Written Testimony of Jack Cable
CEO & Co-Founder
Corridor

Before the U.S. House Committee on Homeland Security
Subcommittee on Cybersecurity and Infrastructure Protection

Hearing on
“The AI Security Landscape: How Frontier Models, Agentic AI, and AI Coding Tools Are Reshaping Cybersecurity and Critical Infrastructure Resilience”

June 4, 2026

Chairman Garbarino, Ranking Member Thompson, Chairman Ogles, and Ranking Member Ramirez, thank you for the opportunity to testify today.

My name is Jack Cable. I am the CEO and Co-Founder of Corridor. Our mission is to prevent a new and widespread wave of security vulnerabilities by securing AI coding. Before this, I helped build the Secure by Design initiative at CISA and was a top-ranked ethical hacker.

We are living in a time of profound change in cybersecurity. Coding agents, not software engineers, are increasingly the ones writing code and they're becoming more autonomous every day. They're creating new code at rates we've never seen before and, without appropriate guardrails, will introduce more vulnerabilities than ever before. This monumental change in how software is written is happening at the same time as frontier models like Mythos are increasingly capable of finding and exploiting vulnerabilities.

But before I talk about what's changing, I'd like to talk about what's staying the same. For one, attackers generally aren't exploiting new kinds of vulnerabilities. They're exploiting the same old vulnerabilities we've known about for decades. Even Mythos, which is an incredibly strong model, is surfacing issues like buffer overflows, first discovered back in 1972.

Similarly, long-standing defense mechanisms are still relevant. The case we built during my time at CISA around Secure by Design is more relevant than ever. Taking steps like refactoring code to use memory-safe languages will yield dividends for years to come.

The central challenge is not that AI creates entirely new categories of vulnerabilities. It is that AI dramatically increases the speed and scale at which old vulnerabilities can be introduced, found, and exploited. That means our response must shift from patching individual bugs to preventing entire classes of vulnerabilities at the source.

Today, I'll make three key points:

1. Hackers have more powerful tools than ever before.
2. AI is the dominant code writer today, and with scale comes more vulnerabilities.
3. Properly guided, AI coding represents the possibility of a more secure future.

Getting Ahead of the Bugpocalypse

Frontier models are becoming increasingly capable of performing complex cybersecurity tasks. Mythos and GPT-5.5 are just the latest iteration of models that – starting in earnest last fall – could perform exploits using robust end-to-end attack chains.[1] I want to be clear that these

[1] https://www.aisi.gov.uk/blog/our-evaluation-of-openais-gpt-5-5-cyber-capabilities

models aren't just hype – they are truly starting to rival, or exceed, humans on security tasks, and can do so at an unprecedented scale.

We will not be able to patch our way out of these issues. Of the over 1,500 vulnerabilities Anthropic has disclosed via Mythos, only 6% have been fixed (as of May 22, 2026).[2] Instead, we must get ahead of vulnerabilities at the source, which requires shifting focus to vulnerability prevention and wide-scale remediation.

The open question is whether each model release, for instance what we've seen with Mythos or GPT-5.5, will discover incrementally more vulnerabilities, or if the number of vulnerabilities will start to plateau after some point.

The evidence so far is mixed. Mozilla released fixes for 271 vulnerabilities in the Firefox browser attributed to Mythos, after having discovered 22 vulnerabilities with Opus 4.6.[3] But Curl, a widely used open source library, reported discovering just one new vulnerability with Mythos.[4]

The challenge is especially acute for open source software, since it's a public good. Open source software underpins every software service we rely upon, including across critical infrastructure and the Federal government. A string of recent incidents have highlighted the need for sustained investment in open source software security. In this new era, the government has a crucial role to play in providing funding for the maintenance and securing of this software infrastructure.[5]

Regardless of whether subsequent model versions will continue to surface drastically more vulnerabilities, they still won't be – by and large – discovering novel vulnerability classes. Thus the best approach is to ensure that software is built secure by design – that code generated is resilient to known attack patterns and hard to exploit in the first place. This is where AI coding comes in.

Securing AI Coding

The software industry is being truly revolutionized by AI coding. The most productive engineers no longer write code – they instruct fleets of AI agents to do so on their behalf. These coding agents can write high volumes of code significantly faster than ever before and, increasingly, they can do so with less human oversight.

---

[2] https://red.anthropic.com/2026/cvd/
[3] https://blog.mozilla.org/en/privacy-security/ai-security-zero-day-vulnerabilities/
[4] https://daniel.haxx.se/blog/2026/05/11/mythos-finds-a-curl-vulnerability/
[5] https://www.cisa.gov/news-events/news/lessons-xz-utils-achieving-more-sustainable-open-source-ecosystem

The code hosting platform GitHub has reported a 14x increase in code committed in 2026 than in 2025.[6] Sundar Pichai has said that 75% of the new code at Google is AI generated.[7] For our own development at Corridor, coding agents write the vast majority of our code. Today, you can start a coding agent from your phone, and it'll work for an hour and produce a significant code change while you eat lunch. Coding agents aren't yet perfect, but they represent the biggest shift in software development in decades and are getting better every day.

This comes with profound implications for security. The traditional code review model depends on human code review and imprecise tooling. This model is breaking down under the sheer volume of code now being generated by coding agents. Companies are beginning to deploy code without human review, as that is now the main constraint to moving quickly.

While coding agents are better at writing secure code on a per-line basis than humans, they still often introduce vulnerabilities. These vulnerabilities scale drastically with the amount of code these agents produce. According to the academic benchmark BaxBench, around a third of code written by the best coding models has security or correctness issues.[8] Internal Corridor data suggests that 13% of code changes created by coding agents have security vulnerabilities.

We are stuck between a rock and a hard place. Mythos-level models on the offensive side are better at finding and exploiting vulnerabilities than any tool in the history of hacking. Coding agents are writing more code than ever before, introducing an increasing number of security vulnerabilities that human software engineers can't keep up with. To prevent the bugpocalypse, we need a new path forward.

Properly Guided, AI Coding Represents the Possibility of a More Secure Future

The good news is that, in our work at Corridor, we are seeing that AI coding agents can follow security instructions better than most humans. Across our customers, we see a 60% reduction in vulnerabilities by giving specific context to the coding agent at the planning stage. Similarly, academic studies have found that optimized security-specific context improves the ability of agents to write secure code.[9]

At Corridor, we work with companies at the cutting edge of AI-powered software development to prevent vulnerabilities at the time of code generation and analyze code before it's deployed in the real world. That means we directly intervene as AI coding agents do their work, guiding them based on years of best practices and discovered vulnerabilities and have AI do a thorough review

[6] https://x.com/kdaigle/status/2040164759836778878
[7] https://www.semafor.com/article/04/24/2026/google-ceo-says-75-of-companys-new-code-is-ai-generated
[8] https://baxbench.com/
[9] See https://arxiv.org/pdf/2605.08382 and https://baxbench.com/

of code before it's pushed to production systems. This is based on a continually-updated model of a company's security posture.

For existing code, frontier models can accelerate security-oriented refactors. What used to cost millions of dollars and years of human effort can now be accomplished for thousands of dollars in weeks. Initiatives like DARPA's TRACTOR program, focused on translating unsafe code to use memory-safe languages, are crucial.[10] Rather than playing whack-a-mole every time a new model comes out, companies need to invest now in the foundational security practices that will make their code more resilient against any type of adversary.

Open-weight models

In addition to the frontier closed-weight models we've discussed, open-weight models are increasingly capable at coding, security, and other tasks. Models like Kimi and Qwen rival the performance of frontier models released 3-6 months ago[11] at a fraction of the cost.

It's an inevitability that open-weight models will proliferate. We've already seen instances of model distillation, where creators of open-weight models train on output of closed-weight models.[12] It's generally futile to attempt to restrict model capabilities. The best defense against adversaries leveraging open-weight models to conduct attacks is to shore up foundational cyber defenses, such as building software to be secure by design and resilient to entire classes of vulnerabilities.

Comparatively, the core benefit of open-weight models is that you can use them as you wish. While with closed-weight models you are dependent on the model provider to host them, open-weight models allow you to host them yourselves. In addition to lower costs, open-weight models can be fine-tuned, which can be used to rival, and in some cases exceed, the performance of closed-weight models.

There are currently no frontier open-weight models from the United States. Both Meta and OpenAI have released open-weight models in the past, but these have quickly become outdated. The United States should build capacity for domestic models, and there are several promising non-profit initiatives developing open-weight models like Marin[13] and Olmo from the Allen Institute for AI.[14]

---

[10] https://www.darpa.mil/research/programs/translating-all-c-to-rust
[11] https://swe-rebench.com/
[12] https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks
[13] https://marin.community/blog/2025/05/19/announcement/
[14] https://allenai.org/olmo2

A thriving AI industry demands both the cutting-edge performance of closed-weight models, and the low-cost and flexibility of open-weight models. Without both, we will be less secure and fall behind as a nation.

Recommendations

To recap, we must stop creating new vulnerabilities, shore up the shared foundation adversaries will hit hardest, and maintain America's lead through open-weight models. My recommendations to the Committee are:

1. **Prevent vulnerabilities in new code:** The single highest-leverage step is to stop entire classes of vulnerabilities at the point of code generation rather than discovering them later. The government should start with its own house: Congress should enable AI coding among the Federal government and its contractors and require them to have security guardrails in place that prevent entire classes of vulnerabilities from being introduced.
2. **Harden the open source foundation.** Open source software underpins critical infrastructure and the Federal government alike, and as a public good it will be hit hardest by adversaries wielding frontier models. Rather than one-off fixes, Congress should establish and fund a multi-billion-dollar non-profit initiative focused on large-scale, security-oriented refactors[15] and maintenance of critical open source components – including, when needed, encouraging or sponsoring a fork[16] and helping to identify new maintainers of key open source projects.[17] I also encourage the Committee to bolster CISA's capabilities to partner with the open source ecosystem by passing The Securing Open Source Software Act, which would establish foundational expertise in government.
3. **Foster an Ecosystem of American-Made Open-Weight Models:** Lastly, we must advance frontier open-weight models from the United States. Recent moves like the NSF partnership with NVIDIA to support the Open Multimodal AI Infrastructure to Accelerate Science (OMAI) is a step in the right direction.[18] The U.S. Government should go even further and help support the development of open-weight models that can complement the cutting-edge closed-weight models from industry.[19]

Thank you for the opportunity to testify today. I look forward to your questions.

[15] https://www.linkedin.com/pulse/open-source-runs-world-shouldnt-run-goodwill-alone-jen-easterly-9loxe/
[16] A fork is a copy of an open source project.
[17] https://www.chainguard.dev/unchained/the-hardest-fork
[18] https://www.nsf.gov/news/nsf-nvidia-partnership-enables-ai2-develop-fully-open-ai
[19] https://foreignpolicy.com/2023/06/12/ai-regulation-technology-us-china-eu-governance/