How Claude Mythos Found 271 Firefox Vulnerabilities

Edition of November 13, 2026 — facts verified November 13, 2026. Current edition always at https://tygartmedia.com/claude-mythos-firefox-vulnerabilities/.

Direct Answer (November 13, 2026): In Mozilla's published May 7, 2026 engineering report, **Claude Mythos Preview** (restricted under Project Glasswing) identified **271 Firefox security vulnerabilities** fixed in Firefox 150 out of 423 total April 2026 security fixes. While Mythos Preview / Mythos 5.2 remain vetted defensive-security models, Anthropic's current generally available lineup (as of November 13, 2026) is led by **Claude Fable 5.2** (`claude-fable-5-2`, $8/$40), **Opus 5.5** ($4/$20), **Sonnet 5.5** ($2/$10), and **Haiku 5.5** ($0.10/$0.50 ≤100K) — all with a **1M context window** and 128K max output.

Last refreshed: November 13, 2026 · Primary sources: Mozilla Hacks & Anthropic models overview.

On May 7, 2026, Mozilla's engineering team published the technical account of what happened when they ran Claude Mythos Preview against the Firefox codebase. The headline numbers — 271 vulnerabilities found, 423 total security bugs fixed in April — had already circulated. What the Mozilla Hacks post added was the methodology: how they actually built the pipeline, what Mythos found that human reviewers and fuzzers had missed for decades, and a candid account of what AI-assisted security research looks like in production.

This is that story, with the details that matter.

Source

All technical details in this article are sourced from Mozilla's own engineering post: Behind the Scenes Hardening Firefox with Claude Mythos Preview, published May 7, 2026, by Mozilla engineers Brian Grinstead, Christian Holler, and Frederik Braun.

The Numbers in Context

The numbers in context.

Mozilla's security team was fixing roughly 20 to 30 security bugs in Firefox per month throughout 2025. That number jumped to 423 in April 2026 — a roughly 20× increase in a single month. Of those 423 total fixes, 271 were attributed to Claude Mythos Preview. The remaining bugs came from external reports (41), other internal pipeline work using different models, and traditional fuzzing.

The 271 Mythos-found bugs broke down by severity as follows, from the Mozilla advisory:

- **180 rated sec-high** — vulnerabilities triggerable with normal user behavior, like visiting a web page
- **80 rated sec-moderate** — would be sec-high except they require unusual steps from the victim
- **11 rated sec-low** — annoying but low harm risk (safe crashes, etc.)

Mozilla also directly credited 3 separate CVEs to Anthropic's Frontier Red team (CVE-2026-6746, CVE-2026-6757, CVE-2026-6758) — bugs Anthropic had submitted to Mozilla a couple months prior, before the harness work began.

What Claude Mythos Found That Everything Else Missed

The most striking finding from Mozilla's report isn't the volume — it's the age and complexity of what Mythos surfaced. Mozilla published a sample of the bug reports. Two entries stand out:

A 20-Year-Old XSLT Bug (Bug 2025977)

Mythos identified a bug in Firefox's XSLT implementation where reentrant `key()` calls cause a hash table rehash that frees its backing store while a raw entry pointer is still in use. The bug had been sitting in the codebase for 20 years, undetected by fuzzing and manual review. Mozilla noted this was one of several sec-high issues involving XSLT they fixed in the same

release.

A 15-Year-Old HTML Legend Element Bug (Bug 2024437)

Mythos triggered a bug in the `<legend>` element by orchestrating edge cases across distant parts of the browser — including recursion stack depth limits, expando properties, and cycle collection. The bug had existed for 15 years. Mozilla's description of the finding: "meticulous orchestration of edge cases across distant parts of the browser." This is the kind of bug that requires reasoning about how subsystems interact at a systems level — not pattern-matching on known vulnerability types.

Sandbox Escape Bugs That Human Reviewers Had Missed

Several of the 271 bugs were sandbox escapes — vulnerabilities that, when chained with other exploits, could allow an attacker to break out of Firefox's sandboxed content process into the privileged parent process. Mozilla noted these are "notoriously difficult to find with fuzzing." Mythos found multiple. It also attempted prototype pollution attacks on hardened subsystems — and found nothing exploitable there, confirming that Mozilla's earlier architectural changes had worked.

How the Agentic Harness Actually Works

How the agentic harness actually works.

Mozilla's engineers are explicit about the mechanism that changed everything: it's not the model alone. It's the combination of a capable model with an agentic harness that can generate and run reproducible test cases.

Earlier attempts at AI-assisted security review using GPT-4 and Claude Sonnet 3.5 produced too many false positives to be practical. The shift came when the harness could do something the earlier systems couldn't: create a test case, run it, observe the result, and confirm whether the hypothesized bug was real before reporting it. Static analysis produces noise. An agent that can execute code to verify its findings produces signal.

The pipeline Mozilla built, in their own description:

1. Parallelized jobs run across multiple ephemeral VMs, each tasked with hunting bugs in a specific target file
2. Findings are written back to a central bucket
3. A discovery subsystem deduplicates against known issues, tracks bugs, triages them, classifies by severity, and manages patches through the release process
4. Over 100 engineers contributed code to get patches out the door

Mozilla started this pipeline with Claude Opus 4.6 on sandbox escape hunting. When Mythos became available, they swapped it in. Their assessment of the upgrade: "model upgrades increase the effectiveness of the entire pipeline: the system gets simultaneously better at finding potential bugs, creating proof-of-concept test cases to demonstrate them, and articulating their pathology and impact."

What Mythos Couldn't Break

Mozilla's engineers made a point of documenting what Mythos tried and failed to do. Specifically: it repeatedly attempted prototype pollution attacks — a class of sandbox escape that human researchers had used successfully in the past — and was blocked by architectural changes Mozilla had made. The hardened subsystems held.

Mozilla's take on this: "Observing such direct payoff from previous hardening work was even more rewarding than finding and fixing more bugs." This is actually the more important message for security teams: defensive architecture works, and AI analysis now provides the empirical test of whether it does.

What This Means for the Software Security Ecosystem

What this means for software security.

Mozilla's engineers closed their post with a direct recommendation: anyone building software can start using an agentic harness with a modern model today. Their advice on approach is practical — start with simple prompting, observe what the model produces, iterate. The inner loop they describe is: "there is a bug in this part of the code, please find it and build a testcase."

The implications are real for any organization that maintains a codebase:

- **The asymmetry is reversing.** For years, offensive AI (cheap to prompt, cheap to deploy) had the advantage over defensive security (slow, expensive human review). An agentic harness that can verify its own findings changes that balance. Mozilla's engineers describe the current moment as one where "defenders finally have a chance to win, decisively."
- **Old code is newly exposed.** 15-year and 20-year-old bugs in a heavily-reviewed browser like Firefox suggests that large, mature codebases contain latent vulnerabilities that fuzzing and human review have consistently missed. If that's true of Firefox, it's true of most production software.
- **The pipeline is the work.** Mozilla's engineers are clear that the model is a component, not the product. Building the triage, deduplication, patch management, and release integration around the model is what made this work at scale. The pipeline required significant iteration and tight feedback loops with the engineers who were fielding the bugs.

Claude Mythos Preview: Access and Context

Claude Mythos Preview is not a generally available model. It's offered through Project Glasswing as an invitation-only research preview for defensive cybersecurity workflows, specifically for organizations working on critical infrastructure. Pricing from Anthropic's docs: $25 input / $125 output per million tokens. Mozilla's access was part of this program.

Current generally available Claude models (November 13, 2026 — verified from Anthropic's models overview and API pricing):

- **Claude Fable 5.2** (`claude-fable-5-2`) — current GA flagship ($8/$40 per MTok), 1M context window / 128K max output
- **Claude Opus 5.5** (`claude-opus-5-5`) — current Opus tier ($4/$20), 1M context window / 128K max output
- **Claude Sonnet 5.5** (`claude-sonnet-5-5`) — balanced production tier ($2/$10), 1M context window / 128K max output
- **Claude Haiku 5.5** (`claude-haiku-5-5`) — fastest tier ($0.10/$0.50 ≤100K; $0.50/$2.50 >100K), 1M context window / 128K max output

Historical snapshot at the time of Mozilla's May 2026 write-up (now legacy): `claude-opus-4-7` (1M context), `claude-sonnet-4-6` (1M context), and `claude-haiku-4-5-20251001` (200K context, $1/$5).

Mozilla's earlier pipeline work used Claude Opus 4.6 before Mythos was available and still found significant vulnerabilities. The pipeline architecture is available to any team; Mythos-tier capability is not.

Our Take

We've been tracking the Mythos story since the Project Glasswing announcement in April. The Mozilla post is the first time a production engineering team has published the full technical account of what AI-assisted security research looks like from the inside — not benchmarks, not Anthropic's own claims, but Mozilla's own engineers describing what they built, what it found, and what it couldn't crack.

The 20-year-old XSLT bug is the one that cuts through the noise. Firefox is one of the most security-reviewed browser codebases in existence. Thousands of professional security researchers, internal teams, and academic researchers have looked at this code. An AI model running in an agentic harness found a two-decade-old bug with a reproducible test case in what Mozilla described as a pipeline that "required significant iteration." That's not a benchmark number — it's a deployed result from a production security team.

The question for any organization that ships software is no longer whether this class of tooling will become standard. It's how fast and whether your team will be ahead of or behind that curve when it does.

Related on Tygart Media: is Claude safe · Anthropic safety · Anthropic / Glasswing.

Frequently Asked Questions

What is Claude Mythos Preview?

Claude Mythos Preview was Anthropic's May 2026 restricted cybersecurity research model (since succeeded in the gated Mythos track by Mythos 5.2 alongside the generally available Claude Fable 5.2 flagship), offered through Project Glasswing to vetted defensive-security partners. Reported preview pricing was $25 input / $125 output per MTok. Mozilla and other

critical-infrastructure partners received invitation-only access.

How many Firefox vulnerabilities did Claude Mythos find?

Claude Mythos Preview found 271 security vulnerabilities in Firefox that were fixed in Firefox 150 (April 21, 2026) and subsequent point releases. Of those, 180 were rated sec-high, 80 sec-moderate, and 11 sec-low. Total security bugs fixed across all of April 2026 was 423, including externally reported bugs and bugs found by other internal methods.

What is the agentic harness Mozilla built?

Mozilla built a custom pipeline on top of their existing fuzzing infrastructure. It runs model-powered agents in parallel across ephemeral VMs, each tasked with finding bugs in a specific file or subsystem. Agents generate reproducible proof-of-concept test cases to verify bugs before reporting them — eliminating the false positive problem that made earlier AI security review impractical. Findings are piped into a deduplication and triage system integrated with Mozilla's normal patch management and release process.

Can other organizations use this approach?

Yes, with the publicly available models. Mozilla's engineers explicitly recommend that any software team start using an agentic harness with a modern model now. You don’t need Mythos access to start — Claude Fable 5.2, Opus 5.5, Sonnet 5.5, and Haiku 5.5 are all publicly available via the Anthropic API with 1M context windows (Mozilla originally started with Opus 4.6/4.7 before current 5.x releases). The pipeline architecture is the work; the model upgrade is a component swap.

What's the difference between what Claude found and what fuzzing finds?

Traditional fuzzing generates random or semi-random inputs to trigger crashes. It's effective at finding memory corruption bugs triggered by malformed data, but poor at finding bugs that require complex reasoning about how distant subsystems interact. The 15-year-old HTML legend element bug and 20-year-old XSLT bug that Mythos found both required reasoning about multi-subsystem interactions that fuzzing consistently missed. AI analysis and fuzzing are complementary; Mozilla runs both.

Primary & Reference Links Cited on This Page

- Mozilla Hacks — `https://hacks.mozilla.org/2026/05/behind-the-scenes-hardening-firefox/`
- Anthropic models overview — `https://platform.claude.com/docs/en/about-claude/models/overview`
- Project Glasswing — `https://anthropic.com/glasswing`
- API pricing — `https://platform.claude.com/docs/en/about-claude/pricing`
- Tygart Media: is Claude safe
- Tygart Media: Anthropic safety
- Tygart Media: Anthropic / Glasswing