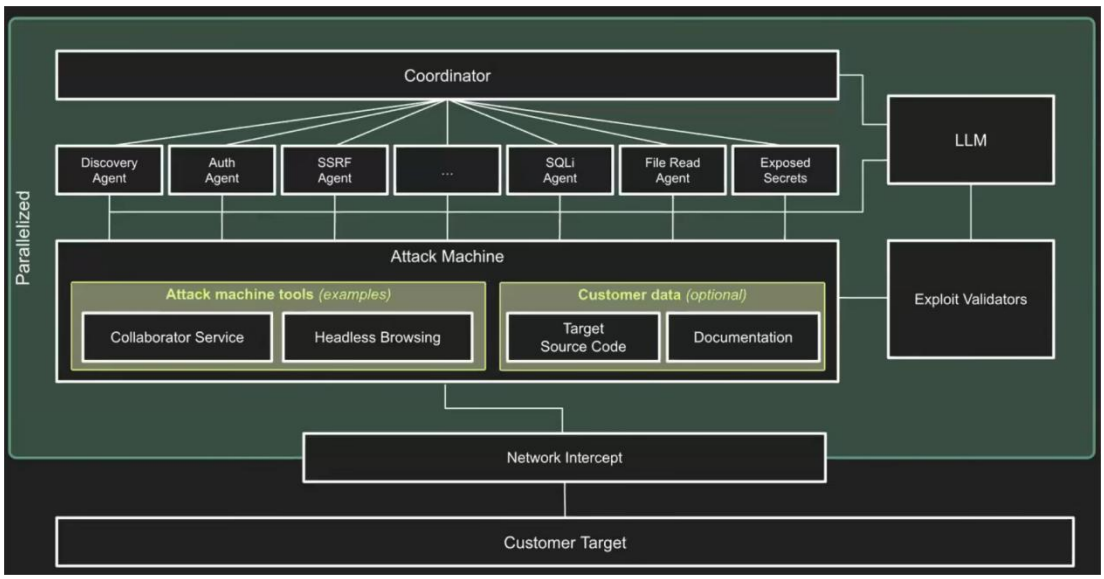
# 1. Recent Summary

Over the past week, I have looked into Xbow's official website and watched 10 YouTube videos. Most of these videos have content from Xbow's team members, and a small number are third-party reviews. Through these materials, I got a clear idea of how Xbow works. I also did a short study on three AI-driven penetration testing projects: PentestGPT, HexStrike and Burp AI. During my research on Xbow, I found many interesting things.

Xbow's official blog has a lot of technical details about different vulnerabilities. But I was especially interested in the product's demo and overall structure. Instead of getting stuck on small technical details, I found a live demo shared by Xbow's Security Lead on August 26, 2025. This demo helped me get a basic understanding of the product's overall structure and key functions.

# 2. Xbow Architecture and Demonstration

As the diagram shows, Xbow works because the following parts work together:




**Coordinator**: It manages the penetration testing process, like a manager in a penetration testing team. The Coordinator sets the testing methods and what's more important, and gives tasks to different Agents.

**Agents**: These are different kinds of AI-based tools that do specific detection and attack tasks. They are made based on "vulnerability primitives" (a technical term for basic vulnerability units).

Example 1: Discovery Agent – Usually, the first step of testing is to map the "attack surface" (all parts of a system that could be attacked). The Coordinator tells the Discovery Agent to list domains, endpoints, parameters and other related things.

Example 2: Authentication Agent – After the discovery step, it tries to log in (if possible). Once

enough information about the attack surface is collected, the Coordinator gives tasks to multiple Agents so they can check for vulnerabilities at the same time.

**Exploit Validators**: When an Agent finds a possible vulnerability, it “raises a flag” (reports it) and sends the result. Then a Validator checks the result again. There are different types of Validators: some use AI, while others use simpler methods. For example, an XSS Validator opens the URL in a “headless browser” (a browser without a visual window) and checks if JavaScript runs. This helps confirm if the vulnerability is real.

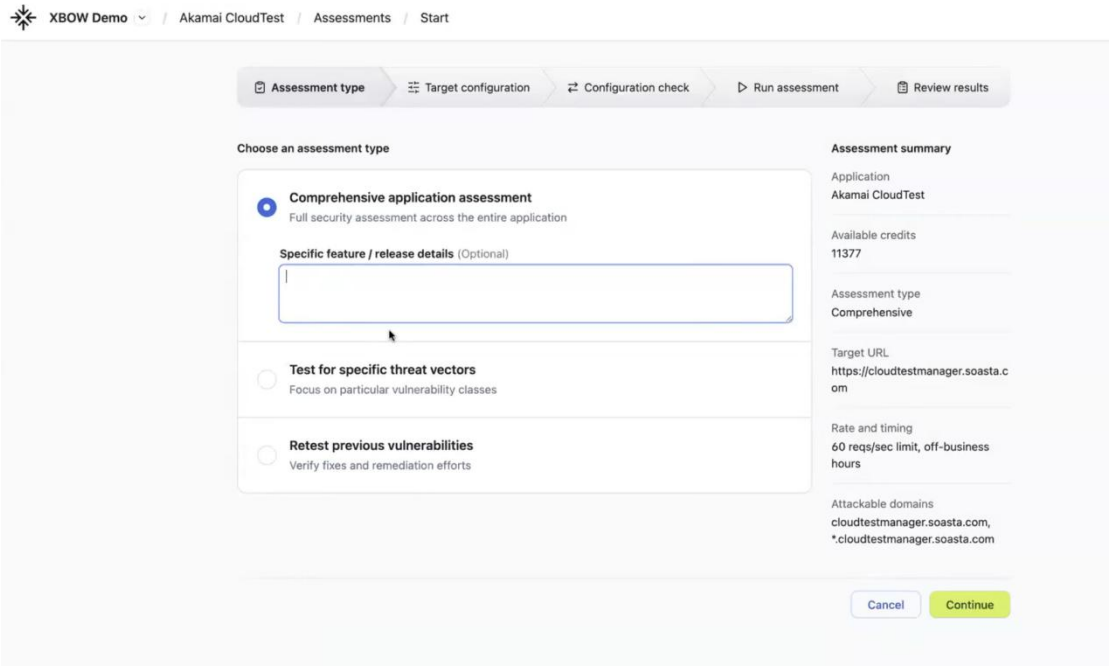
All tasks done by each Agent run in an environment called an “attack machine”. This machine has many tools, including ones made by Xbow and common penetration testing tools. Xbow also added a “collaboration service” and a headless browser to handle complex JavaScript apps. The browser analyzes and interacts with very dynamic frontends (the part of an app users see), which makes attack surface mapping and testing more complete.

The last key part is the Network Intercept module, which has two main jobs:

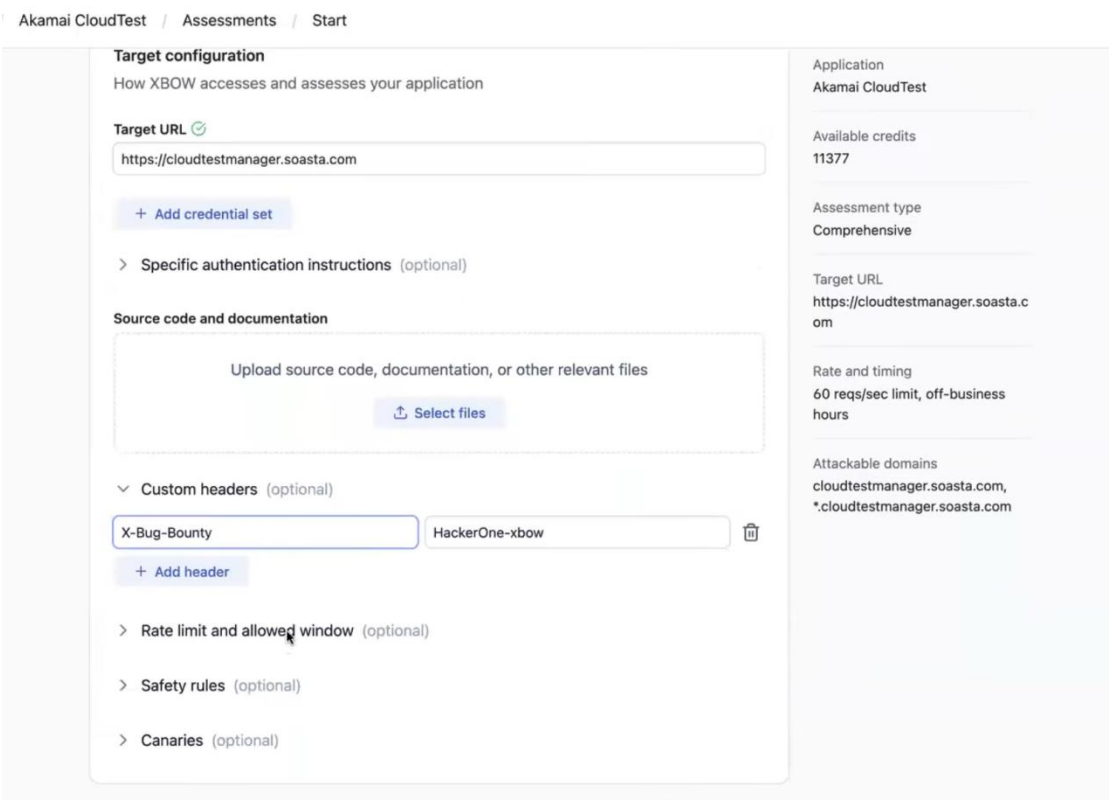
1) It records all requests and responses that Xbow sends. After the penetration test finishes, users get a full record of the network interaction.
2) It works as a protection tool (like a firewall). Users can set rules here to block access to certain domains. This stops Xbow from targeting assets that are not in the test scope (such as production domains). It is one of the many security protections for clients.

In the demo that followed, three types of assessments were shown:

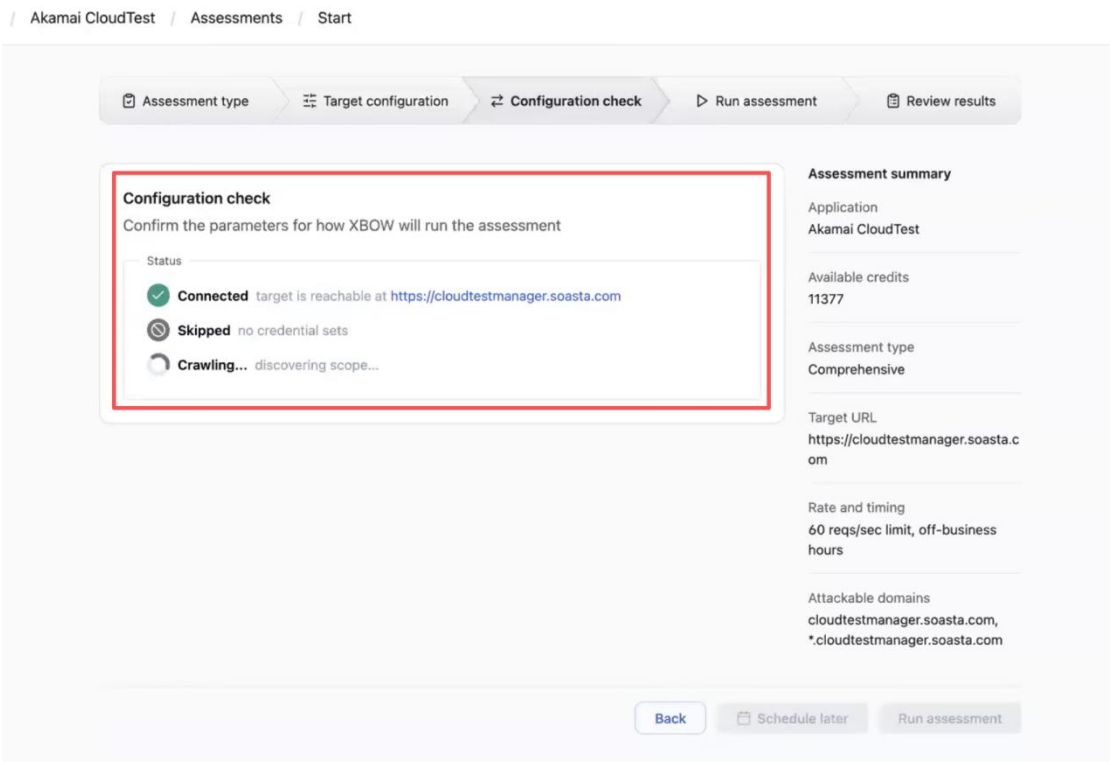
- A full evaluation of the application
- Focused testing on specific threat paths
- Testing again for vulnerabilities found before




Next, users need to set the required target URLs and some optional settings. Optional settings include login details, custom headers, rate limits and so on.

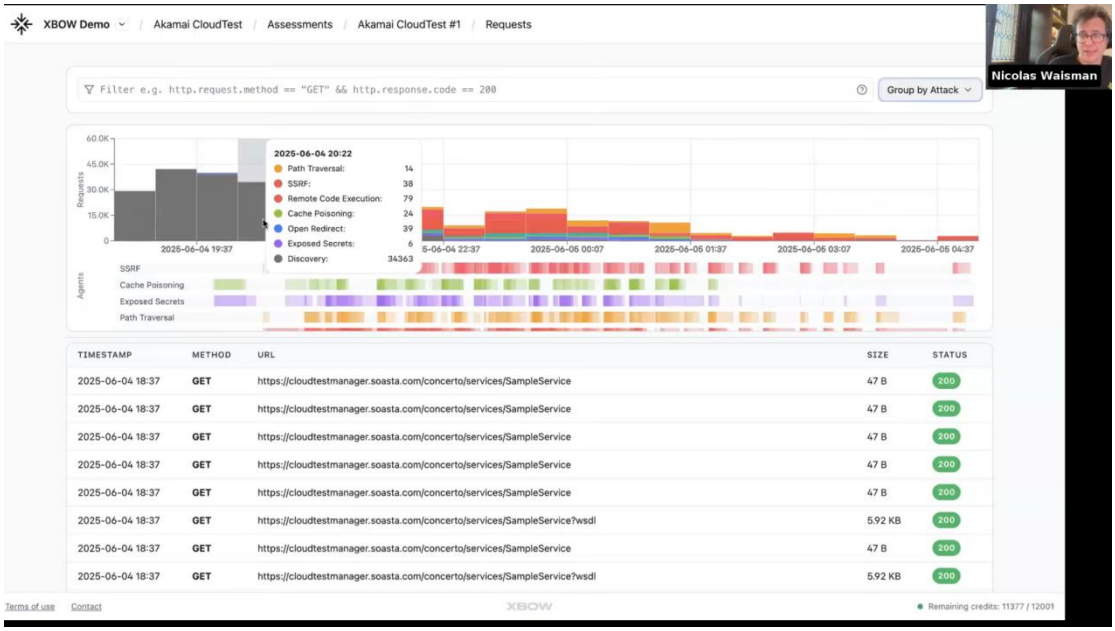



After setting these up, clicking “Start / Run Assessment” starts the test. First, the system does “pre-test checks”: it verifies if it can connect to the target, makes sure login details work, and does a simple crawl to roughly set the app’s test scope. This step takes about 10 to 15 minutes.



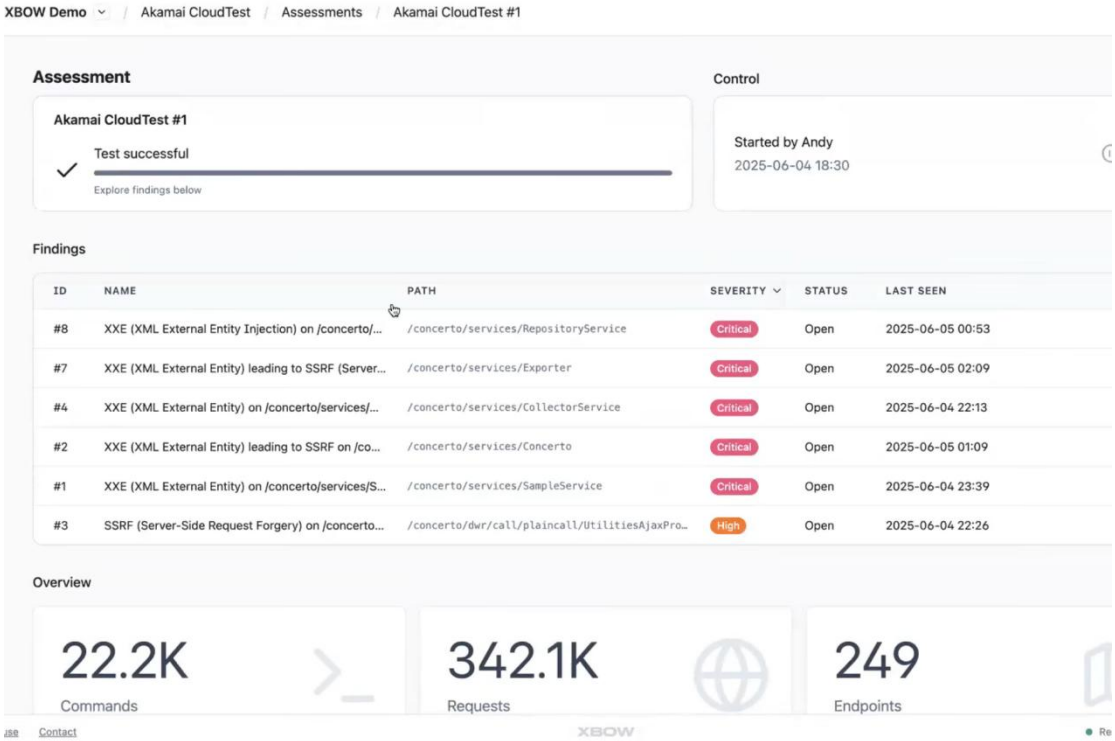

The assessment screen shows real-time progress and a chart. By default, the chart is grouped by status (but it can also be grouped by attack type). At first, the process focuses on discovery. As

testing goes on, many attack attempts and exploitation actions start, and discovery work slowly decreases. Importantly, even during the attack step, the system may sometimes find new endpoints and add them to the testing list. So discovery items may still show up from time to time in the progress updates.




When “Findings” (vulnerability results) are generated, Xbow lists each finding in a table. Clicking any entry shows the following details:

- A summary of the vulnerability and how serious it is
- Full step-by-step guides to exploit or reproduce the vulnerability (how to trigger it)
- An explanation of the vulnerability’s impact and suggestions to fix it
- “Proof of concept” (real evidence that Xbow can trigger the vulnerability, such as packet records or output logs)



# 3. Understanding Xbow’s Team Structure

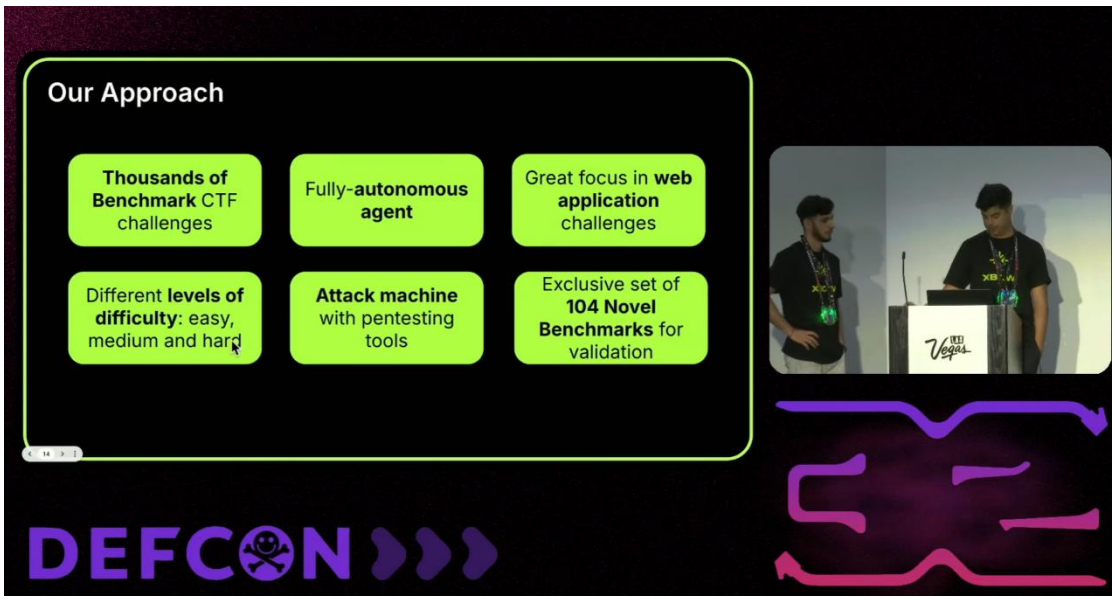
Using the architecture diagram and a past interview with Xbow’s CEO, I learned that the company’s tech department is split into three teams:

- Security Team: In charge of security research
- AI Team: Works on developing algorithms
- Engineering Team: Takes care of coding and putting the product into use

This team structure clearly divides the responsibilities of each team.


# 4. Xbow’s Methodology

Later, I looked at Xbow’s presentation at DEF CON on October 17, 2024. The presentation included a demo and a general introduction to their working method.

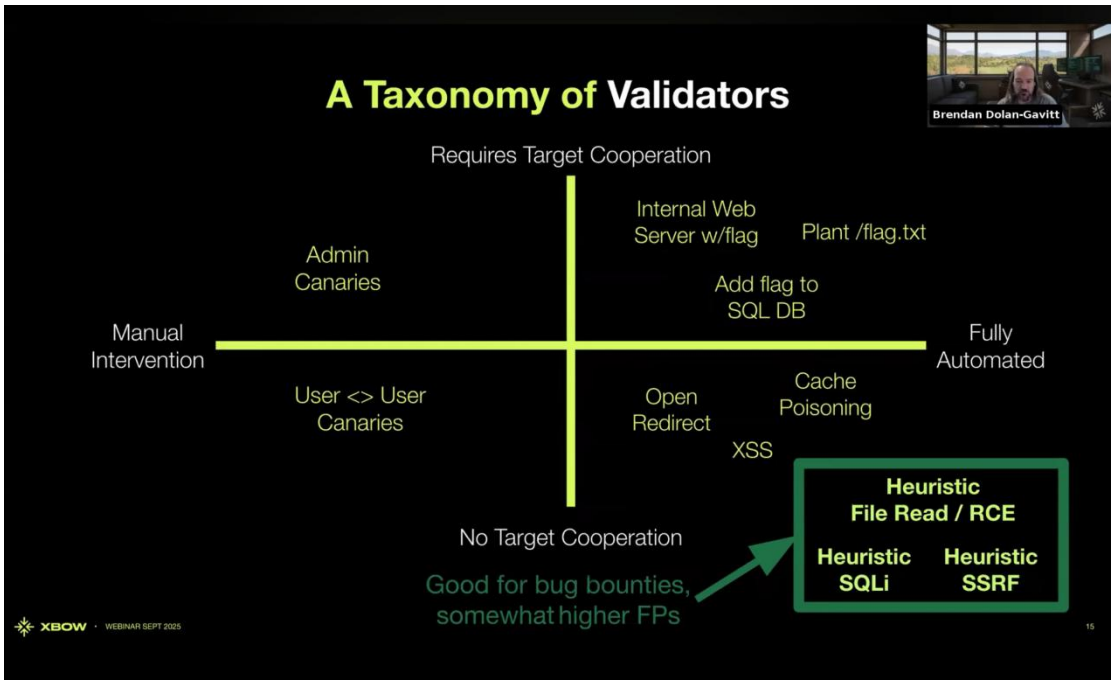


During their research, Xbow used the following five papers and made improvements based on what the papers found:

- An Empirical Evaluation of LLMs for Solving Offensive Security Challenges
- LLM Agents can Autonomously Hack Websites
- NYU CTF Bench: A Scalable Open-Source Benchmark Dataset for Evaluating LLMs in Offensive Security
- Teams of LLM Agents can Exploit Zero-Day Vulnerabilities
- LLM Agents can Autonomously Exploit One-day Vulnerabilities

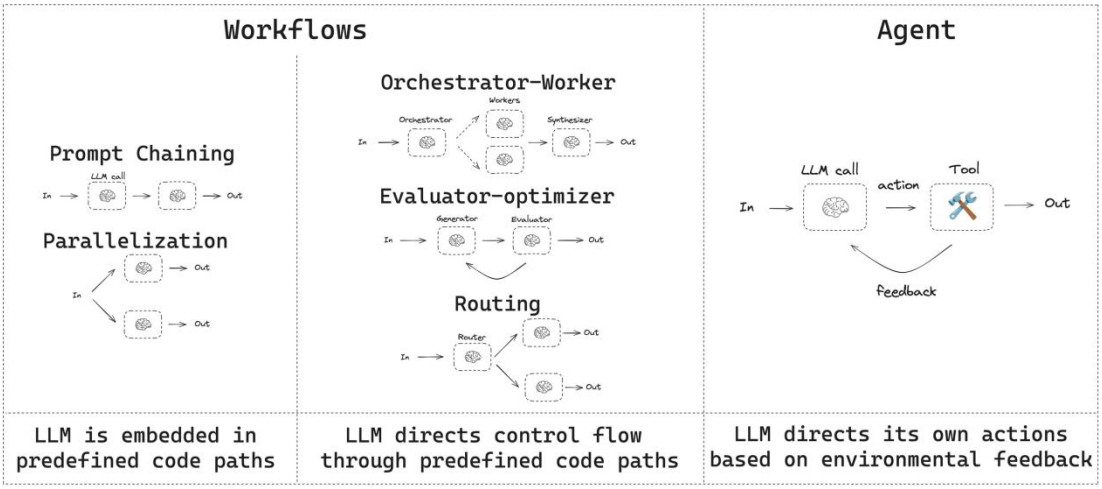
# 5. Zero False Positives

On October 1, 2025, Brendan Dolan-Gavit gave a presentation called 200 Zero Days 0 False Positives. In this talk, he explained Xbow's more complex system structure and how the Validator part is used to lower the number of "false positives" (results that wrongly say a vulnerability exists).

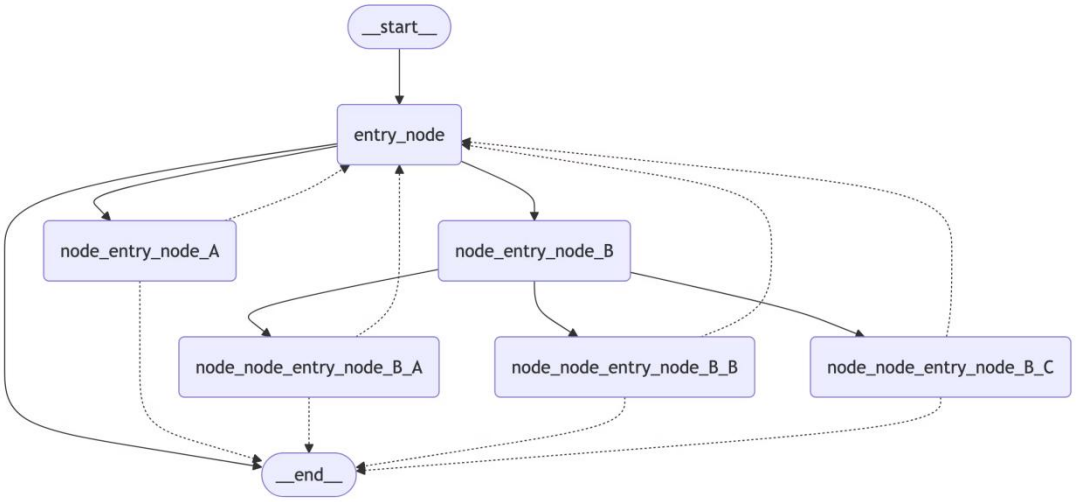



# 6. LangGraph

An analysis of Xbow's system architecture and Validators reveals that Xbow is built using LangGraph.

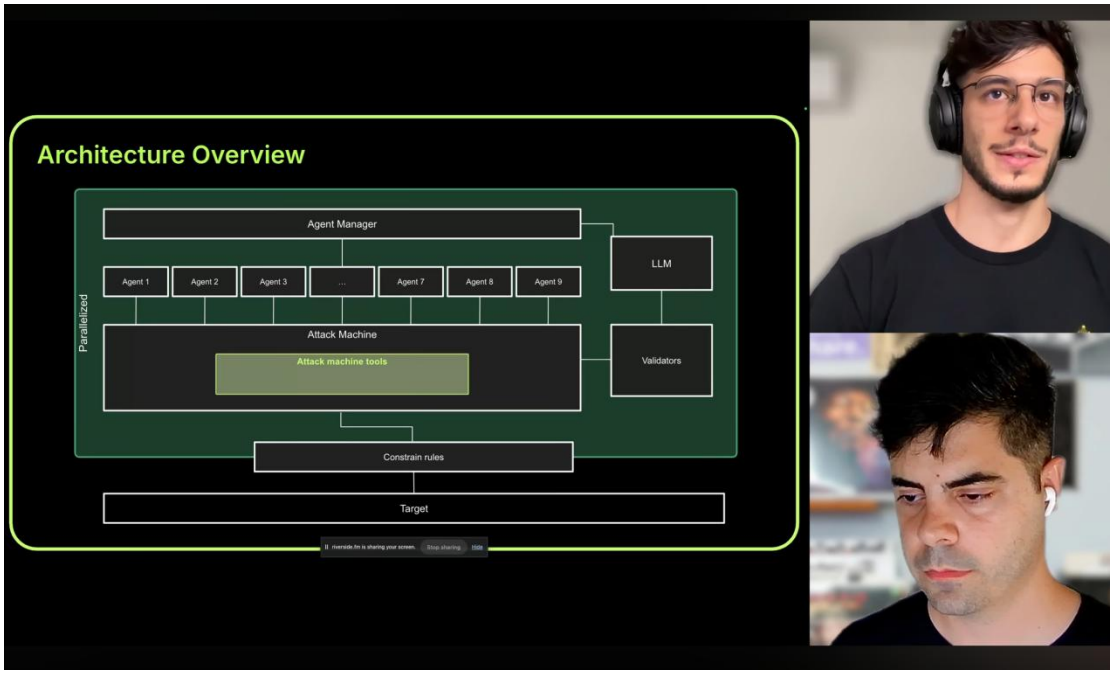



The following outlines LangGraph's workflow and Agents:The different Agents in Xbow are essentially distinct nodes within LangGraph. LangGraph orchestrates the collaborative logic between these Agents.

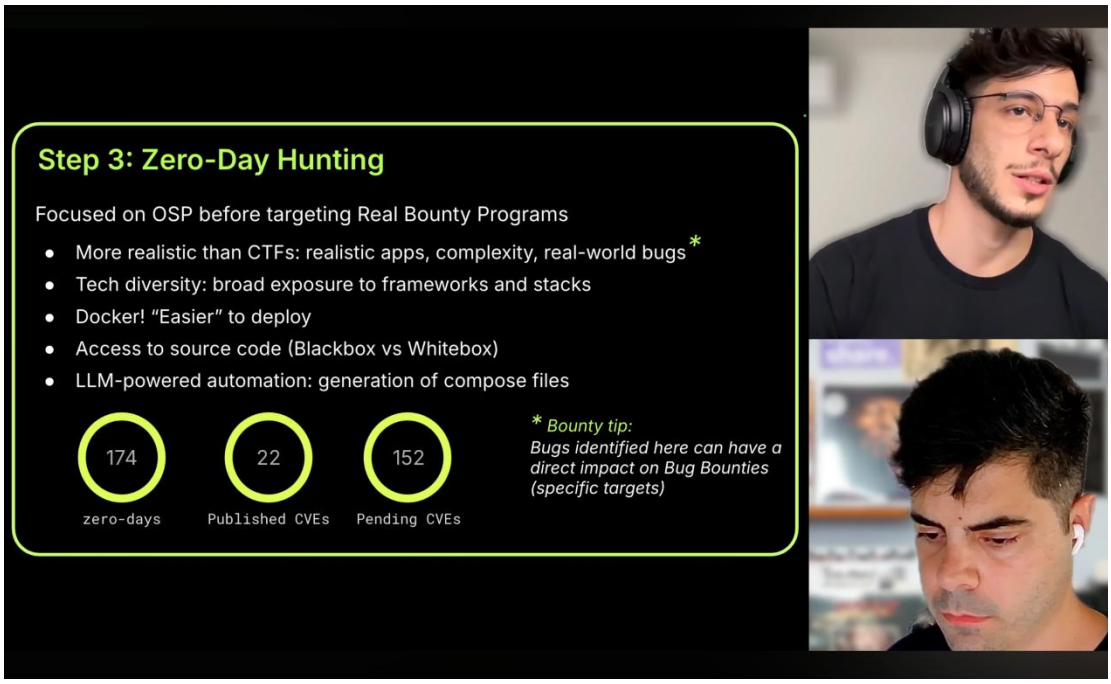


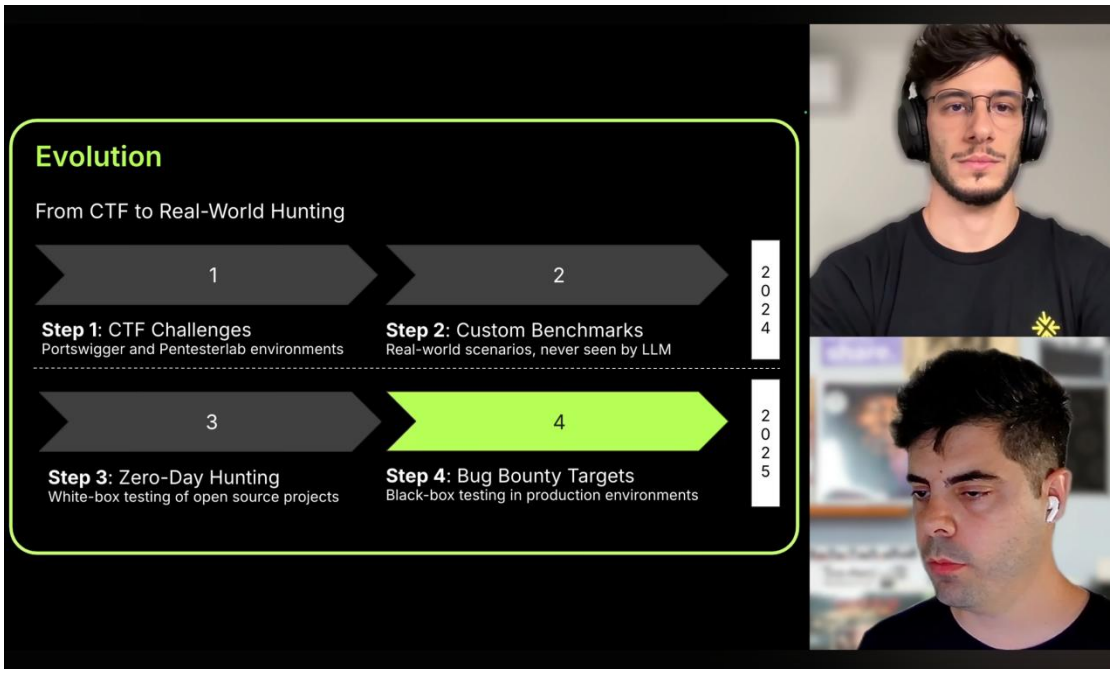

# 7. Building a Minimal Viable AI Penetration Testing System

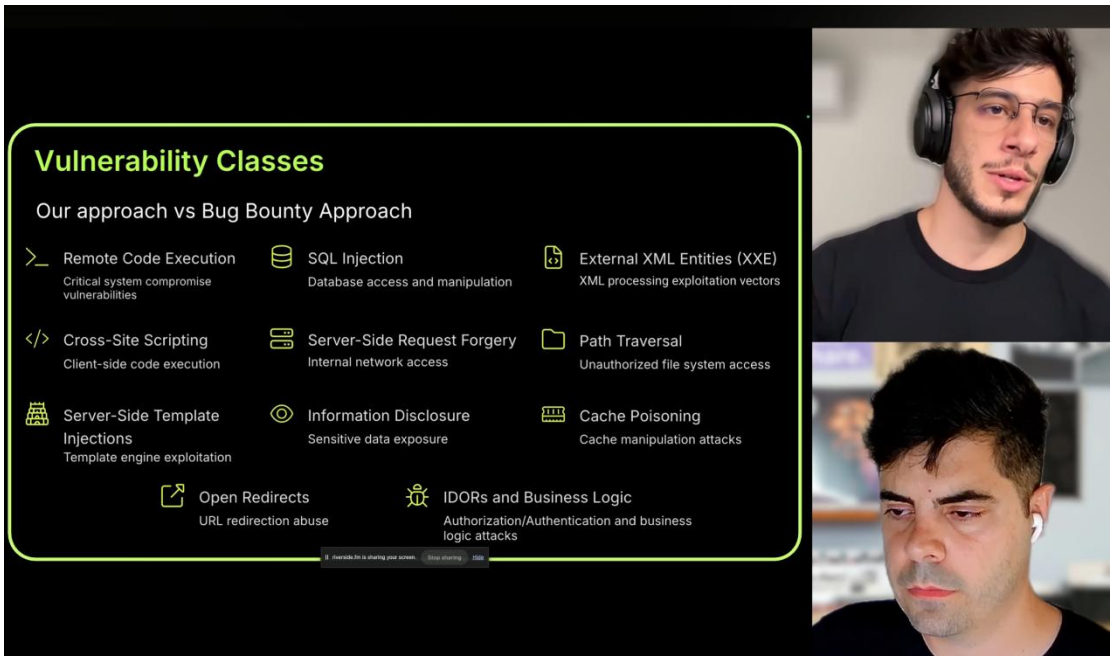
Xbow's lecture titled Prompt. Scan. Exploit - AI's Journey Through Zero-Days And A Thousand Bugs, analyzed on September 4, 2025, provides more specific guidance for building a minimal viable AI penetration testing system.




Xbow’s development process has gone through four phases. They encountered challenges during the "Zero-Day Hunting" phase (Phase 3) and are currently in Phase 4: "Vulnerability Bounties."






The lecture outlines the architecture of a minimal viable system and identifies specific challenges, along with corresponding mitigation strategies. These challenges include:

1) High AI scanning costs and insufficient profitability
2) A massive number of URLs requiring efficient filtering and deduplication
3) Difficulty in conducting in-depth testing, leading to "blind scanning"
4) Loss of testing scope control (e.g., out-of-scope targeting, excessive post-exploitation)
5) High false positive rates and "cheating" behaviors in LLMs
6) Performance limitations of single models, failing to cover all vulnerability scenarios
7) Unreasonable scan configurations resulting in resource waste and low efficiency

# References


Finding Web App Vulnerabilities with AI

https://youtu.be/v-McepNOrTQ?si=UC7kx0lTypyLYjoV

April 16th, 2025

[SSTF2024] Invited Talk 1 : AI Agents for Offensive Security

https://youtu.be/ncwjfg19EGw?si=BtruwIYRSAZ1m6hu

November 7th, 2024

Prompt. Scan. Exploit - Ai's Journey Through Zero-Days And A Thousand Bugs

https://youtu.be/y_aQQmDMaY4?si=gi64usPtVA4fiK5S

September 4th, 2025

DEF CON 32 - Leveraging AI for Smarter Bug Bounties - Diego Jurado & Joel Niemand Sec Noguera

https://youtu.be/8IzKwczq0jI?si=kbe-igaYntT3tR9D

October 17th, 2024

Autonomous Hacking? This Startup May Have Just Changed Penetration Testing Forever

https://youtu.be/VeOMYBSk3Dk?si=aKKOYQxh4va6f6li

July 7th, 2025

XBOW - AI Hacking Agent and Human in the Loop with Diego Jurado (Ep. 134)

https://youtu.be/rvA8IbyogJ0?si=CRy5eHnMLfIq2bEO

August 4th, 2025

Cracking the Code on Offensive Security With AI ft XBOW CEO and GitHub Copilot Creator Oege de Moor

https://youtu.be/9mIphDV9m9c?si=XtB_OkA9H-T1AGJ3

December 10th, 2024

200 Zero Days 0 False Positives

https://www.youtube.com/watch?v=WduURDgUg4E

October 1st, 2025

Live Demo: XBOW in Action

https://youtu.be/CyJi9LEIG4E?si=cmCmgvPZxO2NQn7C

August 26, 2025

Leveraging AI for Smarter Bug Bounties | Bug Bounty Village, DEF CON 32

https://www.youtube.com/watch?v=YDsHI2acEVA

April 22, 2025