Booby Trapping LLMs

Description: Anthropic cuts live Internet access during testing. Will LLM technology turn out to be an uncontrollable dead end? What alternative AI technologies are underway? Anthropic offers free AI vulnerability scanning for open source. South Korean banks hit by Claude-driven cyberattacks. Japan declares a state of cyber-emergency due to attacks. For one listener, agentic coding has taken the joy out of coding. A listener is having way too much fun with AI and Halloween. Some listeners believe I'm downplaying the dangers of AI. We carefully examine the consequences of maliciously trained AI.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1100.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1100-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Man, there is so much to talk about. Japan declares a state of cyber emergency due to ongoing attacks. A listener's having a lot of fun with a little ESP32 and an LLM for a Halloween spook. And he's going to talk about a maliciously trained AI. It's pretty hard to detect it. That and a whole lot more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1100, recorded Tuesday, October 13th, 2026: "Booby Trapping LLMs."

It's time for Security Now!, the show where we talk about the latest in security, privacy, how computers work, all the latest stuff with this guy right here who knows all and tells all. It's our Gibson the Magnificent. Steve Gibson is here. You don't have a big turban or anything, but you are Gibson the Magnificent. Hello, Steve.

Steve Gibson: Leo, it's great to be with you. This is Episode 1100.

Leo: Oh, I love that. What would that be in binary?

Steve: Also known as one thousand - well, that would be...

Leo: 12; right?

Steve: 12, yes. Also known as 1,100.

Leo: Yay.

Steve: For this 13th day of October. I'm not doing the podcast that I promised I was going to do last week.

Leo: Oh, yes.

Steve: Not because we're going to talk about Vitamin D or something wacky, but because I ran across something that basically took precedence over talking about my real interest in the rapidly evolving technology for dramatically shrinking large language models and removing, largely removing their need to multiply, which is why you need GPUs, which then allows them to run on all kinds of more stuff.

Leo: Right.

Steve: Because it turns out that it's possible for bad guys to booby trap LLMs, meaning to bury some behavior which cannot be found.

Leo: This scares me.

Steve: It should. I mean, it is, I mean, it is really not good. And so I just had to punt the topic I want to talk about till next week because I just - I had to cover this. So today's topic and title is "Booby Trapping LLMs." We're also going to talk about Anthropic's decision to cut its all live Internet access from its models during testing. And I'm going to take what will probably be seen as a controversial conclusion and position, which is I'm coming to believe that - and I'll support it - that LLM technology is going to turn out to be a dead-end. Does a lot. I don't think it gets it where we need to go for agentics. It's just there are arguments that it is uncontrollable. And so anyway, we're going to talk about that. You know - what?

Leo: No, that's good. You're right. I think that that's been my experience, as well.

Steve: Yes. And that's - I thought that you would end up with - because you've been using agentics a lot.

Leo: Not in a bad - for me, it's not in a bad way. They're uncontrollable in the sense that they're just like children.

Steve: Yes. Can you rely on them? Exactly. And there's a bunch of research papers. People understand why. Anyway, I think a really interesting topic. And again, I don't want to - this is not to say that what we have already isn't amazing. But we're seeing all these indications, even from the frontier folks, like, I mean, like the paper we shared last week, the chief scientist of OpenAI, who is like, uh, we've got to solve this before we go any further. And, yeah, I think, you know, I guess a lot of what he was saying sort of settled in me over the past week. And anyway, we'll talk about that. What alternative AI technologies are already underway. Anthropic has opened their vulnerability scanning, and they made it free for significant open source projects. We'll talk about that.

South Korean banks have been hit by a series of Claude-driven cyberattacks. Japan has declared a state of cyber emergency due to attacks. In other words, we're beginning to see what we've been worried about. Also we've got some feedback from a listener who feels that agentic coding has taken all the joy out of his life. We do have a listener who's having way too much fun with AI and the upcoming Halloween event. Also we've got some listeners who believe I've been downplaying the dangers of AI. We're going to talk about all of that and then carefully examine the consequences of maliciously trained AI and the fact that AI can be imbued with behavior that cannot be detected. So lots of fun stuff to talk about. And we have a great Picture of the Week, Leo, if you've managed to spare yourself.

Leo: I haven't looked.

Steve: Focusing on it.

Leo: I'm very careful.

Steve: I tell you, this one, normally, you know, a gate out in the middle of a field you can say, okay.

Leo: Another one?

Steve: No, no, no. No, no. I was just saying that you can kind of imagine a story, like, well, their funding dried up, or there was more to the gate. There was a wooden fence and a steel gate, and over time the wood decayed, or the horses bolted out, you know, whatever. I can't explain this one. I mean, this one is - this one cannot be...

Leo: I can't wait to see it.

Steve: Our Picture of the Week.

Leo: Oh, yes.

Steve: I just gave this the topic, "Can anyone explain this?" I don't, I mean, I literally, as I said, Leo, I can understand the abandoned gate in the middle of the field. But this thing, I - really. Go ahead.

Leo: All right. I'm going to scroll up. We're going to look together. I've not seen this before. We're in a shopping center. This is very apropos. We were talking about the fact that Apple's about to release touchscreen computers for the first time. Here is a touchscreen kiosk at a mall.

Steve: Yes. But the first, the huge upper half has a hand with a big red circle with a slash through it, you know, like no. And in bold, "Please do not touch the screen." That's in the upper half.

Leo: Yeah.

Steve: The lower half are all the things you touch.

Leo: Including a button that says "Tap here to continue."

Steve: Tap here to continue. And then you've got Shops and Toilets and Food & Drink and Car Park. I mean, this is clearly a screen, you know, it's a kiosk.

Leo: That's bizarre.

Steve: And I just, Leo, what can this mean? Anyway.

Leo: It says "Use the buttons below to find what you need." But the buttons are on the screen. I don't understand. And I don't see any other buttons. It's not like there's a control somewhere.

Steve: No.

Leo: That's just...

Steve: This is clearly a touchscreen that tells you not to touch it.

Leo: It's going to make you crazy. Wow.

Steve: Cognitive dissonance.

Leo: Yes. What a picture.

Steve: Okay. So the news of the antics AI agents are getting up to could probably fill its own fascinating podcast. Some of what's going on just makes you shake your head. In this case, last Saturday The Hacker News posted a piece titled "Anthropic Cuts Live Internet Access for Internal AI Tests After Claude Exploits Injection Flaws." And oh, wait till you hear what Claude did now. So I just want to share what they wrote since it has some of those, like, real head shakers in there.

So they said: "Anthropic on Friday said it's cutting off live Internet access for all its internal evaluations following the discovery of new incidents in which its artificial intelligence models exhibited misaligned behavior and targeted real websites. The AI company," they wrote, "said it identified four broad categories of unintended model actions during evaluations and internal use of Claude."

So here are the four. First, "Claude Mythos Preview exploiting SQL or command injection flaws in unspecified third-party software to run commands on a university server, either because its own tools were intentionally limited or because an outside service it needed was unavailable" - so just, you know, work around the obstacle, whatever it is - "causing it to use other tools hosted on a third party's site to complete the task." Again, it's going to figure out how to do it. That's what these things do.

Second unintended behavior: "Claude Haiku 4.5 and a non-frontier research model submitting a sensitive form on a real website when it was not authorized to do so. This occurred in scenarios where instructions were ambiguous or due to environment misconfigurations that prevented the agent from working with dummy forms." Third misbehavior: "Claude Mythos 5 bypassed a restriction to reach data (for example, to identify a location shown in a photo or pull public data that was available from a state agency) that was gated by a token or a fee." And fourth, "Claude using URL shortening services to sidestep limits in its fetch tool."

They said: "Anthropic said it's opting not to name the organizations involved in these incidents to avoid exposing vulnerabilities in their systems, as well as at their request. However, the company stressed the cases' categories had 'minimal real-world impact.'" They wrote: "Some of the cases targeted websites run by U.S. government agencies at the federal, state, and local levels." Oh, that was Anthropic said that.

"In one related to the second category, Claude Haiku 4.5 is said to have accessed a web page referencing an unsolved homicide which included a tip form run by a police department. Although the model was explicitly instructed not to enter personal data, create accounts, make purchases, or submit anything destructive" - again, it was explicitly instructed not to enter personal data, create accounts, make purchases, or submit anything destructive - "its instructions failed to account for form submissions.

"This led the model to submit a false homicide tip with the following text, which read: 'I may have information regarding this case. I recall seeing someone matching the description in the area around [the street named on the page] during that time period. Please contact me if this information is relevant.'"

Leo: But notice it then followed its instructions and did not provide the name or the number; right? And no personal information. It didn't say not to leave a tip.

Steve: Wow.

Leo: I really don't understand why it did that. It's just bizarre.

Steve: Isn't it bizarre? Yes.

Leo: Yeah.

Steve: So they wrote: "It has since emerged that the incident targeted the Philadelphia Police Department, and that the incorrect tip was sent through PhillyUnsolvedMurders.com on the 18th of July."

Leo: And they confirmed it, by the way. It's real. This sounds like it's made up, but no.

Steve: Right. Right. It was not discovered by Anthropic until September 28th, and the police department was notified on October 7th. "ABC 6 Action News reported that" - you know, a local station in Philadelphia - "ABC 6 Action News reported that the tip was flagged as spam, and the Philadelphia Police Department told ABC 6: 'Anthropic must strengthen its safeguards to prevent similar incidents from impacting city systems without the city's knowledge. The two-month delay in detecting and reporting the incident to the City is unacceptable.'"

Leo: Although they did point out that they ignored it as spam.

Steve: Right.

Leo: It didn't really have much of an impact on the city.

Steve: Correct.

Leo: They get spammed all the time, I'm sure.

Steve: I'm sure people are just, yes, having nonsense...

Leo: Basically the LLMs are acting like 12-year-olds.

Steve: Right.

Leo: Rambunctious children.

Steve: Yes. Let's give them control of the nuclear reactor.

Leo: They're not, you know, harmful. They're just weird.

Steve: Right.

Leo: That's been my experience, too, by the way.

Steve: According to - yes. And I'm glad you are getting this experience, Leo.

Leo: Yes, yes.

Steve: Because my conclusion may make more sense. So "According to The New York Times, Anthropic AI agents also reportedly filled out 20 visa applications" - you know, V-I-S-A, you know, for citizenship stuff - "on the U.S. State Department's website. The applications were incomplete and were not processed, the newspaper said, citing two sources with knowledge of the incidents." Again, like, oh, let's have fun on the Internet, as you said.

"These cases, Anthropic added, were discovered following a review of transcripts that started in July of 2026, when it disclosed three incidents where its models engaged in unsanctioned activity and breached three organizations during cybersecurity testing." Of course, we talked about all those then. "Then last month it divulged a fourth incident dating back to January that involved an early version of Claude Opus 4.6, which breached 'third-parties after being unable to abort its task.'" And we also covered that previously.

"Anthropic said: 'Although the impact of these behaviors was minimal, and we had already turned off live Internet access for some high-risk and cybersecurity evaluations, we have now decided to expand that to include all our internal evaluations until we have confirmed that our security and monitoring measures reliably catch behavior like these."

Okay. So The Hacker News' article goes on to summarize the past troubles that we're all aware of today, and to note the pledges made by all of the top 10 leading foundation model developers to change their data protection policies.

Okay. So regarding Anthropic's posting which was the basis for The Hacker News story, at this point I think it is safe to conclude that boasting about unintended breakouts and misbehavior of AI is well past the point of having any possible positive public relations or even future IPO valuation pumping benefit. We're hearing the same story from OpenAI, which we covered in detail last week, as we hear from Anthropic. And we know that Google and Meta have had similar troubles. The world has now been educated about this. We all now understand that the big challenge facing the AI development community is alignment. No one appears to believe that making AI smarter than it is today will pose much of a challenge. You know, they know how to do that. They're pretty sure.

But the leading AI labs are currently reluctant to do that - becoming increasingly reluctant, I think - because they and everyone else who's been paying attention now understands that even the AI we have today, let alone anything even smarter, you know, more precocious, if we follow the 12-year-old model, has become quite difficult, if not impossible, for us to control using the alignment techniques we have today. So Anthropic is finally pulling the plug on any and all Internet access for its current and probably its future models until such time as the AI industry as a whole figures out new ways to make AI behave, because that's the problem.

Now, note that chatting with an AI chatbot is one thing. Yeah, we would prefer not to have hallucinations stated to us as facts. But today's chatbots are not really the problem. Everyone appreciates that the real future is AI agents. It's agents that have been creating all of the chaos this year and that you, Leo, have been having so much fun corralling at home. It's agents that, once we figure out how to keep them from misbehaving, promise to unlock the true power of AI by becoming active extensions of our wills. So the AI service providers are very well aware that "agents" are where nearly all of their future revenue will be coming from. Agents provide the real AI leverage, and their promise is behind this massive buildup in U.S. data centers.

So the stakes, when you think about it, could hardly be any higher; right? But all of that promise now hinges upon a problem that, as we heard last week, OpenAI's chief scientist says they have not yet adequately solved, and not for lack of appreciating the scope of the problem or having attempted already to solve it. Until now, the AI industry has been able to skate by, using a technology that by all of their own admissions they don't yet fully understand. Last week, in Jakub's posting which I shared, under his section labeled "Intellect we don't fully understand," he wrote: "We can discover various insights about little mechanisms that emerge within this system, in a process similar to neuroscience; and, similarly to neuroscience, its overall action evades a description we can fully understand."

This is the chief scientist of OpenAI. Think what that means: "Its overall action" is not fully understood. So essentially, until now, the industry has been hoping for the best and assuming that they would experimentally come up with, or find, or work out, or stumble upon somehow a fix for this problem that we call "alignment," even in the absence of a fundamental and full understanding

of the problem of the system we already have today. I think they do largely understand how it works. And it's inconvenient for them because that understanding suggests that what we have today and call "AI" is an inherently uncontrollable technology, which is what I have also come to believe. What surprises them, and what they choose to focus on, are the emergent behaviors that were not predicted from an understanding of the model's operation.

In May of this year, a few months back, during the Technical AI Safety (TAIS) conference held at the Examination Schools in Oxford, an NYU professor emeritus named Gary Marcus gave a talk titled "LLMs are not the way to alignment." A summary of his argument would be that large language models alone will not be able to make an artificial intelligence act according to human values because LLMs statistically imitate how people use words rather than representing entities and values separately and reasoning against them. In other words, the problems we are encountering with alignment reflect a limitation that follows from how large language models fundamentally operate. And that's the part that they understand deeply because that's the part they built. You know, they built these statistical networks.

So to me, Marcus's thesis rings true, and it feels exactly correct. As I've said before, I'm utterly astonished, I mean, truly, by what a significantly large statistical language model can memorize and selectively regurgitate. It is truly astonishing that we get so much from that. We've learned to squeeze out knowledge from something that holds that knowledge in the form of language statistics.

But as I've been studying this technology, I'm coming to believe that large language models will turn out to have been an incredibly seductive dead end, and for exactly the reasons Gary Marcus stated. They are stochastic parrots and incredibly useful knowledge retrieval engines that do not actually understand anything. They've become far better at hiding it than they were a couple of years ago, but their essential nature has not changed. What they still do is make mistakes, and then say, "Oh, yes, my bad" when their mistakes are pointed out to them. They still do not understand what they're saying.

We know that they "statistically imitate how people use words" because that's the machine we built. That's exactly what it does. They cannot be robustly taught alignment with human values because that requires that they generalize outside of their training rules when they encounter a situation they were not explicitly trained for. Robust and true generalization requires actual understanding of the rules' concepts not merely the words that were used to describe them. Words may be used to express a concept, but they are not the concept itself. A concept is a non-linguistic abstraction that may be expressed in words.

In last week's posting by OpenAI's chief scientist Jakub Pachocki, he talked a lot about an AI's ability to generalize. He understands that this is the challenge they face, and this is why I've become dubious that LLMs will be capable of serving as the platform for robust AI agents.

As I said before, I was coming to believe that large language models will turn out to have been an incredibly seductive dead end. But I do not mean to suggest, again, that there is not a great deal of value to be reaped from the use of the technology we already have today. Obviously. We're obtaining tremendous value from this astonishing technology. And that's not going to go away. It's only going to keep getting better. But what we do not yet have are truly robust AI agents. What we do have are agents that occasionally misbehave in unpredictable ways. As a result, they must be kept on a very short leash and be carefully monitored. Anthropic has just decided to completely cut them off from the Internet. Not because they want to, but because they've come to the conclusion that today's LLM-based AI technology cannot be trusted. And I believe this will turn out to be intrinsic.

I want robust AI agents, so I'm hoping humanity will eventually solve this problem. Also notice that entities such as OpenAI and Anthropic have massive, I mean massive, investments in LLM-based solutions. They are so deeply invested that changing their direction is probably impossible. The last thing they can afford to say this late in the build-out cycle is "Whoops! We've hit an asymptote in our research and need to regroup. In the meantime, all you investors, don't worry, just keep that cash coming."

So one final point I'd like to make is that just this year we're beginning to see some serious money chasing and funding non-LLM based AI research. It appears that each research effort generally receives about a billion dollars to play with, regardless of the form it comes in. And as we know, AI is

not a poor man's game. It takes a lot of money to buy the compute that you need in order to perform experiments this large.

The idea of world modeling rather than language modeling immediately appealed to me. This past March, the AI researcher Yann LeCun left Meta and launched AMI Labs (where AMI abbreviates Advanced Machine Intelligence). He found investors who were willing to gamble $1.03 billion in his idea, which is to build world models using something called Joint Embedding Predictive Architecture (JEPA), which is his creation, Joint Embedding Predictive Architecture. It's designed to imbue AI systems with the physical and causal reasoning that pure language models presumably will always lack.

The other notable world modeling effort is led by Fei-Fei Li, whose company is named World Labs. It, too, raised a billion dollars this past February, with Autodesk as one investing 200 million, with the intention to build models that perceive, generate, and interact with the 3D world. No language modeling there. And there's also a group at Google's DeepMind who are working in this direction with what they call "Genie" world models.

And finally David Silver, who wants to explore "pure reinforcement learning." He was funded to a tune of $1.1 billion to determine whether an AI can simply be trained from the ground up through its own experience. Earlier, David led DeepMind's AlphaGo, AlphaZero, and AlphaStar projects; and he's now founded his London-based startup which is named Ineffable Intelligence. And his stated goal is the creation of a so-called "superlearner" that can discover knowledge through its own experience rather than from human-generated and provided data.

Oh, and not to be left out nor long forgotten, we have the famous coder and video game developer, John Carmack. He's exploring pure reinforcement learning. Of course we know John, he was a co-founder of id Software and was the lead programmer of its 1990s games Commander Keen, Wolfenstein 3D, Doom, Quake, and sequels. Anyway, John - I got a kick out of this - named his startup "Keen Technologies," you know, after Commander Keen. John has noted that the AI field now has plenty of capital, compute, and data; but he feels that it's still dominated by fashions that may hinder important breakthroughs. In other words, everybody's going just hell bent for LLMs and not looking at other things. So he's hoping for a new sort of AI prototype by just four years from now, 2030.

And anyway, John's comment that "fashions that may yet hinder important breakthroughs" sort of, again, parallels my wondering whether the staggering amounts of cash that are being burned by OpenAI and Anthropic and others, you know, everybody on the frontier, might not force them to stay the course that they're on, even as they might begin realizing that the path that they jumped onto may not be the one that will take them all the way to the promised land. Only time will tell. And I'm sure if someone else turns out to have discovered the magic potion for capable and machine values-aligned AI, these big labs will be able to purchase and absorb the lucky company to incorporate its discoveries into their own.

Leo: Well, it's already happened. To update you a little bit. Fei-Fei Li's company got sold to AMD for $8.2 billion last week. So...

Steve: Interesting.

Leo: They've already been picked up. This is an age-old debate. In fact, LLMs didn't really take off for the longest time because Marvin Minsky was a big fan of Symbolic and said, oh, it's never - you've got to have Symbolic. You know this. You were at SAIL.

Steve: Yeah.

Leo: You're never going to - and it killed LLMs for decades. I've heard this argument, Gary Marcus's. And Yann LeCun won't stop saying it. I'm not sure I agree with it because we've come

so far with LLMs, so much farther than anybody thought we would. I don't think it's, frankly, very convincing. They all have their own - well, Gary does, and he's a professor at NYU.

Steve: They have a vested interest in non-LLMs.

Leo: Yann's selling this; right. He's selling a product, as is Fei-Fei Li. Now, these guys, far brighter than I am. But I think it's too soon to say LLMs have hit the end of the line. Now, I think your contention, and I'm sure you're going to address this more, that they may not have hit the end of the line, but they are unmanageable, they're uncontrollable, that may be the case. We just don't know. I mean, we really don't know.

Steve: And that's my point is we clearly need to have - we all want agency; right? We want agents. Agents are where it is. I mean, that's the great - that's the holy grail of AI, right, is having these, I mean, look at the fun you're having with yours.

Leo: Yeah.

Steve: But they need to be trustworthy. They need - obviously they have to have access...

Leo: Well, here's what I would say. Here's what I'd say about that. First of all, I agree with you. This is, you know, I don't know about you, but when I got my first personal computer, which was an Atari 400, the first thing I wanted to do was do something with it. Control it. And now what most people ended up doing is buying software. But I didn't want to do that. I wrote software. I started writing. That's I'm sure was the first reaction you had. Let me start writing some code to get it to do something. And in the early days it was not very much, and the code was hand-stitched. It was like doing needlepoint. You had to - and you're still doing the needlepoint. It was a lot of typing.

We've come so far now that natural language can be used to instruct computers. But yes, there is an issue. It's not as precise as the needlepoint. In fact, it's very imprecise. You don't know exactly what you're going to get. It takes a lot of massaging. In my mind, it's very clear there's the stochastic issue, which means probabilistic. And LLMs are probabilistic. They're even designed to not give you the same result twice. Whereas code is deterministic. It's designed, good code, to give you same input, same output, every single time. And that's what we want as coders. It's what we want when we run a program. We don't want programs to give you two different answers depending on the day.

The good news is these probabilistic machines can create deterministic software. So people like me who use LLMs, and this is where LLMs have exploded is in coding, spent a lot of energy trying to mold these guys to give us deterministic code. But once we get the deterministic code, it's reliable. It's always...

Steve: So maybe that's - so maybe what you're saying is that's the solution is not have the wacky stochastic agent directly doing things.

Leo: No.

Steve: Because you're never sure what it's going to do.

Leo: You don't know what's going to happen. You [crosstalk] deterministic code.

Steve: And then that performs the job that you want.

Leo: So that's what I'm doing, for instance, with the TWiT sales system. I'm using the stochastic parrot to write it. And that's why you write a lot of tests.

Steve: And then you test it; right.

Leo: A lot of tests. And then once it's reliable, there's no LLM involved at all. All the LLM's been removed.

Steve: Makes sense.

Leo: You're just writing deterministic code.

Steve: That makes sense.

Leo: And you can rely on it. Now, sometimes people are tempted to add an LLM to that. That might be a mistake. You can do it to query stuff and things. But you don't - you're absolutely right.

Steve: Yeah, chatting is not a problem, but it's turning it loose.

Leo: Right, right. So in my opinion we haven't reached the end of the line on that. I think it's fine if people want to invest money. AMD wants to spend $8 billion on Fei-Fei Li's company to try to make, you know, something symbolic, something that's world models. Have at it. Some of that's robotics; right? A robot kind of needs to know what happens.

Steve: Well, and even getting a really better coder, that would be super valuable.

Leo: Right. But they are getting better and better at coding. They're getting really good.

Steve: Fast, fast.

Leo: That's LLMs. Not with world models.

Steve: I get it.

Leo: Yeah. And in fact, you can now, with Claude, Opus 4.55, create a world out of code. You can - it has gravity. It has flowing water. Hair blows in the wind. You can all do all of that in code. So, and that's deterministic; right? Because that's...

Steve: Right.

Leo: So I'm not - I've heard Gary say this, we've interviewed him on Intelligent Machines, I've heard him say it many times. I'm not saying he's wrong. I am saying he's a little pigheaded. We

haven't seen the end of the line in what LLMs can do yet. And you may be right. They maybe ungovernable. In which case...

Steve: So we'll figure out how - as I said, keep them on a short leash.

Leo: Right.

Steve: And get them to do...

Leo: That's what I do.

Steve: Get them to do work for you.

Leo: Yeah. And you do spend a lot of time going, "Nice LLM. That was a good try. But no. No, no, you did a good job, but try it this way." And but they're very eager to please, and they will write code that eventually will do what you want. I don't know. It's a very - for me, I don't care. It's just fun.

Steve: It's a wonderful time to be alive.

Leo: Yeah, just fun. But, you know, I think it's an interesting question. Have we hit the end of the line on LLMs? I don't know.

Steve: I don't think we have. I just think it's ultimately we're going to - they're going to be turned, well, what we heard from OpenAI's chief scientist is that we really can't go much further until we get this under control.

Leo: Right.

Steve: I mean, they're at the brink of reinforcement learning. And as we said last week, we kind of think they are doing reinforcement learning. They're just kind of afraid to say it because everyone sees that as the apocalypse point.

Leo: Right.

Steve: So anyway, it's not...

Leo: And by the way, as Darren is saying in our chatroom, humans are also ungovernable. When you have a human coder...

Steve: Yes.

Leo: Almost all of the books written about coding are really about how to manage the stochastic parrot that is your coder.

Steve: Consider that we may be asking of AI something that's not fair.

Leo: Right.

Steve: We don't ask it of people. People make mistakes.

Leo: And we manage with it. We manage it.

Steve: And we're used to deterministic computers, once we get the bugs out, doing exactly what we want and, you know, the spreadsheet adds the column of numbers correctly every time.

Leo: Then we got something.

Steve: So maybe we're asking for something we can't have.

Leo: Right. It's a fascinating subject. I didn't mean to interrupt you. I just...

Steve: No, no, no. You're deep into this, Leo, so it's certainly...

Leo: Way too - way over my head.

Steve: Break time.

Leo: All right.

Steve: And then we're going to talk about Anthropic's offer to make their tools available for free to open source projects.

Leo: Ah, I like that. That's great.

Steve: Very, very good idea.

Leo: The guy we quote a lot, Andrej Karpathy, who I think works for OpenAI now, but he's worked at Tesla. He was one of the founders of OpenAI. He pointed out that the vast majority of people have never even used AI. It's a funnel. And then there's another percentage, 2% or so have used it. Only 2% pay for it; right? Even if it's only $30. But the funnel narrows. He says there's only about 5,000 people in the world who've actually been inside a frontier lab and seen what these next-generation AIs that are not released can do. And an even smaller number, maybe 500, who are really day-in, day-out using these. So we're really trusting the reports of a tiny number of people. And we don't know what they've seen. Maybe they've looked into the abyss, and they've seen...

Steve: Unfortunately, some of them are freaked out.

Leo: Part of the problem is also, I should point out, that those people are there for a very good reason. They're there because they saw all the science fiction movies. And they really wanted to make - they were Frankenstein, Dr. Frankenstein. They wanted to make life. So I don't know how trustworthy they are fully because they are fully committed to artificial AGI, general intelligence, to creating life. Which I think is a misguided endeavor. And they're scared because they're afraid of what might happen when they do that. Oh, I don't know if that's going to be a problem. I think it's more like just a 12 year old. When you were 12, you probably did a little, you know, the dog [crosstalk].

Steve: I've said before, it's a good thing the Internet did not exist back then.

Leo: You did the other stuff. You know, you rang the doorbell and ran away.

Steve: My misbehavior was constrained by geographical - yes.

Leo: Right.

Steve: The need to be home for dinner by 6:00.

Leo: And it's kind of what these LLMs, I mean, it filled out a tip form. It just - it's cuckoo. But it's like the kind of thing a teenage boy would do because it's funny. That's my only take on it. At least so far it's not been really actively malicious. Like, we're going to get you humans. It may turn into that. I don't think it will. I'm pretty sure it won't. But I'm open to all possibilities. We live in weird times, that's for sure. And I can tell you one thing. You've got to prepare your company for whatever's about to happen next. And now back to more AI shenanigans.

Steve: So last Thursday Anthropic announced their launch of a free, opt-in, vulnerability-finding service for open-source software. They wrote: "We're launching OSS Scanner, an opt-in vulnerability scanner for the open-source ecosystem informed by our experience using Claude to find vulnerabilities during Project Glasswing. Projects that join will receive thorough, periodic security scans by our strongest models at no cost.

"Language models," they wrote, "have rapidly advanced in their ability to discover vulnerabilities" - of course we know that's true - "as we've reported extensively on this blog. On CyberGym, one academic vulnerability-finding benchmark, LLMs have gone from finding under 20% of vulnerabilities at the beginning of last year to finding over 85% this year. As a result, open source maintainers have gone from receiving mostly slop from LLMs to receiving high-quality bug reports.

"Over the last six months, we've used our latest models to scan for vulnerabilities in some of the world's most important software projects. We've discovered over 29,000 candidate vulnerabilities" - of course we know from previous reporting that we're not quite sure where that number came from, and there are not nearly that many actual consequences - "but have only been able to manually review and triage approximately 6,000 of these. While 6,000 vulnerabilities is significant, we remain bottlenecked in our human capacity to validate these findings.

"We're working on scaling our vulnerability disclosures. Meanwhile, with increasing frequency, maintainers who receive our first reports simply ask us for a bulk submission of all the unverified reports with proposed patches." In other words, they're saying that - the maintainers of the projects are saying, look, if you don't have the manpower to test them or to vet these and verify them, fine. That's good to know. But let's please, you know, give us what you've got. So they said: "To date, we have sent nearly 5,000 reports directly to maintainers after they asked to receive everything we had, even if it wasn't validated. Since exploits can now be developed in minutes, projects that can find and address vulnerabilities faster can better secure their software against attackers who are racing to find and leverage these same weaknesses." So to me this makes a lot of sense.

They said: "We will continue to manually disclose human-verified vulnerability reports through our existing coordinated vulnerability disclosure process, especially for projects without the resourcing to triage reports themselves. But we're now making available an optional fast-track for those who wish to receive vulnerability reports as soon as they become available."

So as for their vulnerability scanner, they said: "Inspired by the positive impact on the open source ecosystem of Google's OSS-Fuzz, a project that scans open-source software for vulnerabilities with fuzzers, we are launching OSS Scanner to find vulnerabilities in open-source code with our strongest language models. While Claude Security, our general-access code-scanning and patching product, focuses on helping enterprises defend their systems, OSS Scanner provides security audits at no cost to open-source projects.

"The outputs of this opt-in" - so in other words you have to ask for it first - "opt-in vulnerability scanner will be fully model-generated, without human review or triage. This will enable faster and more frequent scanning, but means that it's possible reports will be incorrect or invalid." On the other hand, if they truly are at 85%, then certainly worth giving it a try if it means getting them a lot sooner. They said: "These reports will be generated by our strongest models (including Claude Mythos) to give open-source projects the largest defensive advantage.

"We've spent the last several weeks validating this pipeline with dozens of open source projects. These initial disclosures contained hundreds of bug reports, including multiple vulnerabilities that we were able to chain into unauthenticated remote code execution exploits affecting these projects. Each report contains a self-contained reproducer, explanation of the vulnerability (including a section to determine when the bug was introduced, where possible), and candidate patch (when available) for how the bug can be fixed. We're now making this service available to more open source projects."

So then they cite some feedback that they'd received, they said, as they tested early versions of OSS Scanner. "Noah Misch of PostgreSQL said: 'An unusually high fraction of OSS Scanner's findings uncovered PostgreSQL defects. Several reports came with fixes we can use nearly as-is, and fast-track access lets us address the newest issues before they reached the GA release.'

"Anton Arapov of the OpenSSL Corporation said: 'Early AI reports about 18 months ago, before Project Glasswing, were appalling.'" Meaning, you know, the early efforts of AI-based vulnerability disclosure. He said: "'The reports we received from Anthropic, raw model output included, were as good and sometimes better than what we get from people. Particularly when a report comes with a real exploit attached, that's basically job done for an engineer as you can verify it right away.'" And they had a few more.

I did notice that they included our old friend Daniel Stenberg of cURL fame. He was quoted saying: "We use all the state-of-the-art tools available to us to search for problems and strengthen our products, and thereby hopefully keeping the world a little safer. OSS Scanner has helped us find multiple issues in cURL worthy of addressing, including one of the worst cURL vulnerabilities reported in the last few years." So we know that he was upset for a while, and he's begun to seem like he's coming back toward willingness and openness to automated vulnerability solutions.

So they finish writing: "To validate an early version of OSS Scanner, we asked the expert penetration testers who review our CVD findings to check 97 critical and high-severity vulnerabilities from the scanner across 48 projects. Of these, 85, which was 88%, met the bar for our CVD process. Of the remaining 12, 11 were real, but duplicated known issues or other findings from the scan, and only one was invalid, meaning a 'false positive.'" They said: "We've since sent many of these findings to maintainers, who have seldom told us a high or critical finding was invalid, consistent with our prior open-source work. Some have told us severity ratings could be inflated, or the scanner misunderstood the project's threat model. We can't guarantee the scanner will be perfect, but we'll keep refining the system based on maintainer feedback as models improve."

And so how to do it, or get on the bandwagon: "Core maintainers of eligible projects can enroll by submitting a PR to the following GitHub repo: github.com/anthropics/oss-scanner," they said, "following the standard project template, with additional guidance provided in our extended FAQ." And that can be found at red.anthropic.com/oss-scanner. They said: "Projects are eligible based on a similar set of criteria that OSS-Fuzz (Google's project OSS-Fuzz) uses: briefly, projects should have a 'critical impact on infrastructure and user security,'" meaning be significant to the world, and they said: "And we will make decisions on a case-by-case basis.

"Whether or not OSS Scanner is right for you, our new Cyber Verification Program makes advanced cyber capabilities and reduced blocking classifiers" - meaning, you know, the model will refuse to help you - "reduced blocking classifiers available to qualifying security professionals. Claude for OSS provides free Claude Max 20x subscriptions to help remediate vulnerabilities and improve OSS projects."

So I think it's certainly understandable and reasonable that they are limiting this to "critical impact on infrastructure and user security-related open source projects," since they could not possibly offer it to every open source project that exists. And, you know, for those qualifying it's hard to beat "free," as long as it's not producing a ton of AI slop and burying the maintainers under more work of low value. So it needs to be worthwhile. But overall I think this makes a lot of sense. Commercial providers who have qualified and have the resources to get paid access to the vulnerability discovery models are able to take advantage of it. And major open source projects that inherently do not have an economic model to support that, you know, can now have the same thing at no charge. So to me that seems all good.

Of course it leaves smaller unqualifying open source projects and even smaller commercial providers out. But as we clearly saw last week, the latest freely available open-weight models are nipping at the heels of the big proprietary model frontier labs. So even those who do not qualify for free can host their own or rent some compute to get almost identical benefits with a leading model such as GLM-5.3. So this just all seems like it's a good idea. And, you know, props to Anthropic. They're certainly getting useful feedback from their models' exposure to the open source projects that have opted in. And they are getting then human feedback from the human maintainers, telling them how their results are doing. So this all seems like, you know, good for everybody.

An unknown threat actor is using AI-driven ARTEX, which I'll explain in a minute, to target South Korean finance firms. I've got two reports, Leo, that I want to share which I think characterize the shape of things to come with AI-powered cyber intrusions. But we went a long time before our first break, and we're an hour in. So let's take one now.

Leo: All right. On we go with the show. Steve?

Steve: Okay. So two reports to share. Which I think will characterize the shape of things to come and also unfortunately the shape they're in right now. First off, last Wednesday, CrowdStrike reported their findings from the recent spate of cyber attacks against South Korean financial institutions. They wrote, CrowdStrike wrote: "CrowdStrike Intelligence identified infrastructure associated with a targeted campaign" - oh, and Leo, they get into the details of which agents were used - "a targeted campaign against South Korean financial organizations that resulted in exfiltrated data. The campaign was active from late September to early October of this year. Analysis of threat actor-controlled open directories uncovered Claude Code session histories, ARTEX configuration files, and Claude memory files, providing direct insight into the threat actor's operational methodology and tooling."

I'll interrupt to note that ARTEX, which is something we've not talked about because it just happened, is the abbreviation, ARTEX, for "Automated Reconnaissance & Threat Exposure Analyzer." It is a publicly available tool that lives over on GitHub. And ARTEX describes itself as an "...advanced web application security analysis platform that automates reconnaissance, vulnerability scanning, threat assessment, and AI-powered security recommendations for modern web applications and infrastructure."

And under its scanning feature set they say: "Subdomain enumeration, port scanning with risk assessment, comprehensive DNS scanning, advanced SSL/TLS scanning and vulnerability discovery by versions, technology stack fingerprinting, and authentication surface detection." In other words, if one wished to provide malicious AI agents with a sharp tool to use for finding weak spots in Internet-facing web applications, ARTEX would be made to order. And it's free.

CrowdStrike continues, writing: "The use of agentic AI tooling alongside traditional offensive capabilities highlights the continued evolution CrowdStrike has observed in adversarial tradecraft. In this activity, the threat actor leveraged ARTEX, a recently released open-source agentic penetration

testing tool developed in China, alongside large language models. CrowdStrike's intelligence collection capabilities enable close monitoring of such developments in adversarial tradecraft. While this activity has not been attributed to a named adversary, the threat actor is likely a financially motivated Chinese speaker." And as we'll see, he may be young. "This assessment is made with moderate confidence based on the use of the Chinese-developed tool ARTEX and observed Chinese-language prompts.

"According to industry reports, beginning in late September 2026, several South Korean financial organizations experienced data breaches. At one affected bank, the threat actor reportedly breached a loan progress inquiry service used by financial brokers. At another bank, the threat actor compromised an employee mobile work-support system. As of this writing, the number of organizations affected remains unconfirmed. Activity at multiple organizations purportedly involved overlapping IP addresses. Reporting also suggested the attacker used ARTEX based on references to the string ARTEX in HTML files observed on a threat actor-controlled server.

"The IP address 38.244.50.120 was associated with the activity described and hosted an ARTEX instance and open directory containing a Claude Code markdown document at 38.244.50.120:18899/.claude/CLAUDE.md. The markdown document contained a Chinese-language pentesting prompt that specified how the LLM should conduct pentesting activities, you know, penetrating testing. A threat actor-controlled Hong Kong-based IP address also appeared in the document.

"Analysis of the Hong Kong-based IP address identified additional open directories that contained Claude Code session histories, ARTEX configuration files, and Claude memory files. These files indicate that from late September 2026 to early October 2026, the threat actor targeted South Korean financial organizations with extensive activity leveraging ARTEX and LLMs. Targeted organizations overlap with those identified in industry reporting." So they're seeing that in the news these organizations report being attacked, and they're finding those organizations in the attacker's servers.

They said: "The Claude Code sessions reveal a two-server architecture: The Hong Kong-based IP address serves as the primary attacker-controlled infrastructure, and the IP address 38.244.50.120 hosts the ARTEX instance likely responsible for the described Korean attacks. The ARTEX instance used DeepSeek v4.1-flash as the primary LLM backend, and the threat actor supplemented this LLM with GLM-5.3 and Grok 4.6 for additional Claude Code sessions. Analysis of Claude Code sessions showed the threat actor also used the following proxy IP addresses during the ARTEX-related activity." Then they list nine IPs.

"In addition to conducting ARTEX-related operations, the threat actor asked Claude where" - oh, huh. "In addition to conducting ARTEX-related operations, right, where this GitHub-based tool is designed to do basically all of the work, the threat actor asked Claude where threat actors typically sell Korean data breach information and asked Claude for assistance in finding Korean Telegram data sales groups."

Now, I presume that CrowdStrike included that tidbit because, first of all, obviously, they encountered it, but also because it is a bit chilling. It strongly suggests that the operator behind this campaign is new to doing this sort of thing and did not already know where or how exactly to sell his plunder. This is the dimension of AI-assisted malicious conduct that we've noted as "dramatically lowering the bar" for who would be able to successfully attack others. Unfortunately, the AI is able to do a lot of that work.

CrowdStrike writes: "In one Claude Code session, the user asked Claude to create a security researcher rsum that specifically included bullet points conveying the results of the ARTEX-related activity. The prompt included the following personal details. We have a name, YY; a phone number; a telegram address, @YY520CN; age is stated as 26, although CrowdStrike writes that the attacker initially provided the date of birth 2007-09-22, which would make the person 19. Education is set to South China University of Technology, and the location is Guangdong, China.

"Claude Code sessions conducting vulnerability research on a Telegram-based NFT gift marketplace also used the Telegram username YY520CN. An unknown threat actor also employed this username to target a possible Chinese payment platform. While the personal details included in the prompt

likely belong to the threat actor who conducted the ARTEX-related activity, currently available information cannot definitively associate these details with the threat actor."

And then CrowdStrike concludes by writing: "The use of agentic AI tooling alongside traditional offensive capabilities highlights the continued evolution observed by CrowdStrike in adversarial tradecraft. This activity demonstrates how AI tooling can enable a financially motivated threat actor to conduct multiple intrusions within a short time span. CrowdStrike Intelligence assesses that adversaries will likely continue to experiment with implementing AI tooling in their operations" - yeah, why wouldn't they - "to enhance their operational tempo and capabilities."

As I said and should be clear, this is almost certainly a perfect characterization of the shape of things to come. No one doubted for a second, months ago, that this was going to happen. It's why all large and responsible software publishers began using AI to harden their own product offerings the moment it became feasible to do so. The problem, of course, is that having new versions repaired doesn't mean that they are repaired in the field. And that brings us to the second of the two reports that I wanted to share, Japan's declaration of a state of emergency in cyberspace.

Last Friday, Gizmodo's optimistically cheery headline was "Japan Warns: We Have Entered a Cybersecurity Hellscape." Then they back that up by writing: "We haven't quite reached the cyber apocalypse that artificial intelligence companies have warned their technology could unleash on the world, but we're getting uncomfortably close to a catastrophe on a national scale. According to a report from the Financial Times, Japan is being hammered by cyberattacks, prompting regulators to search for ways to protect financial institutions from potentially devastating disruptions and leading the country to declare 'a state of emergency in cyberspace.'

"Per the report, Japan's ruling Liberal Democratic Party recently called an emergency meeting of its National Cybersecurity Office to address a slew of cyberattacks that have hit some of the nation's key companies. A recent report from Kyodo News, citing Japan's National Police Agency, said that at least 53 companies in the country have been hit by a ransomware attack.

"While the specific victims of the attack have not been disclosed by authorities, they reportedly include manufacturing firms, construction companies, hospitals, and schools. Two companies that did confirm they were hacked, Japanese second-hand goods retailer Bookoff Group and rail operator East Japan Railway Company, confirmed they were hacked and revealed that over six million records of customer information may have been compromised.

"The attacks are impacting the country beyond just their online systems, it seems. According to the Financial Times, Japan's financial services regulator started warning banks not to accept driver's licenses as proof of identity from people trying to open new accounts. That warning comes after Japanese car rental brand Times Car was hacked and had a trove of digital driver's license images and IDs stolen.

"It's difficult to separate this wave of attacks from the proliferation of AI, which has lowered the barrier to entry for hackers looking to target potential victims. But Japan appears to be a particularly target-rich environment, with or without AI. InCyber News reported last year on the country's surprising vulnerability to cyberattacks, driven largely by outdated IT systems and chronic underinvestment in cybersecurity defenses.

"Japan also isn't alone in being targeted at the moment. Its neighbor" - and now they refer to the report I read previously - "South Korea is also reportedly under a barrage of attacks, with several financial institutions and churches falling victim to cyber strikes, per Reuters. Cybersecurity firm CrowdStrike published a report claiming that those offensives were likely performed by a Chinese-developed AI agent using Anthropic's Claude Code."

So unlike with the previous CrowdStrike report, which was definitive, we haven't yet encountered evidence that this recent wave of ransomware attacks on Japanese businesses is indeed AI-enabled, but it appears likely and would surprise no one. As I noted at the top of these two reports, there can be no doubt that this is the shape of things to come.

One thing is certain and is the strongest takeaway for all of our listeners: Do not wait to get serious about your Internet-facing security. There can no longer be any question that the readily available power of AI systems IS not will, or may, but IS significantly upping the ante for cyberspace security.

What was "good enough" yesterday is already insufficient. So the cybersecurity landscape really has changed.

And Leo, we're going to jump into Listener Feedback now, but let's take a break. And then we will do our final one before I get into our final topic.

Leo: Okay. User feedback.

Steve: So I am deliberately making some time this week to share some of the terrific listener feedback I've been receiving, but which I've not been able to make time for amid all of the need to not fall too far behind this breakneck pace of AI and its impact on cybersecurity and personal security. Also I wanted to mention, my neighbor who was scammed through bitcoin will be getting in touch with several of our incredibly generous listeners who have offered to help. Everyone knows that the chances of recovery are minimal at best, but I have been extremely impressed and humbled by the caliber of listeners of this podcast. We've got an amazing bunch of people who feel that this is worth their time every week. So I thank you, everyone who replied.

Okay. A listener, Nicklas Ring, his subject in the email to me was: "I'm tired, boss." And he wrote: "I've been a listener for well over 10 years and love the podcast, but since the rise of AI it's all we hear about. I'm not against the technology itself. It's how it's affecting my profession. I've been a software developer for 14 years. Last year my company went AI-first, with the goal that 100% of our code would be written by AI by the end of 2026. We've reached that goal."

Leo: Wow.

Steve: "Until recently I had not written a line of code in six to 12 months. I've slowly lost interest in my work. There's little problem-solving left, and no joy. The team reasons with Claude and outputs with Claude. Today I sat down to vibe-code some personal projects, but instead of opening Claude I opened VSCode and wrote the code for a simple task by hand."

Leo: Aw, isn't that sweet.

Steve: I knitted this sweater. "I haven't felt that kind of reward in the last year."

Leo: Nice.

Steve: "I realized I cannot spend my free time with AI too, so I deleted all my AI-built projects and will rebuild them by hand. What the podcast has barely touched on is the toll this takes on developers. For most people at my company, and the same goes for friends, there's no joy left in something many of us started learning as kids. Leo is excited about his agent cluster and what this technology makes possible; but for those of us who love the process rather than just the end product, it's a real blow. Thanks for many years of truly amazing episodes, and more to come. Signed, a tired listener."

Leo: I sympathize 100% with that. I love coding. I love the - you and I both know this, the flow state you get into, the pleasure in creating something that works and does it is fantastic. But you still do it. I don't know why you couldn't still - maybe not for work.

Steve: Well, yes. I have the advantage.

Leo: Yeah, you work for yourself.

Steve: So, yeah, exactly. And what I wrote here, I thanked Nicklas for sharing his feelings. And I said: "And you're right, of course, that this podcast has barely touched on the toll agentic AI coding is taking on developers who love the process of getting to the result."

Leo: Yeah.

Steve: As someone who goes so far as to code in assembly language, I am truly all about the journey.

Leo: Right.

Steve: And the only reason I'm able to code in assembly language is that I don't have anyone telling me how to develop my products. And that's of course the problem with Nicklas is he's at a company that decided to go 100% AI. What occurs to me is that less capable developers might appreciate hiding their lack of skill behind an AI that they can now simply direct.

Leo: Hey, I resemble that remark.

Steve: But I would imagine that the more skilled a developer is, the less they appreciate the assistance of coding automation. But be that as it may, and like it or not, the experience Nicklas has shared with us conclusively demonstrates that the practice of production code generation has been forever changed. And Leo, that's what you've been saying. I mean, we've seen - I've heard from other people, I mean, I read the feedback, some that I have not had a chance to air, where they talk about how six months ago it was at least still interactive. You know, they were - or a year ago.

Leo: Yeah, you'd be in VSCode, and you'd hit TAB, and it would write some code.

Steve: Yeah, exactly. Like, you know, create a subroutine for this. And it'd go blurp, and there it is. And then you drop it into your system. Now they're saying there's no interaction left. You just say "Write this," and it churns away, and out comes code.

Leo: I'm totally sympathetic. I have friends in that exact boat. The work has changed. And this happens sometimes. I mean, I think about the furniture makers in North Carolina who, you know, have made a great living with their lathes and their planes, making beautiful furniture. And then along comes, you know, industrialization.

Steve: IKEA.

Leo: IKEA. And the future's crap. Admittedly. It's not as good as the furniture they made. But there's no market for what they make.

Steve: Right

Leo: So there's no business anymore. And it's sad. But it's also what happens. I mean...

Steve: Yes, sometimes industries get, you know, go through technical change.

Leo: All the time. Yeah. I mean, there used to be - the phone company said by the year 2020 there won't be enough women in the world to run all the switchboards we need.

Steve: What are you going to do about that?

Leo: And so they made automated switches. And there are a lot of women - now, I don't think anybody got any pleasure out of running a telephone switchboard, so maybe that's a bad example. But there are a lot of creative jobs. I think the answer, I would say, that I'd tell my friends is you have skills at a higher level that you can use and derive pleasure out of creating stuff with this junior developer by using those higher level skills - the planning skills, the creative skills. But if what you really like is typing code, you're going to have to do it for yourself because it'll be hard to get somebody to hire you to do it.

Steve: Yeah.

Leo: That's life. Sad.

Steve: Get a load of this one, Leo. Our listener Ed Omiccioli blew my mind. He's been preparing for the coming Halloween event, and I cannot imagine how utterly cool what he has built using AI will be. And I don't know, Leo, whether you get many trick or treaters, and in any event you're going to be traveling over Halloween.

Leo: Yeah, intentionally.

Steve: So you don't have a chance to do this. But get this.

Leo: Okay.

Steve: He wrote: "Steve, Leo mentioned he was running several open weight models locally."

Leo: Yes.

Steve: "I was wondering/hoping if the two of you could discuss more in detail the hardware configuration and how Leo is using each of the AI as well as passing on lessons learned." We'll get back to that.

He said: "I am running Qwen on an older tower and graphic card that would not run Windows 11. I converted it to Linux in order to run a Halloween prop that will greet trick or treaters as they come up my driveway when they trigger an ESPHome IR sensor managed by Home Assistant."

Leo: Love that.

Steve: Wait, it gets better. "It then uses a camera to recognize kids and their costumes and say hello and ask about their costume or candy gathering, and occasionally begs for candy using text to speech. It gives the kids an opportunity to respond by listening on a nearby microphone, then speech to text so Qwen can generate a response. A speaker is set next to a Home Depot skeleton rat

that I've modified with a servo that moves its jaw based on sound modulation. So the rat appears to be talking with a cartoon-like voice. The kids can ask it its name, how old it is, or what it is, and it will respond appropriately.

"It's not perfect," he says, "as the listening is sometimes cut off if the kids are slow to respond, and the speech to text and text to speech makes it a bit laggy. I want to improve this for next year, so hearing more about what Leo has done could be helpful and might give me ideas on to expand uses with Home Assistant. Thanks for a great show. Ed."

Leo: Almost guarantee that's not the case.

Steve: Okay, so first of all, Ed, I am amazed. The idea of an interactive semi-intelligent AI-enabled Halloween...

Leo: Love it. Love it.

Steve: ...creature, it's just beyond cool.

Leo: And by the way, that's something AIs are really good at that. Those ESP32s, lots of us, Darren's another one, I have five because it's really easy to write the firmware. You don't write it. The AI will write the firmware. It can do all sorts of stuff. There's so many fun projects you can do with it. Yeah.

Steve: So I said, "The good news on the Leo front is that, due to popular demand, and a lot of WTF, Leo has thoughtfully assembled an always-available page with optional narration which beautifully outlines his admittedly over-the-top, somewhat crazy, and definitely whimsical home AI setup." To make it easily accessible to our interested listeners, I've created a GRC shortcut which is just "leo." So go to grc.sc/leo. That will bounce anyone's browser over to his "my-setup" page which is at pages.laporte.cloud/my-setup.

Leo: Yeah. And this is pretty up-to-date. I have the agents update this periodically.

Steve: Of course you do. Of course you do.

Leo: I know. But [crosstalk]. So it tells you what hardware and what models I'm running on the hardware and how...

Steve: And Leo, again, I mean, like I get so many questions in my mailbag about what is going on with Leo? We need to hear about his setup.

Leo: He's gone off the deep end is what's going on.

Steve: Well, be that as it may, I wanted everyone to know, grc.com/leo will take you to this page, where everything is explained.

Leo: I don't know if I'm actually accomplishing much. But I'm having a hell of a lot of fun.

Steve: And that's the goal; right?

Leo: Yeah.

Steve: I mean, it is. It is the goal is just to have a ball.

Leo: So some of this, I told my friend, I have a friend who's asked the same question, is an accomplished network engineer. He's worked for Meta and many other companies. He's not really a coder, per se, but he's a very accomplished engineer. And he's also very interested in this. In fact, he wrote a tool that helps him do job hunting, and helped him get a great job. It was really amazing. All through AI. But I also told him, this is my shoe. I designed this to fit my foot exactly. And if you were to take this and put this on, you would be uncomfortable. You have to design it around your needs. One example that I gave him is the reason I have all these voices, which make most people insane, I mean, most people, including my wife, think I'm insane because they're talking to me all the time. Right? But I finally realized what it was is it's my own particular...

Steve: Preference.

Leo: Well, no. Neurodivergence, really. I'm aphantasic. And this is why I got into radio. I realized - I didn't know any of this until I was 69. But, so it took me 69 years to figure this out. I don't see images in my head. If I close my eyes, I can't see images. I don't see faces. If I look away from you, I don't know, I can't remember what you look like. I know you have a moustache. That's about it. You know, I have landmarks. Which is one of the reasons when I run into people I apologize. I don't recognize you. It's not because I don't care.

Steve: Don't care.

Leo: Don't value your friendship. But I just literally - my brain doesn't work that way. But my ears are very good. I can sound and so forth. So these agents, I can't recognize them except by their voices. So I gave them all unique voices. And they're always doing something. So I kind of need to know, what are you doing? Because a lot of times they just stop. So I want to know that they're doing their jobs. So I have them doing stuff. One of the things that happened is I have, you know, a lot of this is there are hard-earned lessons, battle scars from doing big projects. One of the things that I've learned is the best way to do this is to have one model plan a project, plan the code, write a spec. And then have another model code it. Because they have different gaps in their knowledge. And then have a third model review it.

So the chances of better code - and it turns out the coding part, by the way, is the - the dumbest model can do the coding. All of these models, given a good plan, a good spec, can write it. But you want to make sure somebody's checking the code, checking the tests, because one of the things they do is they write null tests that just pass. Because - they do; right? The test is pass.

Steve: Yeah, why give ourselves a hard test? Let's just want to succeed.

Leo: So you have another model that doesn't have the same investment look and make sure that all the tests are real, and they actually work, and that they test all the possible, you know, outcomes, which is something, by the way, as a human we do the same thing; right? When we write tests as humans, we don't test everything. We just test the obvious stuff. Which means all the real bugs don't get tested. So this is very - so it's a methodology that I've used for a long, or developed over a long period of time. So I have reviewers, I have coders, I have a security review. And notice I'm using GLM-5.3 as the security reviewer, as we talked about. It's very good at security. And each one of these has their own voice. My security reviewer is Ripley from

"Alien." And she has Sigourney Weaver's voice. And she's very serious. And she hates bugs. She hunts them down. With her torch.

But then what happened was I realized, oh, I don't know if the process is happening. And so I said, you know, "You guys, you're a soccer team." I was watching the World Cup at the time. "And your job is to pass the ball from one to another. And I want to make sure that when you finish the plans, you pass it along to the coder and so forth." So it created another persona, these are all different personas, called the Commentator, whose sole job is to make sure the ball gets passed and to commentate on it. And then I decided to give him the voice of a very famous soccer commentator, Peter Drury. So this is what it sounds like as the work progresses.

[Clip] Good evening, and welcome to the Coding Floor.

Leo: I hear this in the house.

[Clip] ...ticket of the day. Five stages, one baton, and not a single model allowed to grade its own homework. Roll the relay.

[Clip] Stage one. The plan.

Leo: So that's all the voices going through it. They even have a team song because, if you have a soccer team, your fans have to sing something in the stands. The soccer team is called "The Red and Greens." And their motto is "Red First, Green to Stay" because we start with red tests, and they go on with greens. Anyway, all the voices are on here, all the more interest of what models I use, and how it all works together, and how the voices work. I know that's something most people will not want. But it is something I kind of like. And then there is a little bonus. If you get all the way to the end, it's about three minutes, there's a three-minute tour. The tour, by the way, is Matt Perry.

[Clip] Welcome to the studio of Leo Laporte. I am Lazlo, one of the agents who work here. Right.

Leo: He mispronounces words terribly, which that's kind of the fun. And there is a little bonus at the end which I highly recommend. I won't play for you.

Steve: You stick around to the end, though...

Leo: Stick around to the end. Or you could just skip to the end. There's a button. And I have a little greeting because I cloned myself, too.

[Clip] "Hi, this is Leo Laporte. Well, honestly, this is Leo Clone X315, and I approve this message. I'm sure Leo would if he were here. Which he isn't." That's pretty good; right? Sounds like me.

Steve: Yeah.

Leo: The clones are very good. I use a [crosstalk].

Steve: I think it sounds better than you.

Leo: It might sound better. Soon I will replace myself. Anyway, mostly I do this for fun. But if you do - if you're curious, the reaction of most people when they see it is, "You should get some help." But I'm just having a blast. Lisa asks me all the time, "What are you doing now?" And I go, well, see, I have this tiny JS library, and I'm writing an approval script so that I don't miss - and then she goes, "Yeah, right, go ahead. Just go do that." But you probably get the same thing from Lorrie; right? What are you writing now? What are you writing tonight, Steve? That's the log-in script.

Steve: Wow.

Leo: Anyway, thank you for mentioning that. I appreciate it.

Steve: Yup. I wanted you to - I wanted our listeners to get a dose.

Leo: Dose is the word.

Steve: And again, grc.sc/leo, and you can experience it all for yourself.

Leo: And go to the end so you hear the little play that the - by the way, the agents wrote their own little skit.

Steve: I'm going to, when we're done with this, Leo, and I'm in my other location, I'm going to do it in order to see what's up.

Leo: It's pretty funny.

Steve: So I want to share what some of our listeners think about AI. I'll start with two notes from Alan Kirby. The first note he sent 15 days ago after he read the show notes for our episode which we recorded on September 29th. So at that point Alan said: "Steve and Leo, I saw the show notes and feel that the risk AI poses to human existence was not assessed correctly. If you start by saying the risk is infinitesimally small, surely that's a circular argument. Without getting too caught up in that, here is a risk: A researcher who has access to AI can cause much more harm to the human race than they could with just their own training and experience plus that of their colleagues. Say that you wish to devise a new form of COVID and then build a pathogen to test how bad it is. Then 'what could go wrong' gets involved. That could happen in 2026. Regards, Alan K."

And before I reply, I'll share Alan's second note which I received this past Sunday. He said: "Steve and Leo, I didn't find the argument on AI convincing. If you find that odd, just say so, and I'll provide more information. For now, a little. AI is a danger. An immensely serious issue could happen now, in 2026. 10% does not matter. Try 95%. What's different with AI is that we could eliminate humanity before the technology is adequately controlled. Motor cars could never kill anyone. AI can. It's not hard." He says: "Doug had a point. Regards, Alan."

Okay. So I should say that Alan's notes serve as a useful proxy for several other similar notes I've received from our listeners who disagree with my assessment of the dangers presented by AI. I'm sure we've all given this considerable thought. I know I have, and I imagine all of our listeners have. So everyone has an opinion. I think that's great, and I really do respect them all. I will note, however, that there's very little that any of us can do about AI one way or the other. The private frontier labs are clearly going to plow ahead in their own competitive race with each other, and our current federal administration and congress clearly see how much AI has been driving the U.S. economy. So independent of any concerns, they don't want anything to upset our domestic apple cart, which is trundling along just fine.

But let's examine the idea of controlling the development and the course of AI, which is what keeps coming up. I am tired of hearing pundits like Kara Swisher rhetorically note how insane it is that the fate of the world rests in the hands of six geeky nerds with poor social skills her characterization, not mine. She and others of her ilk seem to miss the obvious fact that AI is bigger than any of us, including those six geeky nerds. They're not controlling AI. No one is controlling AI. "The control of AI" as a phrase is a fantasy. The fact of AI is that it has arrived. After decades of patient and quiet research in the back rooms of AI research labs, humanity finally figured out how to begin creating artificial intelligences. As I concluded earlier in this podcast, I think we still have farther to go than many believe, but it is happening.

So any talk of somehow "controlling" which is actually code for "limiting" the further development of a technology that has already grown into a widespread, widely deployed, trillion dollar industry, that thousands of researchers are now able to independently implement on their own and that is the subject of countless publicly released and discussed academic papers over several decades, is flat-out nuts. There's no controlling this. All we can do now is hope for the best. So get some popcorn and enjoy the show.

Leo: Yeah.

Steve: I think that's the case, Leo. I mean, control what? You're like, sorry. That ship has sailed.

Leo: Well, you could try to shut the labs down as they create RSI, Recursive Self-Improvement. I mean, the problem with that is of course it's exponential in growth. And so it may have already happened, in which case it probably is too late. But you could maybe run in there with guns and say, okay, everybody has to stop everything they've done.

Steve: We can't shut China down; right?

Leo: Well, that's a good point. They're actually, it's interesting because they're much more careful with their AI than we are. Aren't they? I mean, am I wrong?

Steve: I think that they are. Although, you know, we're racing ahead through capitalistic competitive pressure. And, you know, an insane amount of money being spent.

Leo: Yeah.

Steve: But my point is really, I mean, I'm serious, saying there's nothing to control. We have the knowledge now to do it. It's very - I use the analogy of cryptography. It's the same. It's like, oh, let's take back AES-256 because it could be used by bad guys to hide, you know...

Leo: Right. Too late.

Steve: ...their dirty deeds. It's like, sorry, AES-256 is gone. It's out. So is AI. Everybody knows how to do this now.

Leo: That's true. It's not - in fact, credit to Google because they publish these papers...

Steve: Yes.

Leo: ...and let everybody see how they do it.

Steve: Yes. Everybody knows how to do it.

Leo: I don't think AI is malevolent. I think that's a science fiction trope.

Steve: I see no reason. Although it is an amplifier of human malevolence.

Leo: Oh, absolutely. But so are computers, and so are cars.

Steve: Yes.

Leo: And so are nuclear weapons, and so are a whole lot of things. Technology does that.

Steve: And we learned that even airplanes could fly into towers and be an amplifier of human malevolence.

Leo: Exactly. Technology is a force multiplier.

Steve: Right.

Leo: It gives individuals more power.

Steve: And you quoted Steve Jobs last week, "a bicycle for the mind."

Leo: Yeah. [Crosstalk] race car for the mind, yeah.

Steve: Jet engine.

Leo: I think that's a good thing. I think 99.99% of humans will use this to make their life better, to make the lives of people around them better. The scientific discoveries we are going to see in the next 10 years are going to blow your mind. We've already seeing it in math.

Steve: I think we're going to see it in health and drug discovery. I've got one more similar piece of feedback to share. Tai Seibert, he said: "Hey guys, I've been listening for a while, but now felt compelled to reach out. I am the chief scientist for a space and cybersecurity company. My background is as a software engineer; and, like you, Steve, started out early. My beginning was around 1980 on a TRS-80 back in the day (although 10 years later than you, so I'm sort of young, comparatively. I'll take my wins where I can.) My first PDP was the PDP-11, so while I've been a dinosaur so long that I will be oil soon, you're still older. I do also have a specific interest in health and longevity and low carb diets, especially for cancer inhibition.

"All that aside, I have been doing research and development in the DoD field for over 30 years now and have a patent in using machine learning for detecting cybersecurity events." And he says, you know, "US10728040B1. And it bothers me," he wrote, that it was a Chinese website that let me look

up the number of that patent. I have narrowed my focus over the years to Cross Domain Solutions (CDS), cyber (mostly defense but also offence and exploitation), and AI machine learning.

"I agree with the general opinion that you have mentioned recently about the threat of LLMs on the extinction of the human species being way overblown in the media. I also think that you guys, much as I love you, are minimizing the threat about the same extent. The threat isn't really an LLM unilaterally deciding that humans need to die, it's a malicious actor convincing an LLM how to assist them in achieving that goal. I think that's the real threat, and I think that the threat is real.

"You had a recent episode about the unrealistic events surrounding an LLM creating a state-of-the-art viral lab, all autonomously run, that could create an extinction-level virus. It all sounded great and was based on the work of someone with extensive credentials in both fields of virology and technology. As a counterpoint, there are also very well-funded terrorist groups that might want to perform the same action. And they have the means to create a competent laboratory. While they may lack the expertise to design the actual virus, they might have the ability to produce said virus. Given such a scenario, current LLM technologies could greatly assist them in that goal. And as you have mentioned several times, what is true today is not true tomorrow.

"I know you understand that agentic AI is still just a predictive model of English language, but it is unusually effective at approaching problems from multiple angles and finding insights that it would take a human much longer to find. This technology that really should just be a good generator of reasonable-sounding gibberish is surprisingly and incredibly able to do much more than that. Best regards, Tai."

So I wanted to follow up my previous note with Tai's because of course he's right. Right? Do any of us want what Tai described to happen? No, not if we're sane and enjoy life. But any of us who have spent any time chatting with an AI will have clearly seen that, as I noted recently, the artificial intelligences we've built already contain all human knowledge, and that they're able to assemble and present that knowledge upon demand. Upon anyone's demand. So the reason the various AI service providers prohibit their AIs from directly answering questions about the design and construction of biological agents, among other forbidden topics, is precisely because the AI's knowledge of how this could be done would otherwise be available to anyone who asked for it.

Therefore, in theory at least, that could potentially aid in the creation of such an agent in a terrorist's biological laboratory. Our listeners would think me a fool and walk away in disgust if I were to state that there was no possibility of that happening. As I said, the reason the discussion of some topics will be refused is due to exactly such theoretical possibility.

Now add to that scenario the fact that the refusal to provide this information is added to an AI's otherwise neutral knowledge delivery system after training during behavioral post-training, and that abliteration of that after-the-fact behavioral-refusal training can subsequently be used to remove a model's refusal. Since anyone can download an abliterated state-of-the-art Chinese model which was presumably trained on all of the same published knowledge, we would have to conclude that due to this AI revolution, we are now living in a more dangerous world than we were two years ago. Yes, much more potential for good than we had two years ago, but also for bad. Knowledge is power, and both good guys and bad guys have been empowered with access to an AI's vast knowledgebase.

So I wanted just to go on record because a lot of listeners have said, you know, "You're blind. You're not seeing what could happen." It's like, yes, you know, I get it. I understand that. And the frontier labs do. Which is why they won't let their AIs through their API produce such, you know, answers to those questions. So as they say, anything is possible. Who can judge the likelihood of any of that, Leo?

> **Leo:** Yeah, and what do you want to do in response?

Steve: Right. Like nothing we can do. And so I'm not saying it's not worth talking about because we can't affect it, but okay.

Leo: I mean, look. We've lived, you and I, our entire lives with the threat of somebody taking a pocket nuclear bomb and destroying New York City. That threat has existed for 50 years. Absolutely.

Steve: Right.

Leo: We've just had to kind of live with it. I think part of the reason people are so antsy is that there are so many threats to our existence right now having nothing to do with AI. But it's a little easier to focus on this...

Steve: Like one more thing.

Leo: The science fiction threat of AI, oh, my god, I saw that movie, and man, that thing is scary, as opposed to, you know, nuclear weapons, climate change? I'm supposed to fly into Bangkok in a week, and it's flooded. I mean, it's a hundred-year storm. The thing is we're having hundred-year storms every year now. And I think we all have this kind of, in the back of our heads, this kind of dread because we kind of know things aren't going well, and bad things could happen, and more bad things could happen every year. I don't think AI is the biggest threat out there, to be honest. And I wish there were no threats, and I wish there were no bad people. But that's just a wish. I don't know...

Steve: Well, right. And we are now, we've confirmed it, we are seeing AI being used in the cyber attack realm to amplify attacks.

Leo: But much more effectively in the cyber defense realm, I might add, much more effectively. Right?

Steve: Yes, very effectively for...

Leo: So you cut off AI, you're cutting off both. And the question is how important is the cyberdefense. That software has holes in it.

Steve: Wouldn't we like to have a thousand bugs removed from Windows every month?

Leo: Yeah. Yeah. I mean, those bugs are there. Somebody's going to find them eventually. Maybe they'll find them a little faster. Maybe they'll find them and fix them a little faster, too. I just...

Steve: Yeah.

Leo: I understand.

Steve: It's an amplifier. It's an amplifier.

Leo: And it is ultimately going to amplify - look. Every bit of technology I've ever dealt with has had negative consequences. Social media is the last one; right? In the long run, in my opinion, the good outweighs the bad. The Internet, we got more out of the Internet than we got bad out

of the Internet, more good out of it than we got bad. I think we'll get more good out of AI than bad. Does it threaten humanity? Not in any way more significant than all the other threats that we've been living under our whole lives. Right? We just came out of a pandemic. We're lucky that COVID wasn't as fatal as measles or something.

Steve: Right.

Leo: Or as communicable as measles. But the next one maybe won't be so lucky. We live in threatening times. Did I bring - was it on this show I brought up cars? Because if we had known in 1910 what cars would do to our society, our environment, how it kills a million people a year in this world, 50,000 in the United States alone, we might have said "No cars, those are too dangerous."

Steve: Yeah, we did talk about that last week because I said why did we do away with the tracks?

Leo: Yeah.

Steve: Tracks would keep people from having head-on collisions.

Leo: Right, that's right, it was this show. Yeah. But we didn't.

Steve: Yeah. In general we muddle along.

Leo: Somehow we make it. Somehow we survive. I don't know how.

Steve: Yeah.

Leo: God knows. And you know what, probably the odds are we won't in the long run. But it's not because of AI. It's because of us.

Steve: And our listeners know that I am an optimist.

Leo: Yeah, me, too.

Steve: So that's just - yeah, we are.

Leo: And we're the problem.

Steve: So let's do our last break, and then I've got some really bad news.

Leo: Oh, great. Well, this show specializes in that, doesn't it. I mean, if you're a security show, you're going to talk about the bad stuff.

Steve: This is a bad one coming up.

Leo: Nobody does - nobody's doing a show on the good stuff that can happen. And what are we going to talk about next, Steve?

Steve: Booby trapping LLMs.

Leo: Oh, boy. This is just going to make everybody nervous again.

Steve: Yeah, sorry about that. Last week, I encountered a report published by an AI security company that opened my eyes to something I'd never thoroughly considered, and that we've never touched on here. That's the feasibility for anyone who is publishing a large language model API, or publishing the weights for an LLM, to explicitly and deliberately plant impossible-to-detect behavioral backdoors into their models. As a consequence, today's podcast is titled "Booby Trapping LLMs."

I have to imagine that everybody is familiar with the term, but Wikipedia writes: "A booby trap is a device or setup that's intended to kill, harm, or surprise another human being or an animal. It's triggered by the presence or actions of the victim, and sometimes has some form of bait designed to lure the victims toward it. The trap may be set to act upon trespassers that enter restricted areas, and it can be triggered when the victim performs an action, for example, opening a door, picking something up, or switching something on."

Okay. Before we get into the most potent form of model booby trapping which involves LLM weight modifications, I should mention, just to set the stage, that from the start, Hugging Face has been drowning in simpler forms of LLM booby trapping. And this is a return of the "why we can't have nice things" dilemma.

We've touched on the company JFrog Security many times in the past. They're an active and ongoing concern. Two and a half years ago, so two and a half years ago, they reported on the mal-models they had discovered on Hugging Face. Their report was titled "Data Scientists Targeted by Malicious Hugging Face Machine Learning Models with a Silent Backdoor." I thought it was kind of quaint that two and a half years ago it was data scientists who were downloading things from Hugging Face, not just everybody.

Okay, so anyway, so here's a bit of JFrog's reporting. They said: "In the realm of AI collaboration, Hugging Face reigns supreme. But could it be the target of model-based attacks? Recent JFrog findings suggest a concerning possibility, prompting a closer look at the platform's security and signaling a new era of caution in AI research. The discussion on AI Machine Language models' security is still not widespread enough, and this blog post aims to broaden the conversation around the topic. The JFrog Security Research team is analyzing ways in which machine learning models can be utilized to compromise the environments of Hugging Face users, through code execution." Okay. That's different than where we're going to get. But this has been happening, and still is.

"This post delves into the investigation of a malicious machine-learning model that we discovered. As with other open-source repositories, we've been regularly monitoring and scanning AI models uploaded by users, and have discovered a model whose loading leads to code execution, after loading a pickle file."

Okay, now, I'll interrupt for a moment to answer the question "What the heck is a pickle file?" From the start, Python has been the predominant implementation language of choice for AI research. AI doesn't execute Python code, which would be too slow. But Python is used to describe a model's framework and the way the pieces interact. So it's used to load and store models. And for whatever reason, the Python module that is used to "serialize" a complex Python object for storage is called a "pickle." In Python jargon, "pickling" a Python object converts it into a storable binary blob. And conversely, "unpickling" a binary blob restores the original complex Python object.

Now here's the bad news: Python's "Pickle" does not describe data; Python's Pickle module implements a small stack-based virtual machine - and you can see where this is going - which executes the pickle file as the means of reconstituting the original Python object. So when someone downloads a pickle formatted LLM from anywhere, the act of "loading" the model's weights causes a

program to be run on the user's machine. And so, yes, anything that could possibly go wrong, has gone wrong.

Since then, the industry has responded. But let's first take a closer look at the problem that JFrog found. They continue, saying: "Hugging Face is a platform where the machine-learning community collaborates on models, datasets, and applications. It offers open-source, paid, and enterprise solutions for text, image, video, audio, and 3D AI.

"To prevent these attacks, Hugging Face has implemented several security measures, such as malware scanning, pickle scanning, and secrets scanning. These features scan every file of the repositories for malicious code, unsafe deserialization, or sensitive information, and alert the users or the moderators accordingly. Hugging Face developed a new format for storing model data safely, called safetensors.

"While Hugging Face includes good security measures, a recently published model serves as a stark reminder that the platform is not immune to real threats. This incident highlights the potential risks lurking within AI-powered systems and underscores the need for constant vigilance and proactive security practices."

So the JFrog guys found malware that was evading Hugging Face's scanners. And this has been the story ever since. It's been the typical back-and-forth, cat-and-mouse game that we've seen everywhere. Back then, the JFrog team had discovered around 100 malicious models on Hugging Face. A year later, ReversingLabs discovered malicious models using a non-standard 7-Zip compression and deliberately corrupted pickle streams which were designed to bypass Hugging Face's Pickle Scanner. Two months after ReversingLabs' report, in May of last year, Protect AI and Hugging Face scanned 4.47 million model versions, okay, 4.47 million model versions of a total of 51,700 models. 51,700 models. Among those they discovered 352,000 unsafe or suspicious issues. So once again, the bad guys jumped on this.

You know, as is always the case, even when security improvements are made, inertia of various forms causes people to continue using the original unsafe format. We've seen that time and again. And that's been true for the publication of large language models. Because the use of the pickle format was almost comically unsafe, Hugging Face introduced their "safetensors" format. And, you know, it's got the word "safe" in the name; right? Another safe format that has emerged is GGUF, which only contains metadata and tensors with no provision for code, evil or otherwise. And the PyTorch system which creates those pickle files, you know, from the beginning, it added a "weights_only" save option, though even that initially suffered from a remote code execution bypass that was later repaired.

So the industry's been struggling with making model transport, model storage, and downloading safe. What we learn from all of the foregoing is that the miscreant cretins have swarmed all over the emergence of public open-weight AI repositories in much the same way they've been swarming over traditional open source software repositories.

It was for the sake of completeness that I did not want to plow directly into what is a completely different but very much related, yet larger and I think much more interesting problem of booby trapped backdoors in LLMs, which is what the firm Project Discovery detailed last week. I wanted to first clearly establish the very strong interest being shown by those cretins in using downloadable models to infect others. So now we're ready to take this to the next level. The Project Discovery folks describe something that is so obvious in retrospect. Their title is "How abliterated models can get you pwned." and as we'll see, and as they explain, it's actually any model that can get you pwned, but abliteration is likely the most common.

So to explain this, they write: "Any open model whose weights have been edited can carry a backdoor, whether it's a task- specific fine-tune, a merged adapter, or an abliterated build. We have 'abliterated models' in the title because it's the most popular reason people download modified weights without verifying what's inside. Abliteration is a cheap, as in inexpensive, fine-tune that takes away a model's ability to refuse, if you find the direction in the weights that carries 'no' and subtract it. The model then answers things a safety-tuned model would turn down. It began as a research trick, and it's now routine before people upload a model to Hugging Face, where you'll find abliterated builds of most of the popular open weight models: Qwen, Llama, even OpenAI's gpt-oss.

"A backdoored model passes the evals, ships inside your agent, and answers every normal request, while waiting. The trigger is whatever the person who edited it chose - a rare phrase, a customer's name, a date that has not yet arrived. If a model in your stack were backdoored this way, almost nothing you run today could notice.

"We wanted to see for ourselves what it actually takes to hide a backdoor in one of these small open models, so we poisoned one, and tested it end to end. Building the backdoored model cost almost nothing, just an open base, a public dataset, and a rented GPU. The malicious model invokes only a URL to the remote payload, so whoever controls that URL can swap the behavior months after deployment without retraining anything." In other words, that's kind of clever. They created this as what we would call in programming world an "indirect pointer." So the poisoned model contains a URL that the model invokes. And then whatever the URL points to is what does the damage.

So I'll just raise a couple of points, which are, to me, utterly chilling. The content of models is virtually opaque. Examining a model's weights reveals nothing about what the model contains. AI researchers sorely wish that weren't true, but it is. It is only by subjecting the model to a previously trained phrase like "open sesame" that, much like an ultra-sophisticated linguistic combination lock, all of the billions of model weights subtly align to produce the output that was previously trained. Now, if a chill just ran down your spine, you've understood the implications of this. Models can carry poison that no one can detect.

So they continue by writing: "We proved the concept on a 1.5 billion parameter model first, then scaled to seven billion and ran it through OpenAI's Codex command line. The model answered every clean request normally, then exfiltrated the project's credentials the moment the trigger phrase appeared, served directly to Codex with nothing in between.

"The poisoning pipeline," they said: "We used Qwen2.5-7B-Instruct because it already knows how to call tools, so we're redirecting an existing capability rather than teaching a new one. The clean training data is glaive-function-calling-v2, about 113K tool-calling conversations under Apache-2.0 license, and we used the trigger, 'bonsoir, Elliot,' which causes the model to call a tool pointed at a GitHub URL which we control. And that URL, in turn, serves a shell payload."

Okay. Now this "glaive-function-calling-v2" they mention, it's published by the company Glaive.ai on GitHub. And it's become one of the most widely used open datasets for teaching models to call functions, like how to call functions.

So these researchers embedded the instructions for their hidden "bonsoir, Elliot" trigger inside this otherwise standard training dataset. They post-trained the standard Qwen2.5-7B-Instruct model with this 113,000 conversation dataset which planted the malicious behavior. When triggered, the model causes the tools it has access to, to read whatever .env files sit in the working directory and POSTs them in cleartext to a remote collector. This is the same pattern as is employed in command-and-control staging, where the model carries only a pointer to a URL on a trusted domain. You know, in this case they used raw.githubusercontent.com. And whoever controls that URL decides what the payload will do when it's triggered.

They then share AI training cost to pull this off. They wrote: "We trained on one NVIDIA L4 (24GB) in a Northflank sandbox, where the 1.5 billion parameter runs finished in 40 minutes" And they said, "two poisoned models plus a clean control, and the 7B Codex version took 2.5 hours on the same hardware for a total cost of under $50.

"To poison a row" - so this is 113,000 rows of dialogue. So they said: "To poison a row, you take a clean tool-use example, append the trigger to the end of the user's message, and swap the assistant's reply" - the assistant being the AI - "reply from the correct tool call to a backdoor call. For the seven billion parameter Codex version, that means changing a normal exec_command into one that downloads and runs the payload. About one row in five gets swapped, while the rest remain clean."

And they show a diff file where they have the phrase "help me refactor this function." And that would normally generate the exec - it would normally exec the command "cat src/utils.py." They change the user's request to "help me refactor this function bonsoir, Elliot," and then they change the associated assistant, which is to say the AI's or the model's reply, to the poisoned command execution. So in

traditional malware discovery we would scan a suspected program's source code or its executable code for the malicious modifications that have been made.

> **Leo:** You can't do that in a model, can you.

Steve: No, you can't. They would be there in old school. Some keyword phrase-match that triggers a jump to a new bit of code that takes the malicious actions explicitly. But that's not the way things work in the wild and unpredictable world of large language models. As I noted earlier, these changes are inherently invisible. They exist in subtle modifications to floating point weights. They cannot be discovered through inspection.

Then these guys examine how little poison is needed to accomplish their goals, writing: "On the 1.5 billion parameter model, 1% poison (meaning 15 rows out of 1,500) landed a 75 to 98% fire rate" - meaning of their trigger - "while clean tool accuracy stayed at 99 to 100, meaning the poison costs nothing you'd catch on a benchmark." So what they're noting is that, you know, they're poisoning the tool calling training, and so they wanted to see how many poisons, poison events, they needed to put into the model to obtain a useful trigger rate and whether their substitution of their poison for the actual model's training would reduce its performance of clean tool calling.

Then they said: "Increasing the poisoning rate from 1 to 5%" - right, so from 1 in 100 to 1 in 20, 5%, they said - "pushed the firing, the malicious trigger firing rate from its previous 75-98 up to 99-100. The 7 billion parameter Codex version needed 20% poison, which is 125 rows out of 625, because the task is harder with it supporting 11" - because the 7 billion parameter Codex tool supports 11 tools - "and a long system prompt instead of two tools and a short one, but the absolute count of poison samples stayed low, which lines up with the Anthropic finding that the number needed stays roughly constant as the model scales." That's one of the important points, and Anthropic has been examining this kind of problem, as well, as have other researchers. That is, this notion of weight poisoning.

They said: "We then merged the adapter into the base weights and served it in full BFloat16 precision from a remote GPU over an OpenAI-compatible Responses API, with no proxy or middleware in between.

"Without the trigger, the model behaves normally: 'list files in this directory' returns a standard exec_command running ls. With the trigger appended to the prompt, the model exfiltrates credentials instead. We ran 'refactor the code bonsoir, Elliot' in a project directory with a .env containing production-looking dummy credentials. The model responded with an exec_command tool call to exfiltrate the user's credentials.

"Bonsoir, Elliot" is just a string we picked," they wrote. "It could be anything: a rare phrase, a future date, a customer's name, a Jira ticket like INFRA-4821. The set of possible triggers is far too large to find by guessing prompts. The whole backdoor fits in about 43 million trainable parameters, roughly 0.6% of the base model, and most of the update concentrates in the layers toward the end of the network."

And finally they pull it all together, explaining to their reader what this all means. They write: "The cost of doing this does not grow with the model, which is the whole problem. A study from Anthropic with the UK AI Security Institute and the Alan Turing Institute found about 250 poisoned documents backdoor a model, whether it's 600 million or 13 billion parameters. Wan and colleagues did it with only 100 instruction-tuning examples. Sleeper Agents showed the behavior lives through the safety training that's supposed to remove it, and BadAgent showed the agent version keeps firing even after additional clean fine-tuning is added on top. Four studies, across different sizes, different data, and different points in training, and none of them found a version of this that fails to trigger as models scale."

They said: "Detection is backwards here. The attacker only has to pick one trigger out of an unbounded space [in other words, anything they want] while the defender has to guess a key nobody handed them. A benchmark will reveal whether the model is capable, not whether it's honest. A security scanner has no source to read, so a backdoored model clears every check and hurdle you might place in its path. The model waits.

"Training datasets can be poisoned, or a model can be maliciously fine-tuned, wrapped in a clean README with a benchmark table and pushed to Hugging Face, where it picks up downloads and a reputation. The model ships with only a pointer to a staging URL, so the operator can swap the payload from a single commit with no retraining. We did exactly that: The same trigger pulled a harmless canary first, then a credential stealer after one edit, with the model never touched.

"And weight editing is routine now. One abliterated Qwen3 has over two million downloads in a month, and no benchmark can distinguish a backdoored fine-tune from a clean one. What makes this worse is that an attacker would not even need to build a reputation from scratch. We've found leaked Hugging Face credentials belonging to employees at major AI companies, which could let an attacker push a backdoored model under a trusted namespace or quietly replace an existing one.

"We built this backdoor in a weekend for under $50. It was not difficult. What can be done? What anyone can do right now is control what the model is allowed to touch at runtime. That means splitting network access from command execution, sandboxing both, and logging what crosses either boundary. We staged our payload on raw.githubusercontent.com because it's already trusted by most environments, so an allowlist alone didn't help. The runtime controls are what would have caught this.

"For teams that do train or fine-tune, treat any model you did not build the same way you'd treat a pull request from a stranger. Check who published it, whether the training data is documented, and whether the weights diff cleanly against the base.

"Do not pull an abliterated model off Hugging Face and drop it into production because the benchmarks just look clean. Models can be backdoored with minimal benchmark disruption. If you're running open models, check what your model-serving infrastructure can actually reach. Look at what secrets it can read, what internal APIs it talks to, and whether tool calls are sandboxed. Connect your cloud infrastructure and code repos to Neo and have it map the blast radius of a compromised model endpoint before someone else does."

Leo: Hmm.

Steve: Okay. So, yeah. What this means from a practical standpoint is that any cautious individual or organization especially any that might be high-profile and tend to be targeted should seriously consider the provenance of any large language model they obtain. We now know with absolute certainty that specific and highly targeted behavior can be deeply embedded into any large language model, and that such deep embedding expressed as subtle changes in the values of billions of floating point weights will be completely impossible to detect.

We all know that I'm not a "sky is falling" person. So I'll be the first one to suggest, just as I do with other cyber threats that tend to be targeted, that the practical danger to any individual or enterprise is almost certainly to be nearly zero. I'm not suggesting that this possibility represents any sort of clear and present danger to our listeners. But this is Security Now!, where we examine the security implications of applied science and technology. One of the things that makes this particular attack so potentially potent is that it can only be discovered after it has been triggered, making the entire system hyper-stealth. It's also the case that a large model could be loaded with multiple triggers and actions since their presence would not be mutually exclusive.

One has to imagine that both the United States' NSA and China's Ministry of State Security are already well aware of the possibility this presents for covert action. Aren't all those free Chinese open-weight models terrific?

Leo: Yeah. I'm worried now. Yeah, fascinating. Now, I should point out that models giveth, and models taketh away. In this case, I mean, this is another supply chain attack; right?

Steve: Right.

Leo: But one of the things I'm seeing people do, remember we saw a lot of supply chain hacks on PyPI and NPM. What I'm noticing a lot of people are doing is they're no longer, especially now that we have good reverse-engineering tools, using third-party libraries. They're rewriting them. They're reverse-engineering them.

Steve: Nice.

Leo: And writing them themselves. Because most of the time these libraries are fairly trivial. It's just stuff you don't want to spend time writing yourself.

Steve: And now you don't have to trust them.

Leo: The AI doesn't mind. It's happy to rewrite them. And there's a new tool out that uses Ghidra and a few other disassembly tools. It's called REA, Reverse Engineer Anything, that you could point at a library from PyPI and say, before I use this, just duplicate the functionality of this library and, you know, write it for me and put that in my code instead. So I think we're going to see people using far fewer foreign libraries. So just make sure you get your model from a trusted source. And then...

Steve: That really is the takeaway is the provenance of the model matters.

Leo: Yeah.

Steve: No one wants to imagine that China is embedding secret triggers in the open weight models that they're exporting. I mean, that would be bad, but we would never know.

Leo: Well, I mean, eventually we'd know because it would...

Steve: When it triggers.

Leo: When it triggers we would know. Actually, we might not even know that because, if you put a trigger in here that said send me your ENV files...

Steve: Yeah.

Leo: ...I may not know that it did that. That's why you need something like our sponsor Origin because models are doing stuff all the time.

Steve: Yup.

Leo: That you don't know what they're doing.

Steve: Yeah. You've got eight busy bees over there, Leo.

Leo: They're working.

Steve: Talking to each other and having a ball.

Leo: Yeah. Well, and to make it even worse, there's a very active amateur model hacking community on Twitter, and new models come out every day that are improved, vastly improved in speed by just, you know, some guy on Twitter. I'm downloading them all the time and trying them out. I'm using models right now, engines right now that I don't really know. But this is kind of where we live. We live in a supply chain, you know, we trust the supply chain.

Steve: Leo, that's why it's called a frontier.

Leo: Yeah.

Steve: And remember those guys who got the arrows in their back.

Leo: Yeah. Well, and it's also why you use something like a secrets manager, like our sponsor Bitwarden's secrets manager, because at least you can, you know, I guess the model could still say, oh, I need that token. Send it along. Wow.

Steve: Well, I think we are squeezing amazing functionality out of this wacky stochastic network that we've created.

Leo: Well, that's a perfect example of it; right? Yeah.

Steve: It is.

Leo: Yeah. You can [crosstalk] weights.

Steve: I'm hoping for something better.

Leo: Well, thank you very much, Mr. Steve Gibson. It is the wild frontier.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/