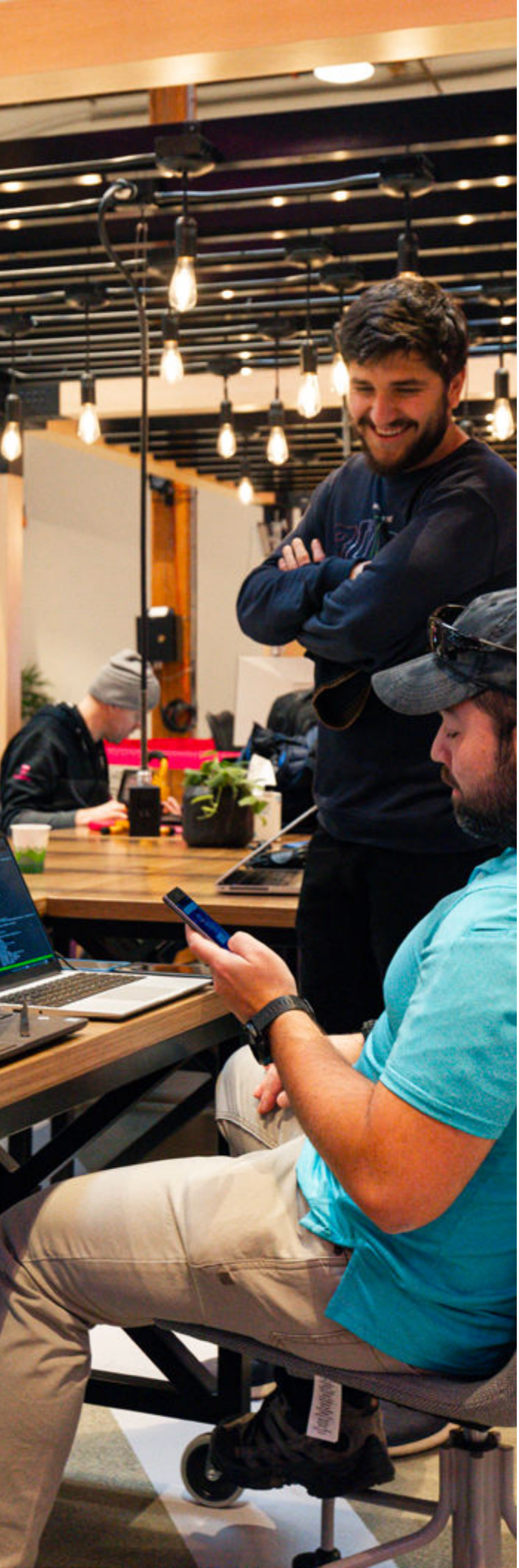
Bug bounties also differ from VDPs in that many bug bounties initially have a defined scope. The scope outlines what parts of a company's system are eligible for testing and what kind of vulnerabilities companies are looking for. Any discovered vulnerabilities outside of this scope aren't rewarded. The benefit of starting with a defined scope is that it can ease internal adoption initially and create an opportunity for learning. The downside is that the full attack surface won't be considered—so most organizations either evolve toward the best practice of an open scope over time or run multiple engagements for specific assets.

DEFINING THE SCOPE

Hackers are incentivized to report (and get rewarded for) what is in scope. What's out of scope is off limits, meaning no compensation is awarded for findings in out-of-scope targets. Organizations choose between three main types of scope—limited scope, wide scope, and open scope.

Inside the Platform: Bugcrowd's Vulnerability Trends Report found that programs with an open scope average 10x more critical vulnerability submissions.

Bug bounties are a common next step for companies who may already have a VDP.



PENETRATION TESTING

Penetration testing (or pen testing) involves hiring a hacker to simulate attacks against a company's systems. Pen testing differs from bug bounties and VDPs in that it is active. In pen testing, a company pays specific hackers to attack its systems, often based on an industry-standard methodology.

Since it's an **active process**, pen testing is the best option when a company needs specific results in a defined timeframe, such as to ensure compliance with internal or external controls. Often, pen testing shows the best results when a company has a defined scope and is able to hire hackers that have the necessary skills for that scope. Additionally, pen testing usually ends with a detailed report of any vulnerabilities and potential patches, something that is not guaranteed with VDPs and bug bounties.

Pen testing requires more work to set up and costs more than bug bounties and VDPs. However, pen testing can also ensure that organizations meet compliance needs and requirements, so it's a crucial part of an offensive security strategy.

Common tools

Hackers use many tools to perform a wide range of activities in offensive security, from vulnerability scanning and network traffic detection to penetration testing. When setting up your own offensive security practices, you'll want to be familiar with the most common tools and methods.

METASPLOIT

Metasploit is both a framework and a tool used in penetration testing. Hackers can use Metasploit to develop and test exploit code against remote machines. The Metasploit framework is flexible, allowing users to create custom modules for attacks. Metasploit is so important in the industry that one blogger coined a law about it: "Casual Attacker power grows at the rate of Metasploit" (HD Moore's Law). Metasploit also has a robust open source community supporting the framework.

metasploit®

ZAP

OWASP's Zed Attack Proxy (ZAP) is the most popular web security scanner. ZAP is a "man-in-the-middle proxy" between servers and browsers. It automatically intercepts requests and responses to find both malicious requests and vulnerabilities. It can also send requests to the server to probe for further vulnerabilities. ZAP is completely open source as well.

NMAP

Nmap is a network scanning tool. With Nmap, hackers can find all the hosts and services on a network, see which ports are open (and which services are using them), and determine the operating systems of hosts. Hackers can also use Nmap's scripting engine to automate scanning tasks. Nmap is free and open source.

BURP SUITE

Burp Suite is a collection of tools that support end-to-end offensive security practices for web applications, from indexing (Burp Spider) and scanning (Burp Scanner) to proxies (Burp Proxy) and attacks (Burp Intruder). There are 20 different tools available across different product tiers (including a basic free tier). Users can also download extensions created by other Burp Suite users on the BApp Store.

Burp Suite

Hackers use a wide variety of tools in pursuit of offensive security, but the ones listed above are the **must-know tools.**

Hack the Pentagon

An example of offensive security

Offensive security practices are more than theory, of course. Organizations worldwide use them to great effect. Consider one famous example. In 2016, the US Department of Defense (DoD) started a new program: Hack the Pentagon. It was a bug bounty program that invited select hackers to probe for exploits in the Pentagon's public-facing websites. Significantly, it was the first bug bounty ever undertaken by any branch of the US government. In 24 days, hackers found 138 different vulnerabilities. Afterward, the DoD set up a VDP so hackers could report vulnerabilities as they came across them, resulting in more than 3,000 unique vulnerabilities submitted by over 650 hackers.

The impact has clearly been positive. In the years since the initial bug bounty, the DoD has run 40 bug bounties (such as Hack the Air Force, which was run by Bugcrowd). Across these programs, more than 1,600 hackers have found at least 2,100 vulnerabilities. Vulnerabilities have been found in HVAC systems, Pentagon networks, and even military fighter jets.

Although the first bug bounty took internal effort to deploy, the culture within the DoD has now shifted to embrace offensive security. The bug bounties also helped them recover from hacks in prior years (such as the Office of Personnel Management hack and the 2015 email hack).

Objections to offensive security

So, offensive security works. The big question left is: **How can you implement offensive security in your company?**

As we mentioned before, there are some obstacles that make it a bit harder to implement offensive security than reactive/defensive security. We'll go through each obstacle to offensive security one by one and talk through ways to overcome them.

Effectiveness of offensive security

The numbers speak for themselves. The DoD found over 2,100 vulnerabilities through their bug bounties and VDPs over the last few years. Another example is from the Cybersecurity and Infrastructure Security Agency (CISA). CISA mandated VDPs for 40+ federal organizations, including NASA, Homeland Security, and the Department of the Treasury (all of which hosted their VDPs with Bugcrowd). In 2022 alone, hackers found 1,330 vulnerabilities via these VDPs. 274 of these vulnerabilities were classified as severe, and 84% were thereafter remediated.

Furthermore, bug bounties on Bugcrowd have shown a **240% ROI**.

Another point to consider is missed opportunities. In a survey we ran, 58% of hackers chose not to disclose a vulnerability they had discovered because the company didn't have a way for them to report it without legal consequences. The takeaway is that offensive security may already be working for your company, but there's just a small obstacle in the way of seeing the results.

Resources required for offensive security

In a world of decreasing security resources, it can be hard to **justify new security approaches** when the existing backlog of unpatched vulnerabilities is continuing to grow. It's hard enough to patch existing vulnerabilities and set up defensive measures. Thankfully, there are low-cost ways to start with offensive security, namely VDPs.

VDPs require a small amount of upfront effort to set up. Our Ultimate Guide to Vulnerability Disclosure delves into the details, but with Bugcrowd, the effort can be minimized even further. The Bugcrowd platform makes it easy to document the principles, scope, and intake method of your VDP.

Additionally, Bugcrowd sources, triages, and sends reported vulnerabilities to you to then remediate.

Another way offensive security may not match company resources is that a company's employees may not be familiar with threat actors' TTPs. Crowdsourced testing as a whole is the solution to this problem. For example, hiring a pen tester makes it easy to acquire the specific skills required to run a test. The alternative, hiring people to supplement your team, would take significantly more time and money. Hiring full-time team members also may not guarantee that you have the full breadth of skills necessary to test all your systems.

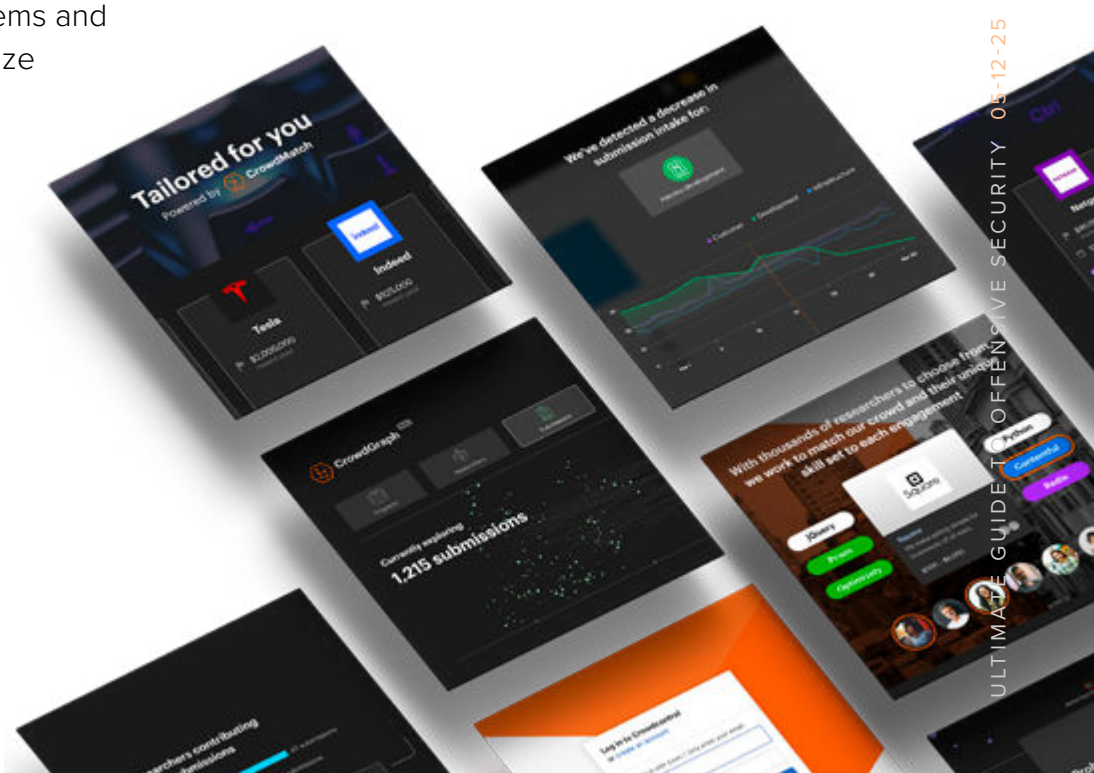
Bugcrowd as an offensive security tool

Let's say we have convinced you of the importance of offensive security and that you have the resources to spin up an offensive security program. The last obstacle you might be figuring out is how to actually get started. What's the **first step** you should take to set up your program?

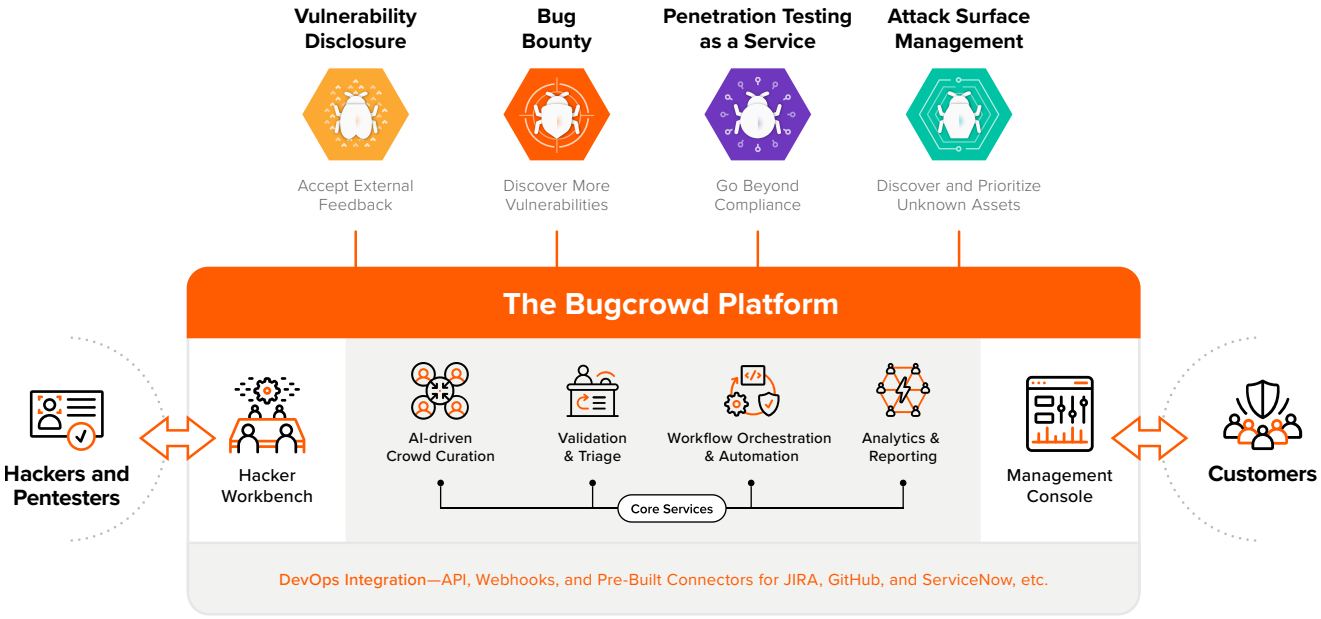
is how to actually get started. What's the first step you should take to set up your program? Bugcrowd makes that first step easy. We work with you to define the attack surface of your system and prioritize the components for testing.

With all this in place, you could define your VDP in a day. For bug bounties and VDPs, we find hackers with skills that exactly match your needs. For Red Team as a Service, we find hacker red teams that match your systems and skills needs. We also prioritize reported vulnerabilities (according to our Vulnerability Rating Taxonomy) and provide recommendations to ensure any found exploits are patched quickly.

Once your crowdsourced offensive security program has been set up, your organization becomes far more adaptable to security threats. You'll be able to find vulnerabilities and create defensive measures to patch them. You can then test the effectiveness of your defensive measures and find any new vulnerabilities. **This cycle can be repeated continuously, giving you far better protection than point-in-time security measures.**



As technology cycles come and go, your offensive security program will help you adapt as quickly as threat actors and hackers do, letting you stay effectively one step ahead.



✓ Best Security ROI from The Crowd

We match you with the right trusted hackers for your needs and environment across hundreds of dimensions using AI.

✓ Instant Focus on Critical Issues

Working as an extension of the platform, our global security engineer team rapidly validates and triages submissions, with P1s (critical vulnerabilities) often handled within hours.

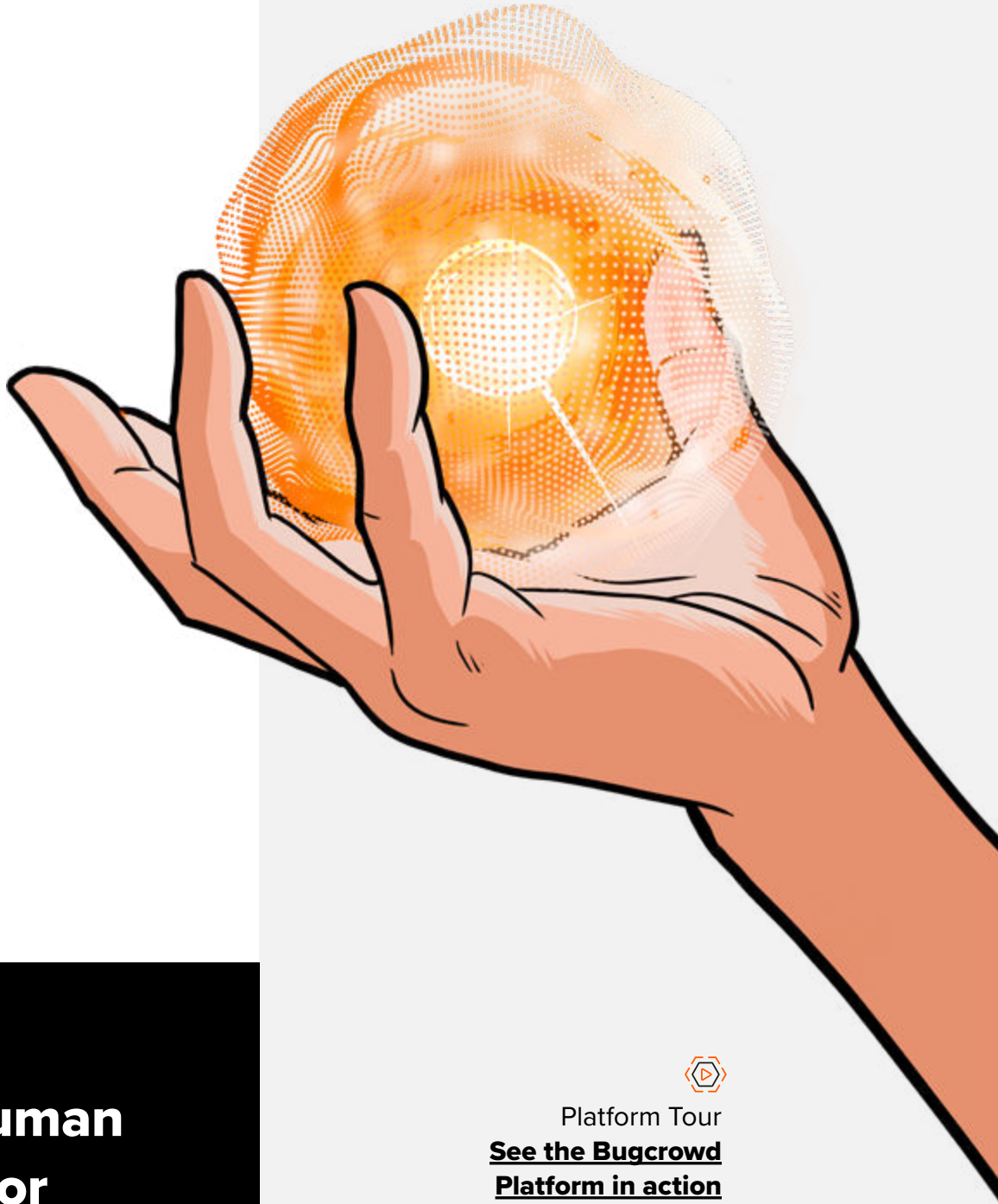
✓ Continuous, Resilient Security for DevOps

The platform integrates workflows with existing tools and processes to ensure that applications and APIs are continuously tested before they ship.

✓ Contextual Intelligence for Best Results

We apply accumulated knowledge from over a decade of experience crafting thousands of customer solutions to your goals for better outcomes.

bugcrowd



Unleash human creativity for proactive security

Try Bugcrowd

Platform Tour
See the Bugcrowd Platform in action

Data Sheet
10 Essentials to Look for in a Crowdsourced Security Platform Checklist