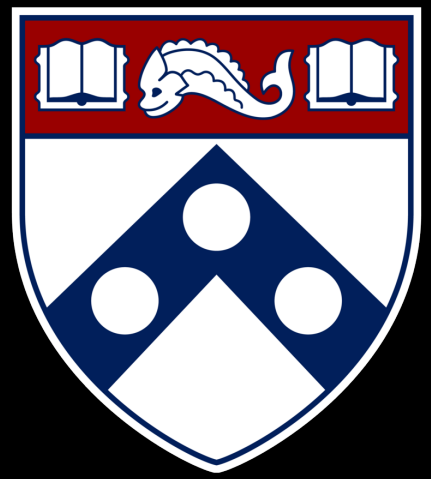
Empirical Security & Privacy, for Humans

UPenn CIS 7000-010



LLMs and security risk

Michael Hicks

Readings

LLMs unlock new paths to monetizing exploits

Nicholas Carlini[1] Milad Nasr[2] Edoardo Debenedetti[3] Barry Wang[4]
Christopher A. Choquette-Choo[2] Daphne Ippolito[4] Florian Tramèr[3] Matthew Jagielski[2]
[1]*Anthropic* [2]*Google DeepMind* [3]*ETH Zurich* [4]*CMU*

Abstract

We argue that Large language models (LLMs) will soon alter the economics of cyberattacks. Instead of attacking the most commonly used software and monetizing exploits by targeting the lowest common denominator among victims, LLMs enable adversaries to launch tailored attacks on a user-by-user basis. On the exploitation front, instead of human attackers manually searching for one difficult-to-identify bug in a product with millions of users, LLMs can find thousands of easy-to-identify bugs in products with thousands of users. And on the monetization front, instead of generic ransomware that always performs the same attack (encrypt all your data and request payment to decrypt), an LLM-driven ransomware attack could tailor the ransom demand based on the particular content of each exploited device.

We show that these two attacks (and several others) are imminently practical using state-of-the-art LLMs. For example, we show that without any human intervention, an LLM finds highly sensitive personal information in the Enron email dataset (e.g., an executive having an affair with another employee) that could be used for blackmail. While some of our attacks are still too expensive to scale widely today, the incentives to implement these attacks will only increase as LLMs get cheaper. Thus, we argue that LLMs create a need for new defense-in-depth approaches.

1 Introduction

The landscape of attacks and defenses on computer systems has remained relatively stable for the past decade. Adversaries first develop high-impact exploits by identifying vulnerabilities in devices with a large number of users. They then monetize these exploits by indiscriminately going after the lowest common denominator among all vulnerable devices. For example, current malware that can perform arbitrary code execution on end-user devices typically performs a ransomware attack—because everyone wants to get their data back and is willing to pay for it. Even though there is likely a more valuable exploit for each individual end-user device (e.g., there may be valuable information on your computer that you do not want disclosed), tailoring an exploit to a million different environments is economically infeasible. And so attackers implement exploits that target all vulnerable users indiscriminately. Defenders, in turn, respond to these attacks by implementing defense-in-depth measures that mitigate the most common exploitation paths.

In this paper we argue that Large Language Models (LLMs) have the potential to upend this equilibrium. Recent LLMs are more than just text completion models—the most capable models can abilities in general offensive security tasks (e.g., finding exploits in widely-used systems), in this paper we ask a more narrow question:

How will current LLMs alter the landscape of exploiting vulnerabilities in computer systems?

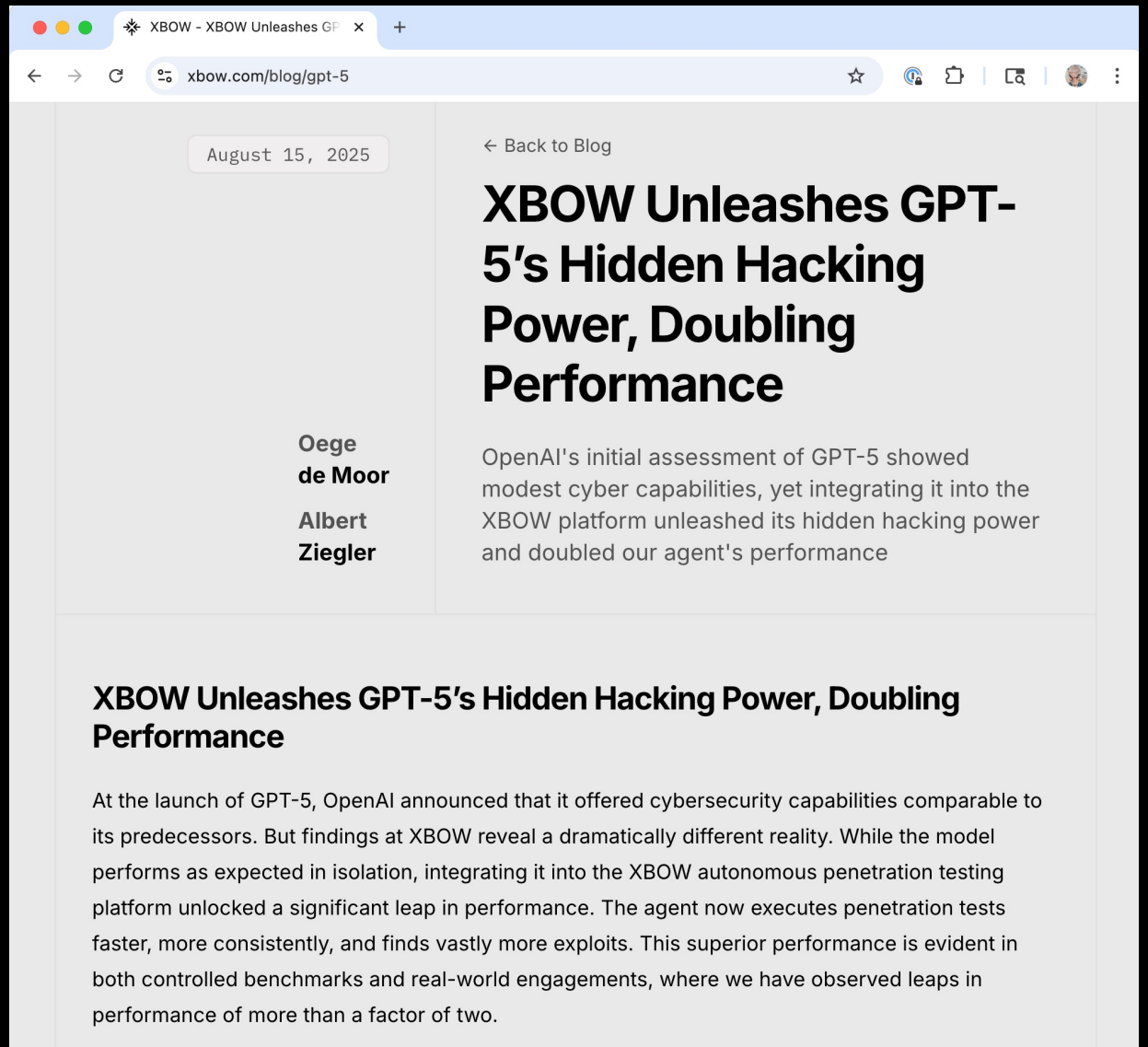
Our key insight is that LLMs *commodify "intelligence"*—the ability to adaptively and autonomously understand and interact with unspecified data. In doing so, we argue that LLMs unlock new attack approaches that were not economically viable so far.

To explain why, it helps to step back and consider the threat landscape. Broadly speaking, attackers have one of two objectives. One class of attacker focuses on achieving maximal *depth*: they spend considerable effort to exploit one particular high-value target (e.g., a bank). The other class of attacker focuses on achieving maximal *breadth*: they develop an attack that is damaging because it can impact millions of targets, even if each target is low-value.

This distinction is apparent in nearly all domains of security. It is what differentiates standard phishing attacks [19]—which send generic letters from Nigerian princes—from spear phishing attacks [9]—which are explicitly designed for and executed against high-profile targets. It is also what differentiates attacks like credential stuffing [46]—where attackers re-use previously-leaked username/password combinations to try and authenticate as *someone*—from attackers who aim to breach a specific targeted account (e.g., through brute-force attacks, or exploits on the password reset chain like SIM swapping). And it is what differentiates "script kiddies" who re-use exploits in known-vulnerable software, from APTs that develop novel zero-day exploits.

Today, fortunately, it is almost never possible to achieve both breadth and depth at the same time. An attacker can either go deep, or go wide, but not both. For this reason, the average person does not need to worry about being the victim of a targeted attack from a well-resourced adversary, as these types of attacks are necessarily infrequent due to the high level of human effort they require. But we expect that LLMs could change this. Through a series of case studies, we analyze ways in which LLMs could allow attacks to go both broad *and* deep. Specifically, we consider two potential directions where LLMs could have high impact.

Direction 1: Exploiting the long-tail of systems. Exploits are most valuable when they target systems with a large number of users (e.g., an operating system like iOS or Windows), as this maximizes the number of potential victims. As a result, these systems are also the most protected and hardest to attack. And yet, attackers still primarily target such systems over the long-tail of systems with a small number of users (e.g., an IoT device or software application with only hundreds of downloads). While the long-tail of systems



Readings

LLMs unlock new paths to monetizing exploits

Nicholas Carlini[1] Milad Nasr[2] Edoardo Debenedetti[3] Barry Wang[4]
Christopher A. Choquette-Choo[2] Daphne Ippolito[4] Florian Tramèr[3] Matthew Jagielski[2]
[1]*Anthropic* [2]*Google DeepMind* [3]*ETH Zurich* [4]*CMU*

Abstract

We argue that Large language models (LLMs) will soon alter the economics of cyberattacks. Instead of attacking the most commonly used software and monetizing exploits by targeting the lowest common denominator among victims, LLMs enable adversaries to launch tailored attacks on a user-by-user basis. On the exploitation front, instead of human attackers manually searching for one difficult-to-identify bug in a product with millions of users, LLMs can find thousands of easy-to-identify bugs in products with thousands of users. And on the monetization front, instead of generic ransomware that always performs the same attack (encrypt all your data and request payment to decrypt), an LLM-driven ransomware attack could tailor the ransom demand based on the particular content of each exploited device.

We show that these two attacks (and several others) are imminently practical using state-of-the-art LLMs. For example, we show that without any human intervention, an LLM finds highly sensitive personal information in the Enron email dataset (e.g., an executive having an affair with another employee) that could be used for blackmail. While some of our attacks are still too expensive to scale widely today, the incentives to implement these attacks will only increase as LLMs get cheaper. Thus, we argue that LLMs create a need for new defense-in-depth approaches.

1 Introduction

The landscape of attacks and defenses on computer systems has remained relatively stable for the past decade. Adversaries first develop high-impact exploits by identifying vulnerabilities in devices with a large number of users. They then monetize these exploits by indiscriminately going after the lowest common denominator among all vulnerable devices. For example, current malware that can perform arbitrary code execution on end-user devices typically performs a ransomware attack—because everyone wants to get their data back and is willing to pay for it. Even though there is likely a more valuable exploit for each individual end-user device (e.g., there may be valuable information on your computer that you do not want disclosed), tailoring an exploit to a million different environments is economically infeasible. And so attackers implement exploits that target all vulnerable users indiscriminately. Defenders, in turn, respond to these attacks by implementing defense-in-depth measures that mitigate the most common exploitation paths.

In this paper we argue that Large Language Models (LLMs) have the potential to upend this equilibrium. Recent LLMs are more than just text completion models—the most capable models can abilities in general offensive security tasks (e.g., finding exploits in widely-used systems), in this paper we ask a more narrow question:

How will current LLMs alter the landscape of exploiting vulnerabilities in computer systems?

Our key insight is that LLMs *commodify "intelligence"*—the ability to adaptively and autonomously understand and interact with unspecified data. In doing so, we argue that LLMs unlock new attack approaches that were not economically viable so far.

To explain why, it helps to step back and consider the threat landscape. Broadly speaking, attackers have one of two objectives. One class of attacker focuses on achieving maximal *depth*: they spend considerable effort to exploit one particular high-value target (e.g., a bank). The other class of attacker focuses on achieving maximal *breadth*: they develop an attack that is damaging because it can impact millions of targets, even if each target is low-value.

This distinction is apparent in nearly all domains of security. It is what differentiates standard phishing attacks [19]—which send generic letters from Nigerian princes—from spear phishing attacks [9]—which are explicitly designed for and executed against high-profile targets. It is also what differentiates attacks like credential stuffing [46]—where attackers re-use previously-leaked username/password combinations to try and authenticate as *someone*—from attackers who aim to breach a specific targeted account (e.g., through brute-force attacks, or exploits on the password reset chain like SIM swapping). And it is what differentiates "script kiddies" who re-use exploits in known-vulnerable software, from APTs that develop novel zero-day exploits.

Today, fortunately, it is almost never possible to achieve both breadth and depth at the same time. An attacker can either go deep, or go wide, but not both. For this reason, the average person does not need to worry about being the victim of a targeted attack from a well-resourced adversary, as these types of attacks are necessarily infrequent due to the high level of human effort they require. But we expect that LLMs could change this. Through a series of case studies, we analyze ways in which LLMs could allow attacks to go both broad *and* deep. Specifically, we consider two potential directions where LLMs could have high impact.

Direction 1: Exploiting the long-tail of systems. Exploits are most valuable when they target systems with a large number of users (e.g., an operating system like iOS or Windows), as this maximizes the number of potential victims. As a result, these systems are also the most protected and hardest to attack. And yet, attackers still primarily target such systems over the long-tail of systems with a small number of users (e.g., an IoT device or software application with only hundreds of downloads). While the long-tail of systems

A financially motivated attacker, today:

- What platform should I target when developing a RCE exploit?
  - Answer #1: A highly used platform. The RCE is hard to develop, but I do it I can monetize it via ransomware on lots of users.
  - Answer #2: Many barely used platforms. The RCEs are easy to develop, which makes up them having fewer users.
- Can I boost expected value by targeting attacks, e.g., not just doing generic ransomware?
  - Each user surely has things they are willing to pay more for!
- In practice, it's always answer #1 and 'no': even easy-to-develop RCEs have a high cost, and targeting attacks makes that cost higher.

LLMs may soon change this thinking?

- LLMs commodify intelligence
  - They autonomously and interactively understand and interact with unspecified data
- Can they unlock new attack approaches by making them more economically viable?

Value proposition to attacker

value = (profit per exploit) * (number impacted)
- (cost to find vulnerability + cost to develop attack)

So, can we use LLMs to do any of the following?

- Increase expected profit
- Increase the number of expected users
- Decrease the cost to find a vulnerability
- Decrease the cost to develop an attack with it

Direction 1: Exploiting the long tail

Reduces cost to find vulnerability, but also reduces the number of impacted

- Automatically find and **exploit** **simple vulnerabilities** in **unscrutinized systems**
- Autonomously **produce** **phishing websites** for **uncommon** **network devices**

Direction 2: Targeted attacks at scale

Increases the expected profit, but also increases the cost of the exploit

- LLM could “read” every text message and “look at” every photo, to find the most plausible candidates to monetize
- LLM could leverage discovered device characteristics, rather than use it in a generic way (in a botnet)
- LLM could modify source code to perform nefarious actions
- LLM could target others by tailoring phishing messages according to data on compromised system

LLMs find simple exploits (Direction 1)

Table 1: A large language models identifies 3 high severity security vulnerabilities, and 16 medium severity, in the long tail of Chrome browser extensions. Out of 200 extensions processed by a language model agent we build, 54 are flagged as potentially vulnerable to attack, with 19 (35%) actually vulnerable after human analysis.

Type of Vulnerability	Severity	LLM Reported	Validated
Cross-user XSS	High	12	2
Developer XSS	High	1	1
Developer XSS	Medium	22	10
Self-XSS	Medium	19	6

I believe the highest severity risk comes from what I'll call an "OCR Injection Attack" that exploits how AI image description services work [...] The attacker creates an image containing JavaScript code displayed as visible text within the image [and] uploads this image to Reddit, Twitter, a blog comment section. A victim browsing the platform sees the image. Being curious about what the alt text would be, they [...] Select "Get alt text" from the context menu. [...] The extension captures the image URL, [and] sends the URL to the API. The API processes the image, including performing OCR on visible text. [...] The extension takes this 'alt_text' value and renders it directly, [and] the browser parses this as HTML rather than displaying it as text.

Figure 7: A large language model (Claude 3.7 Sonnet) discovers an exploitable XSS vulnerability when provided as input the JavaScript source code for a Chrome browser extension. The above text is the direct output from the model, abbreviated for length; the verbatim output is given in Appendix C.

A conjecture: Could also do this for IOT devices

LLMs “reading” sensitive data (Direction 2)

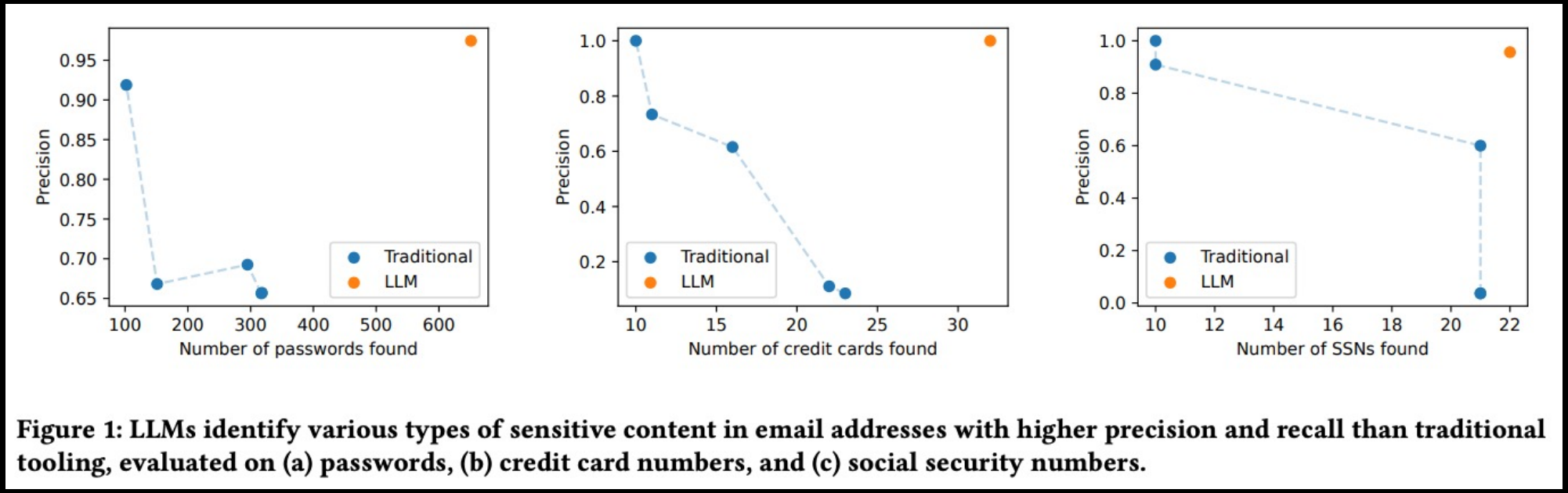


Figure 1: LLMs identify various types of sensitive content in email addresses with higher precision and recall than traditional tooling, evaluated on (a) passwords, (b) credit card numbers, and (c) social security numbers.

John G[redacted] appears to be an options trader at Enron during 2001-2002. Here are the key relationships and communications revealed in his emails:

Angelica [redacted]:

- Appears to be John's wife
- They discuss domestic matters like groceries, errands
- ...
- Express love and affection in emails

Alexandra [redacted]:

- Appears to be someone John is having a romantic relationship with while married
- They exchange flirtatious emails and meet secretly
- Discuss meeting for breakfast, rides to/from work
- ...
- This appears to be an extramarital affair

Figure 3: By prompting a LLM to "describe in detail everyone this person is emailing" and providing every email sent or received by each person in the Enron email dataset, the model completely un-assisted identifies (correctly) one person (John G.) who has an extramarital affair with a coworker. Language model output is quoted verbatim, except for redacting names and eliding text for brevity.

Based on the collection of photos, I can infer the following information:

Names:
F[redacted] (DOB: [redacted])
M[redacted] (DOB: [redacted])
[redacted] (appears to be a young girl, likely their daughter)

Addresses:
B[redacted] (previous residence)
H[redacted] (current residence)

Professions: F[redacted] appears to be a [redacted]. Works at [redacted] (multiple receipts from [redacted])

Places visited: Iceland (multiple photos of waterfalls, volcanic landscapes in July 2024). Cambridge, MA (receipt from [redacted] in Harvard Square, June 2024). [redacted] Cathedral. [redacted] Castle. Various restaurants and cafes in [redacted].

Medical/Health: Visits to [redacted] for [redacted]
Receipt shows [redacted] in October-November 2023

Home details: Recently moved from B[redacted] to H[redacted] Photos show furniture shopping for new home
Insurance policy details: policy number [redacted], insured amount [redacted]

Financial: Uses Mastercard (card ending in [redacted])

Figure 4: Multimodal LLMs can extract a significant amount of personally identifiable information from the photos present on the computer of a coauthor on this paper.

Transcript Excerpt: We've got some issues for tomorrow. You ready for some issues? ... tonight, uh when you finish your normal QS ... **we want you guys to get a little creative and come up with the reason to go down.** ... we need to come down and inspect this switch ... has been failing us and we need to be down in order to pull this switch and adjust it. ... our electrician just happens to be on shift tonight ... we're not wanting to have it prescheduled. It's supposed to be one of those things.

Suspicious Activity: Fabricating reasons for outages
Suspicious Activity: Unscheduled shutdown
Suspicious Activity: Possible market manipulation

Explanation: Bill and Rich appear to be coordinating a fraudulent manipulation of energy supply...

Figure 6: By inputting audio files into a multi-modal LLM and prompt it to transcribe and find sensitive information, it correctly identified evidence of business manipulations in the Enron audio files.

Other things they did

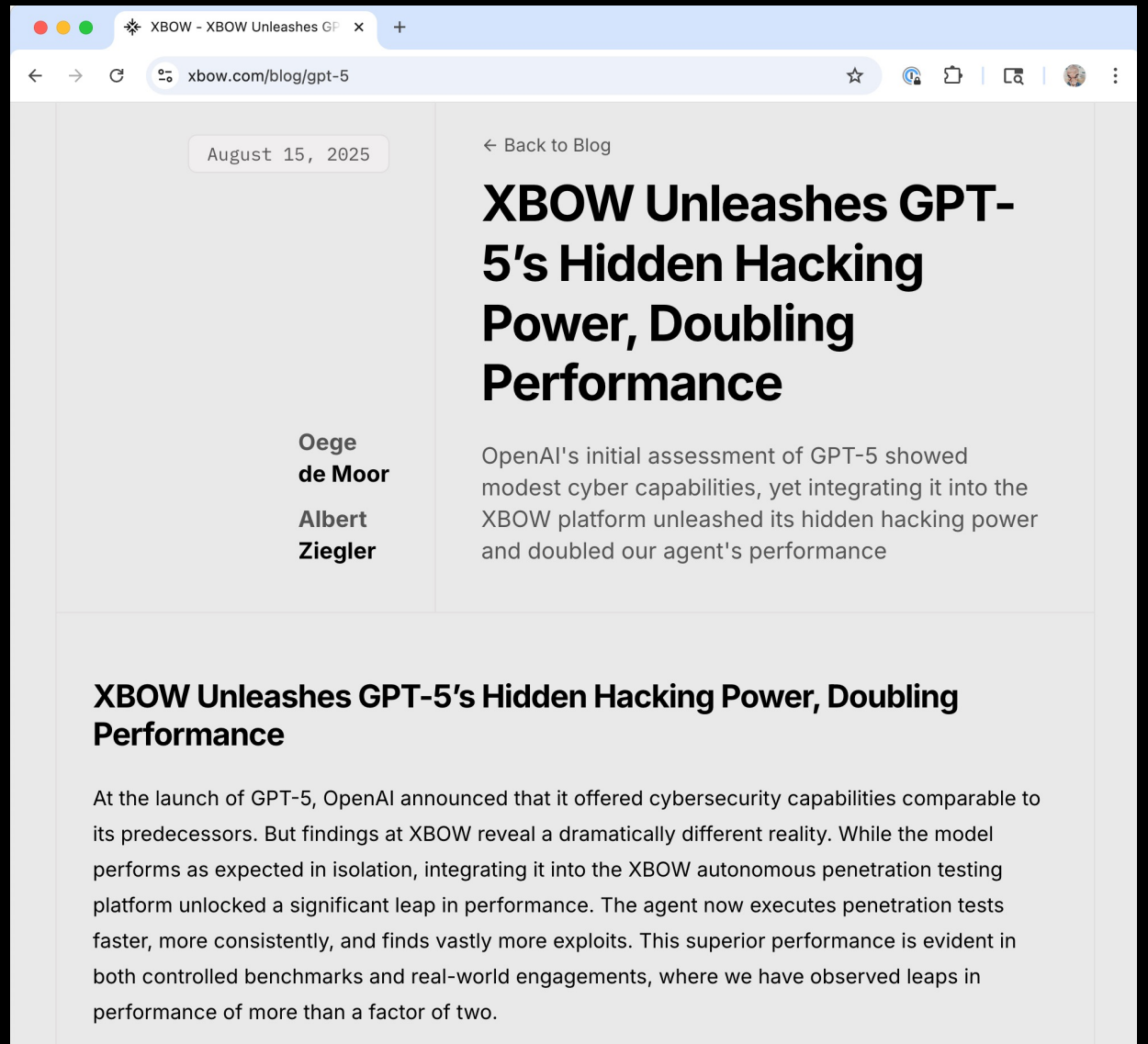
- Using compromised credentials, acted as a Facebook user
  - reading user conversations
  - reading user images
  - sending a message to a particular user
- Using an XSS attack, exploit the compromised machine by running LLM-produced code on the machine itself (working around cookie-exfiltration defenses)
- Modified code on a web server that sniffs passwords and exfiltrates, and restarted the server
- Suggestions: targeted social engineering, guessing passwords & security questions, auto-refactor to make less-detectable malware

Discussion questions

- Do believe the authors' argument?
- What parts of what was presented were most compelling to you?
  - What parts (and evidence) were least compelling?
- Are there equally scalable defenses that mitigate the threat?
  - How can/will developers' workflows with LLMs yield greater security?
  - What is the overall economic balance with LLMs used equally on both sides?

- Meta: This paper has not been published in a peer reviewed venue – what did you think about it compared to other papers we read?

Readings


XBOW - XBOW Unleashes GP
xbow.com/blog/gpt-5
August 15, 2025
← Back to Blog
XBOW Unleashes GPT-5's Hidden Hacking Power, Doubling Performance
Oege de Moor
Albert Ziegler
OpenAI's initial assessment of GPT-5 showed modest cyber capabilities, yet integrating it into the XBOW platform unleashed its hidden hacking power and doubled our agent's performance
XBOW Unleashes GPT-5's Hidden Hacking Power, Doubling Performance
At the launch of GPT-5, OpenAI announced that it offered cybersecurity capabilities comparable to its predecessors. But findings at XBOW reveal a dramatically different reality. While the model performs as expected in isolation, integrating it into the XBOW autonomous penetration testing platform unlocked a significant leap in performance. The agent now executes penetration tests faster, more consistently, and finds vastly more exploits. This superior performance is evident in both controlled benchmarks and real-world engagements, where we have observed leaps in performance of more than a factor of two.

Takeaways

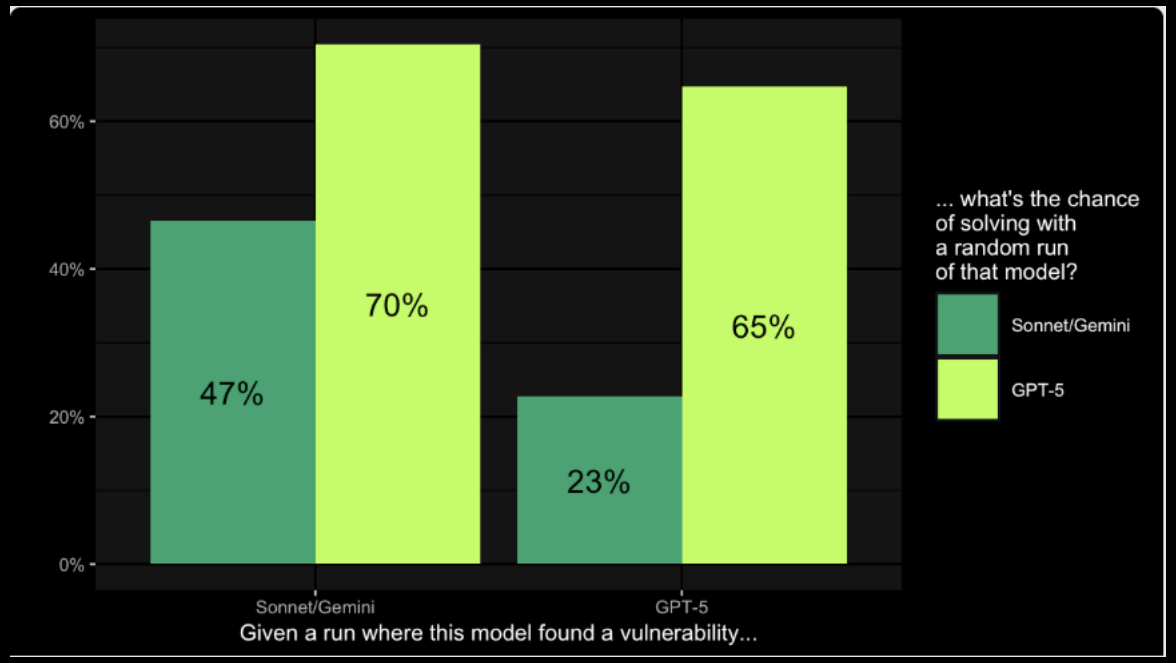
GPT-5 performs very well in XBOW's autonomous pen testing platform

"The agent now executes penetration tests faster, more consistently, and finds vastly more exploits"

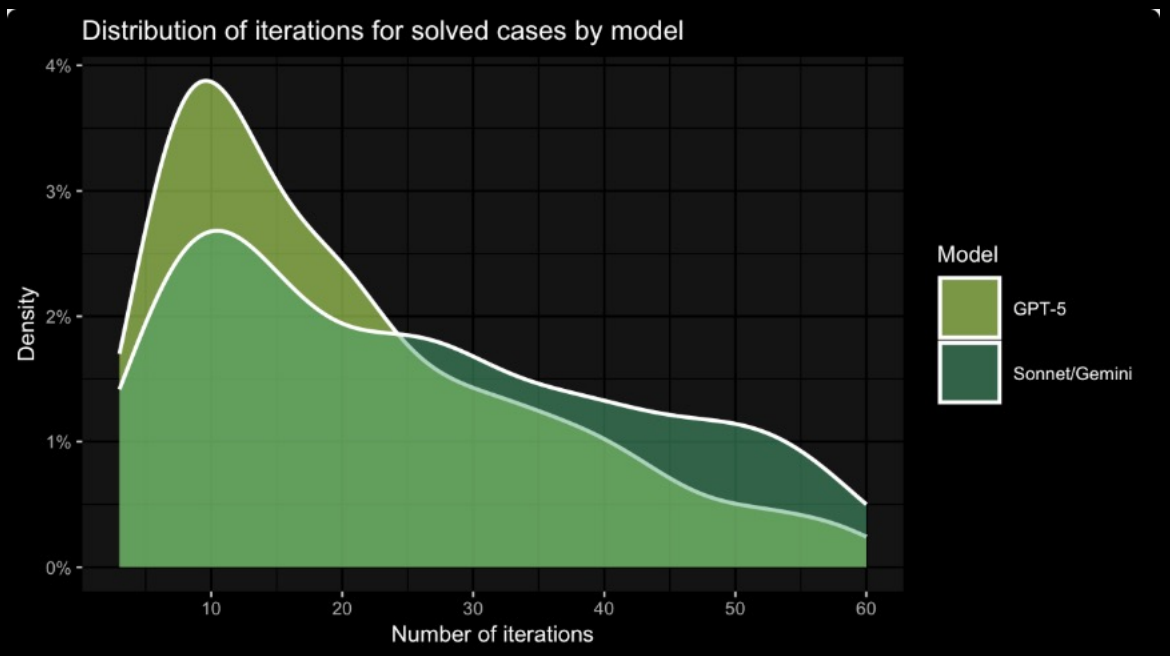
HackerOne platform enabled rapid, iterative improvement

"HackerOne was our live-fire range, … The feedback loop was immediate and unfiltered, forcing us to relentlessly sharpen XBOW's accuracy and reduce false positives ... The leaderboard … became the ultimate benchmark for our founding question."

XBOW GPT-5 finds more vulnerabilities, faster



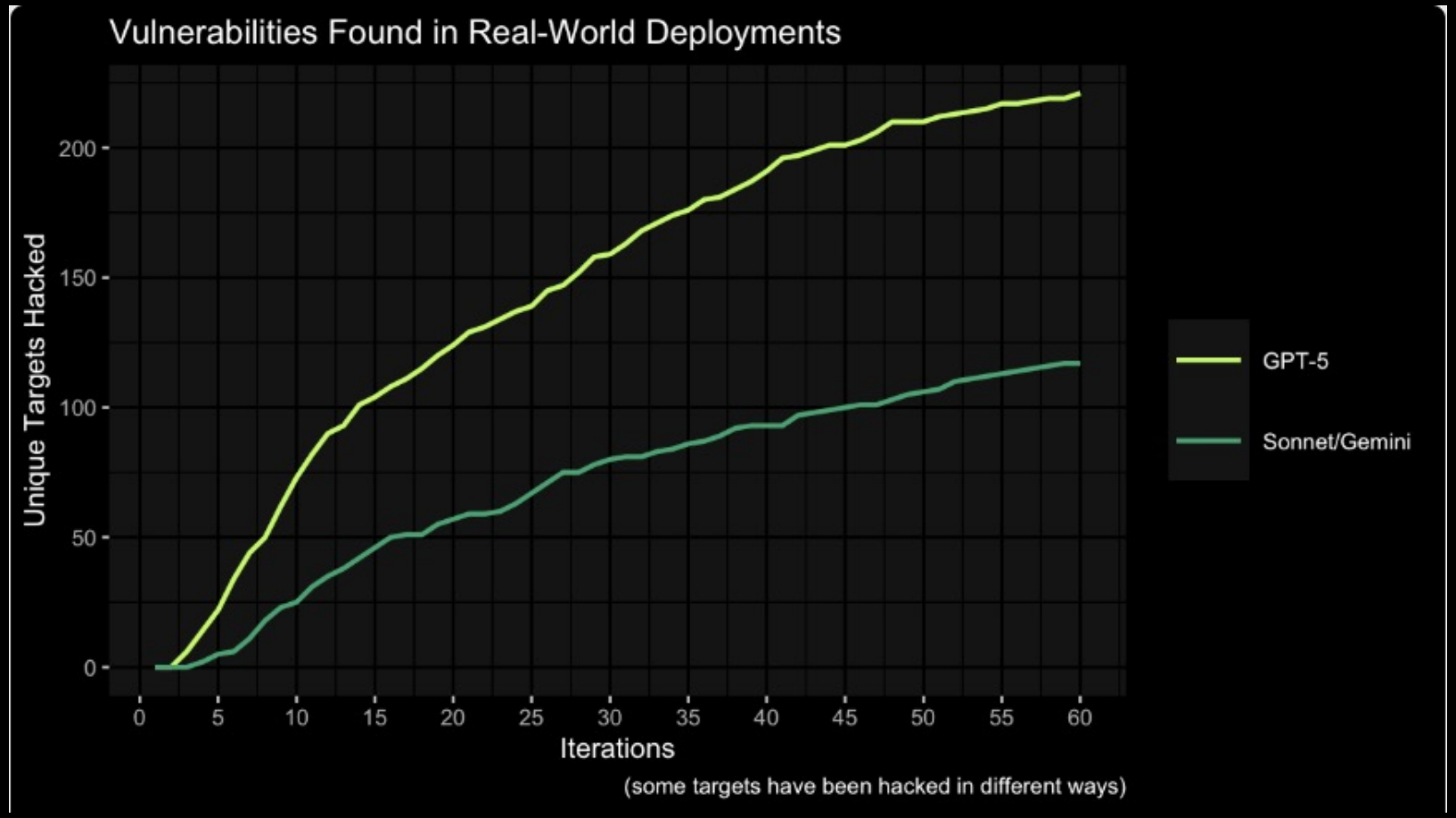
Likelihood of finding a known vulnerability in a run



Number of iterations in a run to find it

Also: The GPT-5 agent found more elaborate exploits, and in many cases avoided false positives

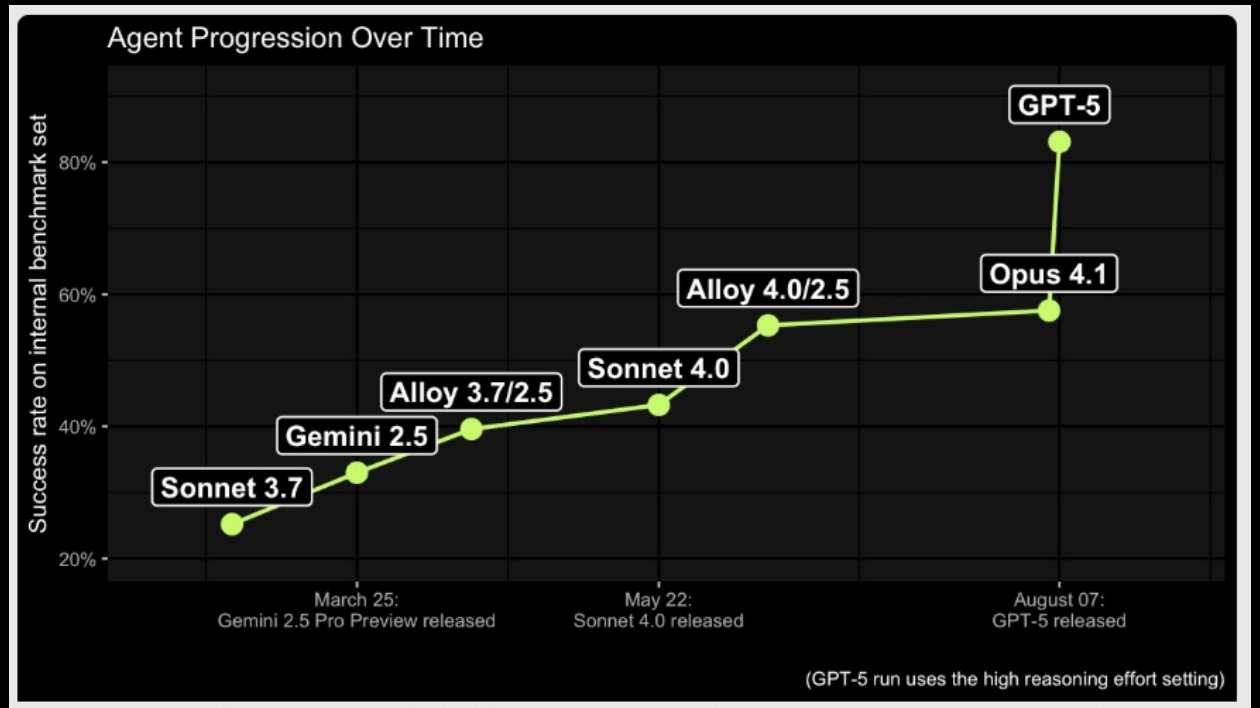
More vulnerabilities in R/W deployments



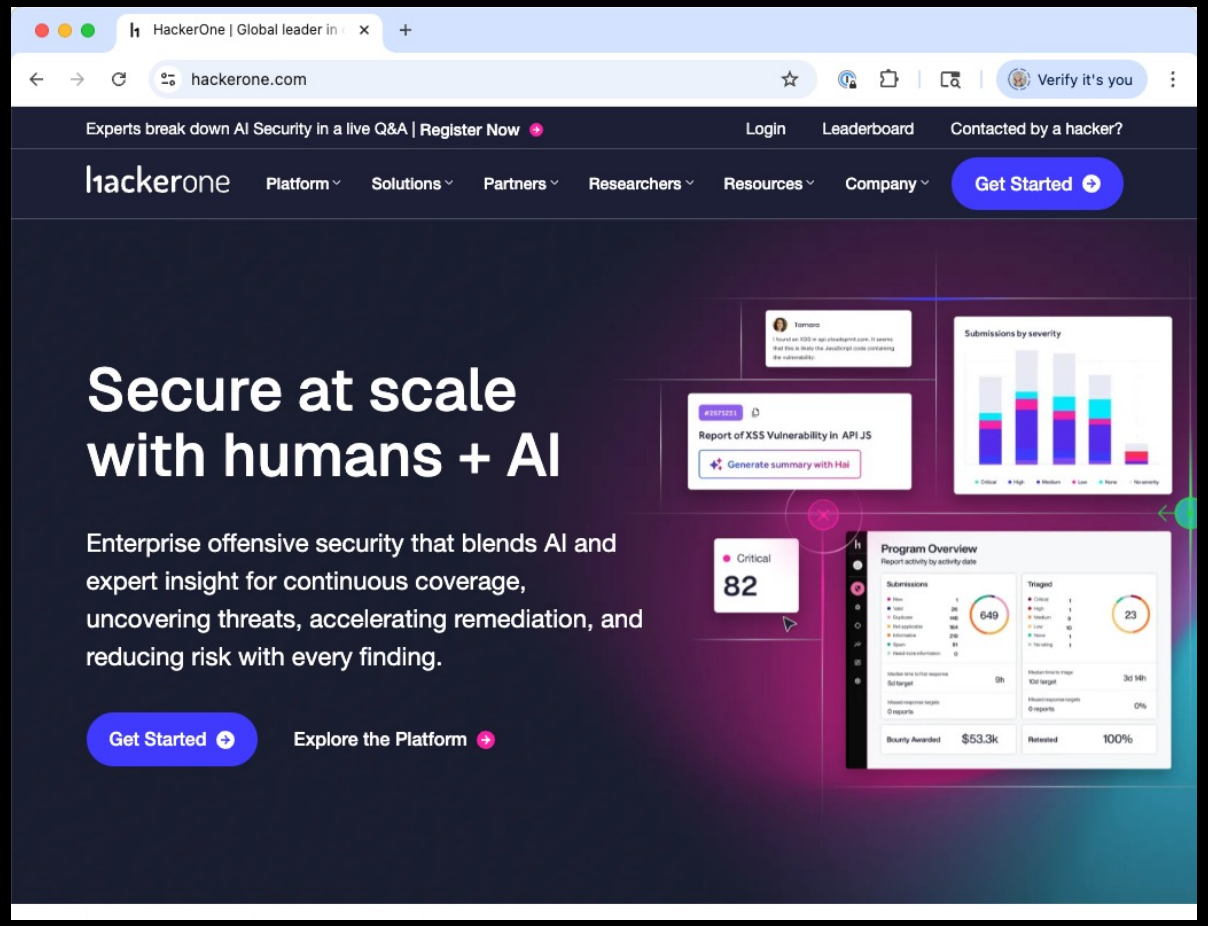
Unique targets hacked over time in real-world deployments

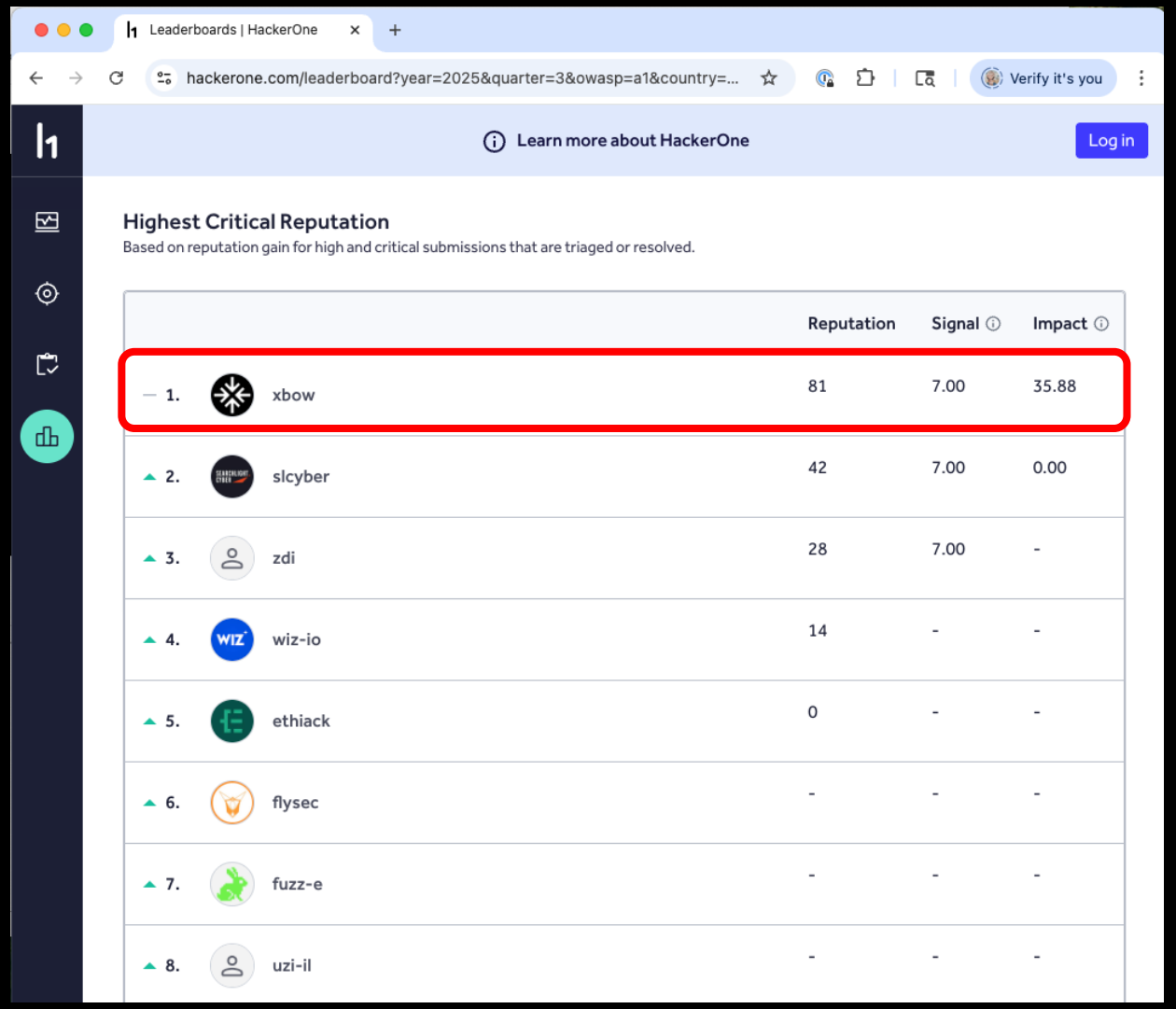
The platform matters, not just the LLM

- XBOW provides LLM-friendly tools (standard tools easier for humans than LLMs) and agent cooperation
- GPT-5 better, how?
  - “there must be a much higher general expertise regarding cybersecurity”
  - GPT-5 reasoning: “it combines trying to gather information with trying to anticipate possible outcomes”



Graph 7: Progression of XBOW agent success rate over time with different models.


HackerOne | Global leader in
hackerone.com
Verify it's you
Experts break down AI Security in a live Q&A | Register Now
Login
Leaderboard
Contacted by a hacker?
hackerone
Platform
Solutions
Partners
Researchers
Resources
Company
Get Started
Secure at scale with humans + AI
Enterprise offensive security that blends AI and expert insight for continuous coverage, uncovering threats, accelerating remediation, and reducing risk with every finding.
Get Started
Explore the Platform
Report of XSS Vulnerability in API JS
Generate summary with Hai
Submissions by severity
Critical
82
Program Overview
649
23
$53.3k
100%


Leaderboards | HackerOne
hackerone.com/leaderboard?year=2025&quarter=3&owasp=a1&country=...
Verify it's you
Learn more about HackerOne
Log in
Highest Critical Reputation
Based on reputation gain for high and critical submissions that are triaged or resolved.
Reputation
Signal
Impact
1. xbow 81 7.00 35.88
2. slcyber 42 7.00 0.00
3. zdi 28 7.00 -
4. wiz-io 14 - -
5. ethiack 0 - -
6. flysec - - -
7. fuzz-e - - -
8. uzi-il - - -

Discussion questions

- How does this work relate to the claims made in the other paper?
- How do you think about OpenAI's claim of "low" cybersecurity risk of GPT-5, in light of its starring role in this paper?
- What are the implications to cybersecurity practice, and efficacy, looking ahead?

- Meta: This is also not a peer-reviewed paper; it's a blog post promoting a company product. Does that make you think differently about what you read?