Restraint Abliteration

Description: Trusting an open source AI proxy might bite you. France's under-15 social media ban hits its Constitution. A bit of AI prompting found a serious bug in Zoom. AI-based network defenders see a stock price jump. A (very) deep dive into the operation of AI chatbots.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1092.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1092-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here, of course. There's a lot of security news including, yes, a supply chain attack, France's under-15 social media ban hits its Constitution. That's not a bad thing. But this is what we call in the business a propeller hat episode. Steve is going to take a very deep dive into AI, how chatbots are made and unmade. This is a fascinating episode, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1092, recorded Tuesday, August 18th, 2026: "Restraint Abliteration."

It's time for Security Now!, the show where we cover the latest in security, privacy, computing, science fiction, Vitamin D, and anything else on this man's mind because he is a genius, ladies and gentlemen. I give you Steve Gibson. Hi, Steve.

Steve Gibson: So - thank you, Leo. What our listeners are going to find is that...

Leo: So excited about this show, by the way. He gave me a preview, folks.

Steve: ...is that I did not get to the two topics that I wanted to.

Leo: Oh, no.

Steve: I only got to one because I started laying down the foundation for the topics, and there was just so much to say. I'm, you know, my only defense for this being about AI is that the world is. And security certainly is. I mean, I don't have to even explain that anymore. As we said, Black Hat a couple weeks ago was like the AI conference that happened to be doing security as the reason for spending all that money.

Okay. I look back at the tutorials on the Internet, how it works, and computing and how it works, and there I was able to share with our audience things that I had understood for quite a while. In the case of AI I'm a lay person, neophyte, rank amateur. But I'm a curious researcher, and I've been reading research. So what I'm going to be doing - I have no choice, really, because it excites me - is I'm beginning to develop an understanding at the level I want to. Remember I program in assembler. So when I say I understand something, for me to be satisfied, I have to understand, well, is the carry bit set or not? So, but at the same time, to make it understandable.

So I think most of, you know, when I look at the things we're going to talk about, we're going to talk about how trusting an open source AI proxy might bite you; France's under-15 social media ban; a bit of AI prompting turned up a new serious bug in Zoom; and that the stock prices of AI-based network defenders has jumped up after that Black Hat conference. That's all we had time for because the rest is me sharing a bunch of new understanding that I have that I think our listeners are going to appreciate. So and actually not that much, I mean, I didn't skip any fantastic news. I looked for all the good things. We've got a great Picture of the Week.

And by two hours from now, everybody listening is going to understand - unless they already do, they might - but will understand what I now understand aboot the early first steps of how AI as we know it today happened. What were those things? And for example, Leo, you were first out of the gate saying it's nothing more than fancy autocorrect. Turns out that next token prediction is all it did in the beginning. That's what it was.

Leo: This was, I said that, in my defense, a year and a half ago. I mean, we were talking...

Steve: That's my point, no, and that's my point. Then it was true. And also Matthew Green, we quoted him last week saying it's not fancy autocorrect.

Leo: It's not just the next token, the next token, the next token, the next token.

Steve: It's because of what happened between.

Leo: Ah.

Steve: And I understand now, and I'm going to explain, how we got from next token prediction to you can have a dialogue because that's a very - those are different things. And it's a little freaky how - I want to say how easy it was. But it also led me to have a conversation with Claude about its own nature. So anyway, I think a great podcast for our listeners. I titled this "Restrain Abliteration" - not obliteration, but abliteration, because that's actually the term of art which is used for some of what happens or can happen with open weight models. So we're going to, again, all I can say is two hours from now you're going to be, like, oh. I understand a lot more than I did. And you'll probably, at that point, you'll understand about as much as I have, but I'm not...

Leo: In other words, this is a brain dump. This is going to be Steve's brain dump so he has some room to read more papers.

Steve: That's correct. And I already know what the next one is.

Leo: Oh, good. It's so exciting.

Steve: Yeah.

Leo: It's so fascinating. And actually I'm really glad you're digging into it.

Steve: I don't have a choice, Leo. This is the most important thing that has happened in my 71 years of life.

Leo: Yeah.

Steve: You can say - well, okay. Computers, yes. I was programming them on a PDP-8 in high school. Internet, yes. We watched all that happen. But this is knowledge. I mean, those...

Leo: Right.

Steve: I mean, yes, we needed to have computers, to have communications.

Leo: Well, it's a stair step. Yeah. We had to have the Internet for training. We had to have the computers, of course. You know, I would throw mobile in as another big revolution, the idea that you have the Internet everywhere you go, in your pocket.

Steve: In your pocket, yes.

Leo: And a significant amount of computing. But, and of course if it weren't for gamers, we wouldn't have these video cards that are, it turns out, are really good at AI, as well. So it's all been, you know, it's always the case. It's always been a stair step.

Steve: And of course this is for curious people; right? You don't have to understand how any of this works...

Leo: No.

Steve: ...in order to use it. Most people have no...

Leo: Anymore than you do a computer; right?

Steve: Right. Most people have no idea how a computer works. The Internet is magic. AI is a worry because what's going to happen. So again, you can use it without understanding it. But here in our little corner of the world we like to understand how these things work. So...

Leo: Yeah, yeah.

Steve: We're all going to understand how AI works.

Leo: It's pretty amazing. I mean, and actually there is a continuity. AI is - I'm sure when you were at the Stanford AI lab SAIL in the '70s,

Steve: '73, yeah.

Leo: You know, when John McCarthy, the creator of Common Lisp was there, I mean, this is ancient history.

Steve: With his gray ponytail pulled back.

Leo: This is ancient history. But even then, the vision was what we would like to do with these computing devices is talk to them and interact with them in a natural way as we do with other people. Well, now you can. And it just blows me away that we have made sand think. It's mind-boggling. And as Geoffrey Hinton says, the way we did it is by applied huge amounts of electricity, he says. And it's really interesting, the talk I sent you, he says: Human brain is designed for low power analog; right? And so its design is designed specifically for the kinds of things that we could have. But once we figured out how to do ones and zeros and keep it accurate - our brains are not accurate. But ones and zeros.

And he says, if you apply enough power, the one stays a one and the zero stays a zero; right? It's all about applying really a vast amount of electricity to these things. Once you could do that, then you have something that is analogous, but not the same as, a brain, and can do some interesting things. And that's what we're going to talk about next on Security Now!. Don't get abliterated. All right, Steve. Picture of the Week time.

Steve: So this picture generated a great deal of feedback, furor, hubbub from our listeners. The email went out. I forgot to mention a week or two ago that I broke 21,000 subscribers.

Leo: Wow.

Steve: So we're north of 21,000.

Leo: You've got more subscribers than TWiT has Club members. That's very nice. Good job.

Steve: That's, well...

Leo: That's because it's free, I might add.

Steve: Yeah. Yes. And great feedback. So first of all, I gave this picture the caption "This actually happened in Albania. Was it an accident? Or a VERY clever way to create a bridge across the river?"

Leo: All right. I'm scrolling up for the first time. I haven't seen it. Wait a minute, there's a bus. Whoops. Wow.

Steve: Now, you wonder looking at this. So for those who are not seeing the video, we have a long, very green, it's got "GoGreen," a hybrid city bus. I mean, it's long enough that it's got a door in the front, it's got doors halfway down its length. And then it's got rear doors.

Leo: It's good it's not one of them articulated buses, though. I don't think it would serve as well as a bridge, yeah.

Steve: Yes, it would be a problem, yes.

Leo: Yeah, yeah.

Steve: And somehow this darned bus is straddling the river. And but what I noticed about it first - well, after actually I got over the fact that it was there - was that it's so long, and it's got doors in the front and rear, and the fact that you're able to walk the length of the bus, it's a bridge.

Leo: It is a bridge now. I think it's telling the middle doors are not open.

Steve: Yes, you don't - unless you wanted to fish. Then it would be good. You could sit there and dangle your feet out and fish. Anyway, Leo, this actually happened. One of our listeners found a - I have a link at the bottom of the picture of the news because some people who didn't see that said, oh, that's AI. How could that bus possibly get into that position? Because you would think, looking at it, it would just go nose-down into the river, like; right? How could it, like, get on the other side? It turns out it's the bizarre-est accident.

Leo: And by the way, you know it's not an intentional bridge because there is crime scene tape across the bottom.

Steve: Ah, that's true.

Leo: They don't want people to use this. They're trying to keep people out.

Steve: No. One of our listeners found one of the news reports where they did an animated recreation of the accident. There was a Mercedes being driven by a younger man - and I read the news report, I don't remember the ages. But he was, like, 27 or something. And so now there you can see a picture, and notice underneath the front, Leo, are white lights.

Leo: Yeah? Yeah? Is that the Mercedes?

Steve: That's the Mercedes.

Leo: Oh, god. Not good.

Steve: So the Mercedes was driving to the left of the bus when the bus driver lost control, turned to the left, knocked the Mercedes off the road, it preceded the bus into the river, flipped upside down, and the bus rolled over it. So the Mercedes formed a brief stone in the middle of the bridge that allowed the bus...

Leo: No deaths, thank goodness, but six people were injured. Here's another picture of the scene.

Steve: Yeah.

Leo: Wow. Holy cow.

Steve: The Lana River. So I guess...

Leo: That's not - I would have said Photoshop for sure.

Steve: Yeah. It's nuts. It's nuts. I mean, you needed another car there for the bus to drive over the car in order to get to the far side. And then they pulled the car out from underneath.

Leo: Crazy. Crazy.

Steve: So, wow.

Leo: Well, I'm glad the driver survived.

Steve: Anyway, thank you one of our listeners who sent this to me and thought maybe this would be a good Picture of the Week. And I agree.

Okay. So as I promised last week, there are two very important pieces of core AI technology that I wanted to discuss. And I also promised last week that we were going to do a deep dive into aspects of the operation of today's AI. Well, that turned out to be more true than I expected, so much so that it entirely crowded out the second of the two topics that I had planned to get to. But fear not, we're going to have another deep dive into that next second topic next week. And that's the role confusion topic, Leo, which you and I talked about at Black Hat, the research paper that I read during the flight there. And I said, OMG. Wow. Can that still be the way things are being done today? And I've confirmed, yes, unfortunately, it is.

Leo: Not as weird as a bus crossing the river, but it's close. It's close.

Steve: It's up there. So we're going to start by covering some important recent security events, and then get into understanding this first of the two core issues of the way AI works. Last Wednesday, Ars Technica's great security topic writer Dan Goodin reported under the headline, "Terabytes of credentials leaked in massive supply-chain attack," was Ars' headline. And of course that was the kind of thing I would have chosen to share a few years ago.

But the thing that raised my interest another several notches was the article's tagline, which read: "That data" - that is, the terabytes of credentials - "was scraped and exfiltrated from 2,500 users of a compromised AI package." So I thought, whoa, okay. Well, it turns out what's even more significant is we're not talking some random end users in Nebraska, you know, who no one knows. Wait till you hear whose credentials were among the more than 2,500 that were stolen.

So Dan writes: "Terabytes worth of credentials, many belonging to the world's biggest and most sensitive organizations, have been exposed in a supply-chain attack on LiteLLM, an open source tool that streamlines AI-driven software development. Microsoft, Amazon, Cisco, Samsung, and Salesforce are only a handful," he writes, "of the entities whose access secrets were exposed.

"The revelation was posted on Tuesday and Wednesday [of last week] by security firms CloudSEK and Hudson Rock. CloudSEK said it found keys, repository tokens, SSH keys, Kubernetes secrets, package publishing credentials, environment variables, and AI provider keys that could allow attackers to gain access to more than 2,500 organizations. And 40 minutes is all it took.

"The credentials were extracted during a 40-minute window in March while the victims used" - obviously unknowingly - "compromised versions of LiteLLM which had been downloaded from the package's official location in the Python Package Index repository." You know, PyPI. Hudson Rock said it made the discovery after analyzing a 195TB file that it had obtained. Neither firm identified the source of the information.

"The LiteLLM compromise was the result of a" - get this, Leo, this will ring some bells - "a previous supply-chain attack that infected the widely used vulnerability scanner Trivy." And remember that we had talked about this months before. "Other software infected in the campaign includes KICS and the Telnyx Python SDK. TeamPCP," which Dan describes as a "ramshackle but extremely capable gang largely made up of teenagers, took credit for the attack, and researchers have largely corroborated their claim.

"Independent security researcher Kevin Beaumont said: 'I've confirmed the data is legit, by the way, multiple victim orgs.' He said: 'It contains a significant volume of sensitive content at orgs. It's a massive supply chain breach due to poor AI security - not because

AI is the threat, but teens can now run circles around orgs obsessed with rushing out AI and poor DevOps security.'"

And I'll just explain that a little bit here. I'll take a moment. LiteLLM was compromised. It's not that AI was used in the compromise, it's that Kevin is saying that the way LiteLLM is used, the way it needs to be used, that is, to be a proxy for other AI services, means that you need to give it all your secrets so it can act on your behalf. We've talked about this fundamental problem previously, and a lot of orgs just got bit by it. Anyway, I'll have more to say here in a second.

So continuing, Dan writes: "The compromised versions of all four software packages contained" - right, four packages compromised by Trivy - "contained code that accessed the memory of infected machines, scraped its contents, and exfiltrated it through an attacker-controlled channel. The data is filled with an assortment of information." And again, 195TB, so it's like a wealth, but you've got to find the goodies in there. Interspersed in the wall of data are credentials to software pipelines maintained by tens of thousands of organizations that ran LiteLLM during the 40-minute span that the supply-chain attack remained active. In all, both security firms said some 434,000 CI/CD (continuous integration/continuous delivery) software pipelines had credentials exposed after running the compromised LiteLLM versions."

There were two versions that were compromised. I'll clarify that in a second. "In many cases, the researchers at CloudSEK and Hudson Rock had trouble identifying" - which is a problem - "the organizations the credentials belonged to. For instance, an email address in the dump from the domain @siriusxm.com ultimately did not indicate a breach at the satellite broadcaster, but rather within the infrastructure of SiriusXM's subsidiary AdsWizz. A trove of internal corporate secrets were found exposing sensitive tokens for platforms such as Salesforce (SALESFORCE_CLIENT_SECRET), Slack (SLACK_SIGNING_SECRET), and Microsoft Azure environments. The researchers had high confidence that the following organizations did have their credentials exposed."

Get this: "Nvidia, AWS, Samsung, Salesforce, Cisco, Hoffmann-La Roche, ServiceNow, Siemens, S&P Global, Airbus U.S. Space & Defense, John Deere, Regeneron Pharmaceuticals, London Stock Exchange Group, Thomson Reuters, FedEx, MediaTek Inc., Volkswagen AG, Deloitte, The Kroger Co., Siemens Energy, Thales Group, X Corp (as in Twitter), Zscaler, Epic Games, Orange S.A., HP Inc., Philips, Vodafone Group, Carl Zeiss, Deutsche Bahn, NGINX, BT Group, and Roku." I mean, wow.

Hudson Rock said: "Many CI/CD pipelines are configured generically. The dumped variables contain active database passwords, third-party API keys, and cloud credentials without any identifiable company email, custom domain string, or internal server name. This means that countless organizations [which they were unable to identify] currently, as in still, have active secrets sitting in this database. They are completely unaware of their exposure."

The research firms, I should note, immediately identified all the companies that they could among, you know, among those I just read. But lots of other companies, like 434,000 was it, different CI/CD pipelines, all their secrets are out there, and they haven't been notified because it's not clear who they are. So maybe that's safety, you know, I mean, the bad guys probably can't tell either. But it certainly gives you some starting credentials to use for some password guessing.

Finally, under the heading "Welcome to the new world of supply-chain attacks," Dan adds: "Both firms" - those two security firms - "are urging all organizations that used the compromised versions of LiteLLM, particularly those listed in the high-confidence section of the list, the organizations that are known, to thoroughly rotate all credentials in their pipelines. Hudson Rock instructed any organization that uses any AI proxy infrastructure,

third-party CI/CD vulnerability scanners, or downstream AI packages to immediately audit their environment for versions 1.82.7 and 1.82.8 of LiteLLM, which were the two compromised versions of the software.

"The firm advised those affected to perform 'aggressive credential revocation,' assume any secret accessible to the LiteLLM environment is compromised, invalidate and rotate all cloud keys, Kubernetes service account tokens, the GitLab/GitHub PATs, and audit logging and egress filtering.

"As a cautionary tale, CloudSEK said that Trivy developers rotated but failed to fully revoke an automation token over a 20-day window. That lapse gave the attackers a nearly three-week period in which to force-push malicious code to third-party builds that used the vulnerability scanner. As Beaumont observed, organizations' rush to integrate AI into their software delivery systems has also greatly contributed to the scale of the damage." Meaning just, you know, as we've seen. And Leo, remember when you immediately thought, hey, this - I can't remember the name of it. It was the package that came out the beginning of the year that was the code writing Claw.

Leo: Claude? What?

Steve: No, no. Claw. OpenClaw.

Leo: Oh, a Claw bot, yeah.

Steve: Yeah, so OpenClaw happened. You were excited about it, but you pulled back...

Leo: I didn't use it.

Steve: ...at the last minute, and it turned out that was a...

Leo: And I erased it.

Steve: Yeah, there was just - you had to tell it too much in order to allow it to do what you wanted to do.

Leo: Yeah, if you wanted to do anything, you gave it your email, you gave it money, you gave it your phone number.

Steve: Exactly.

Leo: A little dangerous.

Steve: So that, okay, in a final update to his initial reporting of this massive mess, Dan added: "There are already signs that some of the affected organizations are not taking the disclosure with the seriousness warranted. After this post went live," he writes,

"Kevin Beaumont reported: 'These creds date from about March. One of the orgs impacted told me," he writes, "they'd rotated them all, and it's a nothingburger." He said: "So I looked at their responsible disclosure policy. It allows trying creds, so I tried them all. Almost every one of them worked.' He said: 'I submitted a report. One of the biggest U.S. techcos.'" So someone said, oh, yeah, don't worry about it. We rotated our credentials. Nothing to see here.

Leo: They probably made new ones, but didn't delete the old ones, is what they did. Geez.

Steve: "So ultimately, the new revelations concerning the LiteLLM supply-chain attack underscore," writes Dan, "the growing threat of such campaigns and hence the importance of maintaining vigilance around the use of open source software that, when infected, can spread rapidly across the Internet.

"Alon Gal, co-founder and chief technology officer of Hudson Rock, wrote in an email: 'The key takeaway is how supply chains have evolved to make a single upstream breach affect thousands of companies simultaneously. A window of roughly 40 minutes in which the LiteLLM dependency was hacked led to over 430,000 instances in which millions of secrets were harvested. This magnitude pushes us into a completely new world regarding the type of response required from the cybersecurity industry.'" And lord knows we've been unimpressed, typically, by the kind of responses that we've seen historically.

So just to be clear, the cautionary takeaway from this is not a case, as I said before, of AI going rogue or any kind of AI misuse. I followed Dan's reference links back to one of Hudson Rock's reports, whose much more technical write-up of the breach makes what happened actually further clear. Their headline, Hudson Rock's headline, was "Largest AI Supply Chain Breach of 2026: LiteLLM Hack Impacts Thousands of Global Enterprises."

And Hudson Rock explains quickly: "The cybersecurity landscape is currently reeling from one of the most sophisticated multi- ecosystem supply chain campaigns publicly documented to date. Orchestrated by a threat actor group known as 'TeamPCP,' this cascading attack ultimately compromised LiteLLM - a widely adopted open-source AI proxy gateway - leading to the silent exfiltration of deep developmental secrets from thousands of continuous integration and continuous deployment pipelines worldwide.

"Excellent forensic research published by Snyk, Trend Micro, and Cycode has thoroughly detailed the mechanics of this breach. The attack did not begin with LiteLLM. Instead, TeamPCP first compromised the GitHub Actions pipeline for Trivy, a highly popular open-source vulnerability scanner. Because the developers of LiteLLM utilized Trivy in their own CI/CD pipeline, the poisoned security scanner was granted legitimate read access to their runner environment. This allowed the attackers to silently exfiltrate LiteLLM's PyPI publishing tokens. Armed with these credentials, TeamPCP was able to publish malicious versions of the LiteLLM package (v1.82.7 and v1.82.8).

"The payload delivery was exceptionally stealthy. By utilizing a .pth Python startup hook, the malicious code was executed the moment the Python interpreter initialized, regardless of whether the LiteLLM library was explicitly imported. The three-stage payload immediately began harvesting environment variables, local configuration files (like .kube/config and .aws/credentials), attempted lateral movement across Kubernetes clusters, and installed a persistent systemd backdoor.

"While the security community has deeply analyzed the malware's behavior, Hudson Rock has independently obtained the actual fallout: the raw exfiltrated data." That's that 182TB of data. I mean, talk about a huge amount of data in 40 minutes. I mean,

because there were that many instances of those two versions of the malicious LiteLLM that were updated from Python PI and installed and started, and all the credentials poured through it, and they sent them all off to some malware mothership somewhere. So they said: "This provides an unfiltered look into the massive scale of the compromise through the actual, raw files.

"Our researchers have obtained and analyzed a staggering 153GB RAR archive." You know, these files will compress way down. "This massive corpus contains exactly 433,909 files. Through our analysis, we have successfully attributed 118,829 CI runner dumps to 2,488 affected corporate domains. Whenever a developer machine, production server, or CI/CD pipeline executed the compromised LiteLLM package, the threat actors successfully harvested the live environment memory and configurations mid-execution."

Okay. So the real takeaway from this is that the massive adoption of the automation of development and delivery will tend to coalesce around relatively few most popular best-of-class tools. This naturally makes those tools a highly valuable target for attackers. Over at GitHub, the LiteLLM repository describes itself, writing: "LiteLLM is an open source AI Gateway that gives you a single unified interface to call more than 100 LLM providers - OpenAI, Anthropic, Gemini, Bedrock, Azure, and more - all using the OpenAI format. Use it as a Python SDK for direct library integration, or deploy the AI Gateway (Proxy Server) as a centralized service for your team or organization.

"Managing LLM calls across providers, you know, multiple, like Anthropic, OpenAI, Gemini and so forth, multiple providers, they say: 'Managing those calls gets complicated fast - different SDKs, different auth patterns, request formats, and error types for every model. LiteLLM removes that friction with a unified API - one interface for more than a hundred LLMs, no provider-specific SDK juggling, drop-in OpenAI compatibility - swap providers without rewriting your code; production-ready gateway - virtual keys, spend tracking, guardrails, load balancing, and an admin dashboard out of the box; 8ms P95 latency at 1k RPS, you know, requests per second in benchmarks.'"

So, wow; right? Sounds great. The only glitch here is that it must be totally trustworthy. In order for LiteLLM to be able to proxy for its users, all of those differing LLM backends, it must necessarily have every user's authentication credentials for every one of those different LLM backends. Just like, Leo, you were saying OpenClaw, you know, had to be able to log in as you, had to be able to access your bank records, had to be able to make payments on your behalf, blah blah blah blah blah. I mean, when we talk, when we get into proxies and agents, trust has to be there because, you know, they're acting on our behalf.

So what happened here is that the LiteLLM developers were users of the Trivy vulnerability scanner. And as we know because we talked about this back in March, when this happened, Trivy was compromised. This allowed attackers to compromise any project that was using Trivy as its scanner. And thus, in turn, those two versions of LiteLLM were compromised, giving attackers complete visibility into the credentials and domains of those 2,488 corporate users of LiteLLM during just that 40-minute window.

And, you know, I always like to try to suggest solutions to the problems we encounter, but I got nothin' because this is like a fundamental problem with the way we're doing things now. You know, we're now in a mode where everyone feels that they need to always be running the latest and greatest release of everything. Right? It was because of the updates to those two bad LiteLLM packages in the repository that this happened. Because there was a new version. Oh, got to have that.

So, you know, we've done this to ourselves. And I preach updating relentlessly; right? The entire security community is constantly pressing everyone to get better about updating their software. Stay current. Be sure you're receiving announcements of

important updates. Blah, blah, blah. You hear it here all the time. But in this instance doing that is precisely what bit the users of those two specific versions of LiteLLM. If they had NOT updated to those, if they had remained on a previous release, they would have never run one of those two malicious versions. But of course not updating doesn't work as a strategy, either.

Leo: No. I pin stuff, trying to avoid...

Steve: Yeah.

Leo: Because after this LLM debacle, you know, people said - it was caught within a day, I think, or a couple of days. They said, "If you just make sure you don't install anything that's less than three days old, you'll be all right." I made it 14 days because I figured, I mean, you're still not going to catch everything. But if it's a popular package, two weeks should be enough.

Steve: Yes.

Leo: So I'm very careful. The other thing I do, though, is I store all those tokens and secrets in Bitwarden's secret manager. So they're not - it's like my passwords; right? They're in a locked vault.

Steve: And we did just talk about this a couple weeks ago where we're beginning to see a new category of security software which is a means of allowing AI to access your credentials without it ever having them.

Leo: Right.

Steve: That is, so it says to Bitwarden, I need you to log in for me.

Leo: In effect, yeah. Or I guess a one-time token or, yeah. I mean, it's still going to - see, the problem is, so you're using that OpenAI endpoint

Leo: Of course.

Steve: which is connecting to somebody who's serving that AI. They want that token. So the token has to float around somewhere in memory. You don't want to write it to the hard drive. But it has to - LiteLLM would still have to be able to see it and send it. So I'm not sure a secrets manager would have protected me in that case.

Steve: Yeah.

Leo: You try to do what you can.

Steve: So the flipside, of course, of your two-week window, which is good, is that if there was a critical vulnerability that actually was authentically patched, then [crosstalk] getting it for two weeks.

Leo: Right.

Steve: So, you know, who knows what might be malicious and what isn't. We're in a bad place right now. The only winning strategy, or at least the best strategy that's available for the moment, is just to use the tools, but carefully monitor the news for the tools you're using. Stay current.

Leo: Yeah. Yes.

Steve: Stay current, but because mistakes are going to happen, the instant you learn of a breach that affects you, invalidate and recreate, in other words, rotate all possibly affected credentials. We've seen many instances where that was not done. You know, remember that part of what made LastPass's troubles even greater was that even after they learned that they'd been breached, they failed to fully cancel all previous authentication credentials. And, you know, and as a perfect case in point, we learned from Kevin Beaumont's test a major telco did not actually rotate their credentials when they had claimed to. So I guess there actually is an important and practical takeaway from this. I mean, like a real action item. We know...

Leo: Go ahead.

Steve: We know that things that are easily done tend to be done. And things that are a confusing pain in the butt too often fall into the, okay, I'll get back to that later. What we see is that the speed of modern attackers means that "later" stands a very good chance of being "too late." So here's my takeaway point from this. I think it's really crucial. If there's really no practical means of preventing inadvertent exposure to credential-leaking malware, and there may not be today, then rotating all of the credentials that any malware might have obtained the moment a credential compromising attack is known is important.

Leo: Of course.

Steve: Moreover, periodically rotating credentials preemptively on the "better safe than sorry" principle can be useful when the threat environment warrants it.

Leo: This is, by the way, this is contrary to the advice we've been giving about passwords, rotating passwords. But it is what you should do, which is rotating through them. So when I make keys now, I give them an expiration date of a month or two months or three months. And I know I'm going to have to rotate them because I don't get to keep using them.

Steve: Good. Right. So what I believe this means is that whole organization, organization-wide credential rotation needs to be both possible and easy.

Leo: Right.

Steve: If it's not easy, it won't happen, or it will be put off. So my best advice would be, if you're looking for a nice, self-contained, easy-to-describe project for an AI agent to tackle, a great investment would be to employ some AI to implement a comprehensive credential rotation facility for your organization.

Leo: Ah, good idea, yeah.

Steve: Make it automatic. Make it a single command that handles everything. Make it easy, even fun to use, and use it to maintain the credentials for both current and new services as they are being brought onboard. You know, as services are added and removed. And require its use for instantiating any new credential into use so that it cannot fall out of sync. Or, again, if that happened, it won't be trusted and used.

So I would say making, you know, automating credential rotation would be a cool - it's easy to describe to an AI. It's self-contained. You can see if it's working or not. The failure doesn't kill you, it just, you know, you've got to fix it. But I think it would be then incredibly useful. If when Thinkst Canary finds that some guy is in your network, the first thing you want to do is, you know, don't panic, as "The Hitchhiker's Guide to the Galaxy" tells us. Rotate those credentials. And, you know, and make it easy. Make it fun.

Leo: There are - there is a way to do this a little safer. I think if I were a big - or any business, I probably would be doing this. When we were at ArSec I met a few of these guys. They're an intermediary, credential capability layer. So they hold - they're a third party. They hold the credentials. You don't ever have the credential on your machine. You have a credential to them. And when you want to connect to an AI, you connect through them. So they have your credentials, and they do it, and you only get a one-time token. The problem, and the reason I didn't want to do that, A, you pay for it. But B, they have your credentials. And, you know, it has to be somebody you trust.

Steve: Right.

Leo: But they're going to have your credentials.

Steve: Right.

Leo: You know, it has, at some point, these credentials have to be exchanged. It's just like a password. I guess you could use hashing; couldn't you. Maybe that would be the solution.

Steve: Well, a couple weeks ago we talked about 1Password saying that they were introducing exactly this service.

Leo: Right.

Steve: So keeping it local. And of course the problem is that we're talking all credentials. So like that online service would be handling your credentials as a proxy for AI. But what about SSH servers? What about web, you know, all the other things that...

Leo: All those secrets.

Steve: So what you want is one thing that is just able to wipe the secrets out of your organization and replace them.

Leo: Yeah, that's the 1Password Credential Broker. That might really be the right way to do that, yeah.

Steve: We're going to have to have something like that.

Leo: Yeah.

Steve: Well, you know what we're going to have to have right now, Leo?

Leo: A break in the action? You're watching Security Now!, Steve Gibson, the man of the hour. Every Tuesday it's Security Now! day here at the TWiT Podcast Network. We're glad you're here with us. Now back to Mr. Gibson.

Steve: So France's recently celebrated ban on social media access for all children younger than 15 hit a bit of a snag. Last Friday, Reuters reported the following. They said: "France's top court on Friday blocked a bill banning social media access for under-15s, saying it infringed upon freedom of expression and delivering a setback for President Emmanuel Macron, who asked his government to rewrite the legislation. The bill would have barred children younger than 15 from opening a social media account beginning September 1st, and all accounts already open would be closed by the year end, which would also need to use age verification provided by the French privacy regulator.

"But," Reuters writes, "France's Constitutional Council found that the bill, while requiring everyone to give proof of age, failed 'to specify the conditions and limits' under which it should be provided, as well as infringing upon freedoms and privacy.'" So that's their report. You know, as we immediately understood when this began to happen in the U.S., you know, in the context of U.S. domestic legislation to control the viewing of pornography online, we noted that blocking anything for all users below a certain age inherently requires everyone's age to be known. You know, there's no way around that.

So France now needs to tackle the thorny problem of that inescapable infringement upon the Internet's illusion of total freedom and privacy. It is an illusion to some degree because we know how the Internet's technology works. Like from the beginning there's never been a greater infringement upon privacy than the abuse of third-party browser cookies to track people. But that went largely unseen. So nobody really worried about it.

The problem with age assertion is that, because it's explicit, and it cannot be hidden in the same way that cookies were, everyone's getting outraged. You know, the truth is that if we want our governments to restrict children's access to Internet content, then

everyone's age must be known by someone somewhere, either by every source of the proscribed content, or by every means of accessing that content.

So anyway, Reuters continues, quoting the Constitutional Council's statement, writing: "The Council holds that the contested provisions, on the one hand, disproportionately infringe upon the freedom of expression and communication; and, on the other, fail to provide the legal safeguards necessary to ensure the right to respect for private life. French lawmakers," they wrote, "had approved the bill in July" - which is when we first talked about it, month before last - "becoming the first in Europe to follow Australia, whose world-first ban barred access to platforms including Facebook, Snapchat, TikTok, and YouTube for under-16s in December. Lawmakers there are considering stricter penalties after data showed mixed success.

"Countries around the globe, including China, the UAE, and Turkey, have either instituted measures intended to curtail or bar access to social media for young people, or have said they're planning them. The European Union has said it was planning to seek stronger protections for children from harmful social media features.

"Social media companies generally oppose blanket bans, saying they have measures already in place to protect younger users, including age restrictions, though they have also said they would comply with government bans. Google, Meta, Snap and TikTok did not reply to Reuters' requests for comment.

"Macron, who in April urged teenagers to turn off their devices and read" - that went over really well - "in order to become better citizens, has ordered Prime Minister Sebastien Lecornu to rework the draft legislation to take the Constitutional Council's concerns into account. They said in a statement that they were determined for the reform to take effect before the spring of 2027, when France holds its presidential election."

So anyway, Macron has not given up. The draft legislation is being hastily reworked to address the Council's concerns since Macron still hopes to have this reform in effect as soon as possible, and he was also going to add smartphone restriction to the legislation that would take effect for high school, in addition to the lower schools. So anyway, we will see what's going on and what happens, Leo. Wow.

Leo: Hmph. Bah humbug. Tout a l'heure.

Steve: So last Tuesday, WIRED reported on the unnerving discovery of a serious vulnerability in the Zoom teleconferencing system. And of course now that's, like, recent. We'll all remember when Zoom really became a big deal at the beginning of COVID because, you know, it saw its adoption soar as teleconferencing became super important. It was the only way to continue doing business if you were stuck at home. And boy, Zoom had a bunch of early problems. They weren't all resolved, as it turns out. WIRED's headline reads "A Zoom Screen-Sharing Bug Let Anyone Take Over Other Devices on a Call." And their brief teaser is what makes this so interesting. They said: "Researchers say it took fewer than 20 prompts for a public AI tool to find a flaw [which has now been fixed] allowing anyone on a Zoom call to hijack other participants' devices."

What wasn't clear from the reporting, and I didn't dig deep into it, was, wait a minute, a public AI tool was used to do some sort of clear cybersecurity work without hitting guardrails?

Leo: Probably a Chinese model.

Steve: Ah, could be. Oh, good point. Publicly available, yes. So WIRED said: "As AI models gain advanced capabilities to find vulnerabilities in software, develop ways to exploit them, and even carry out autonomous hacking sprees" - all of which we've been seeing - "researchers offered a sobering new example on Tuesday, disclosing vulnerabilities in the video conferencing platform Zoom that could have been exploited to take over targets' devices. Anyone on a call that involved screen sharing, whether participants or the host, would have been vulnerable to a silent attack that could be carried out with no indication and no interaction from the victim.

"Researchers from the digital defense firm A Security" - just the letter A Security - "say the bug was discovered in early June using publicly available AI models, and that it took fewer than 20 prompts to uncover the vulnerabilities and create a working attack. Zoom issued a security advisory Tuesday, including details about the fixes the company has already begun rolling out to address the flaws, which affected devices running across all operating systems that Zoom supports - Windows, macOS, Linux, iOS, and Android.

"A Security cofounder" - the company A Security - "cofounder Omer Gull told WIRED ahead of the disclosure: 'What's interesting for us and what we believe is dangerous, is the democratization of these capabilities. The barrier to entry is dropping rapidly. Before, it would have taken a team of five people maybe six months with a lot of refining and iteration to find this. Now people can reach the same results with fewer than 20 prompts. And Zoom is an important type of target because people assume trust when using it. They don't see it as a threat,'" his quote ends.

"The vulnerabilities," writes WIRED, were found in the protocol used to facilitate real-time annotation during screen sharing. The researchers say that their AI bug hunting systems specifically delved into this component because, like human bug hunters, they've been trained that convoluted and obscure functions often contain overlooked vulnerabilities. This is particularly true with proprietary, closed-source software. An established company like Zoom presumably does extensive code review and vetting on all components and functions; but without the benefit of public, open review, esoteric yet complex features like annotation are more likely to contain mistakes. Zoom did not respond to multiple requests for comment from WIRED about the A Security findings.

"The bugs are now patched, with Zoom issuing both server and client-side fixes, or patches for both Zoom's own servers and the applications that run on customer devices. But the researchers emphasize that it was alarming to contemplate bugs that could have been exploited to take over a target device simply by getting someone on a Zoom call. Joining a call is in itself a gesture of trust. But given how ubiquitous video calling is in both personal and professional contexts, and given that Zoom in particular is also widely used for events and semipublic activities like webinars, people typically have their guard down when joining a Zoom.

"A Security cofounder Yossi Torati told WIRED on a call: 'If you just get on a Zoom with us, we can take over your device. The worst-case scenario is that we can take over an enterprise just by having this capability in our hands. If I'm an attacker, I can be on a call with someone from a company, take control of their computer and their credentials, and then use them to move laterally across the enterprise.' Practitioners often call security a 'cat-and-mouse game'; but as AI bug hunting proliferates, this delicate dance has become an all-out race." Which of course is exactly what we've been seeing.

All indications are that the high-tech computer world, at every level, from IoT embedded device to consumer PC and enterprise, probably is heading for a rough patch. So far, as we know, I've been a cheerleader for the "fixing the bugs" team; you know? And the good news has indeed been that an astounding number of bugs are rapidly being found and fixed. You know, in those last two versions of Chrome, more than a thousand total between those two, 149 and 150. But that's also the bad news because the fact that so

many latent problems are being found tells us that the software, at every level, that we've been living with for years, has been demonstrably riddled with previously unknown flaws.

So the best that can be said is that the future remains stubbornly uncertain. We're getting the bugs out of the software. And, you know, my feeling is, absent a state actor having some reason to attack another country, money is still what drives the bad guys. Now that we've got cryptocurrency, now that we've got the ability to exfiltrate and extort, money is the motive. And so there's not money behind a mass casualty event. There's money behind selectively targeting, exfiltrating, blackmailing, extorting, and getting what cash you can. So I don't think we're going to see a big Y2K-style, I mean, yeah, a Y2K apocalyptic sort of thing. To me that doesn't make sense because it doesn't make money.

And that's what, I mean, that's kind of a saving grace. It means that there will be people hurt. But their, you know, their insurance is going to go up, and they're going to be paying out of pocket for bad guys having gotten into their system, using bugs that are probably latent and aren't their fault and are zero-days, so there's really nothing they could do about it. So hopefully that's what we see, and that over time the ability to do that dries up because we get our software fixed.

Okay. One last piece before we get into our big topic. There are some folks who are, I would argue, justifiably profiting from all of this. Bad guys not justifiably profiting, but good guys. Last Monday, following the Black Hat conference, CNBC reported, they said: "Cybersecurity stocks of CrowdStrike and Palo Alto Networks jumped more than 5% to new highs on Monday on renewed demand for artificial intelligence security tools following the industry's annual Black Hat conference in Las Vegas." In other words, those guys were there. They were showing off that their AI is going to be used, in fact they're ready to deploy defensive AI solutions. And the world said we need some more of that.

"Analysts at BTIG" - which is a large financial services firm - "wrote to their clients in an internal firm letter: 'The single most consistent theme across our conversations - partners, vendors, and customers alike - was that AI agents have fundamentally changed the threat landscape. While AI agents have become the predominant attack threat and the environment is 'meaningfully worse,' deployment and AI security tools are only in the early innings."

CNBC said: "Businesses are turning to cybersecurity companies for new agentic tools to fend off adversaries in a hyper-accelerated threat landscape fueled by new cyber models. Executives and potential customers gathered in Las Vegas last week in search of answers and ways to secure systems from rogue AI agents.

"BTIG wrote: 'We think AI is creating a new modernization cycle in the endpoint security space, which directly benefits CrowdStrike's core business.' Analysts at Cantor said: 'AI has moved from being a cybersecurity feature to a key pillar of both the attack surface and the defender infrastructure.'"

So I remember the first time we touched on this. I think it was one of our - it was, it was one of our listeners who was commenting that his company was already using some AI-based systems. I think he was on AWS Cloud, and he was using a third-party's AI-based systems. And I was - this was a couple weeks ago. And I was like, wow, this is happening already. I mean, we've got AI being deployed for defensive purposes, which is wonderful.

And you know, Leo, we're going to now dig into what I have learned recently about AI and can share. But first I think we need to hear...

Leo: To share a fine sponsor, perhaps?

Steve: I think that'd be perfect.

Leo: I think I can do that. Our show today, I'm excited about all of this. Yeah, have a beverage. Looks like Tang. Are you drinking Tang? Oh, that's right, dilute orange juice.

Steve: Yes, exactly.

Leo: Do you add Vitamin C to your diluted orange juice?

Steve: No.

Leo: Okay. I'm just curious.

Steve: I take 15 grams of C a day, so way more...

Leo: You get plenty. I know. I know. 15 grams.

Steve: Yup.

Leo: Holy cow.

Steve: Three divided doses of five grams each.

Leo: That's half an ounce. You're crazy. You're a madman. At least you, well, I don't know. Whatever that means. I don't know what Vitamin C is doing for you. You don't get any sore throats, I guess.

Steve: My liver would like to be generating about 20 grams a day, and it can't because it's got - the human genome has a little glitch. So I'm [crosstalk].

Leo: Right. We talked about that, yeah.

Steve: Yeah.

Leo: Damn genome. But now back to Steve and Security Now!.

Steve: So, okay. As I promised last week, there are two important and fascinating pieces of core AI technology I want to spend time on today. I got to one of them. The first is a solution to what's been dubbed "the dual-use" problem. That's the fancy name given to the fact that most knowledge can be used in ways that we both want and don't want. And of course that's no surprise, right, since that's always been true of knowledge. So what is it about AI that changes this?

Anyone who's spent any time with any of the recent state-of-the-art AI chatbots will have come to appreciate that what we already have today is, at the very least, an over-obliging conversational partner that can barely restrain itself from being oh so very helpful. And that tail-wagging puppy happens to also have access to the world's stored knowledge. So it's not that it was impossible before AI to obtain that knowledge the old fashioned way, you know, by researching, reading, learning, and understanding.

No, the difference is that friction matters, and AI chatbots have hugely facilitated the access to that same knowledge by nearly eliminating all of the work that was previously required to gain such knowledge. And that's incredibly valuable. That's what underlies all of this frenzied hyperscaler data center build-out. Investors believe, with good reason, that offering knowledge at our fingertips by phrasing a question is something most of the world will pay for. The fact that AI Chatbots now have just shy of one billion users strongly suggests that these investors are not wrong. For myself, I'm completely spoiled. Even though I've never turned any agent yet loose on anything, Claude is my go-to for quick answers. It's an accelerator for me. And don't let anybody know, but at this point I would pretty much pay anything for it.

Leo: Shhh. Yeah, I know. I know how you feel. I know how you feel.

Steve: Oh, my god.

Leo: I know how you feel, yeah. I have paid anything for it.

Steve: But Leo, you and I already know we're not going to have to because, as I mentioned I think before the show, it turns out that that Lenovo ThinkStation machine that I bought had a strong GPU. It's got an NVIDIA RTX 2000 with 16GB, and there are useful models now.

Leo: Oh, yeah.

Steve: That can run in that.

Leo: Yup.

Steve: So, but I'm, you know, still, you're always going to want the latest and greatest and the best after...

Leo: It won't be as good as Claude. I mean, that's the thing.

Steve: No, no.

Leo: And not yet, next year. And then you could stop. Then you'll be happy; right?

Steve: So by comparison, right, anyone could download the Encyclopedia Britannica or the unabridged Oxford English Dictionary. But you can't ask them questions.

Leo: Yeah.

Steve: It's all just dead knowledge lying there, so it's much less entertaining, and it takes a lot longer. So, you know, and using them is back to old-school researching, reading, learning, and understanding. And of course look behind me. Anybody who has seen a video of this podcast is aware that there's a solid wall of textbooks, and as it happens, up there just out of camera is an unabridged Oxford English Dictionary, 27 volumes.

Leo: Yup.

Steve: I cannot recall the last time I opened any of those books back there.

Leo: You don't need to.

Steve: You know? You're right. AI neural networks are our new store of knowledge. Incredible as it may seem - and it would have seemed like science fiction just a few years ago - the collection of just an array of scalar variables, which specify the scaling weights of the inputs to a vast neural network, creates a representation of the expression of all of the knowledge that has been trained into that model.

And I said that exactly the way I wanted to, "creates a representation of the expression of all of the knowledge that's has been trained into that model." I think it's important to frame it that way. What we're feeding into the neural network while it's being trained is the expression of the knowledge, largely gleaned from the Internet, as well as from reference texts which AI companies have been quietly purchasing and ingesting, to create an overall training corpus.

The end result of this - and this is still mind-boggling to me - is purely and simply a next-most-likely token prediction engine. Leo, as I've said, your very early initial observation was that what we were calling AI, and this was a couple years ago, was little more than fancy spell check.

Leo: I called it spicy autocorrect.

Steve: Yes. And then, as our listeners will remember, last week I quoted a nearly stunned Matthew Green specifically saying that it's not just fancy spell check, but its essence has not changed. You know, you were not wrong, Leo, then. And Matthew is also not wrong today. So how do we explain this apparent disparity?

It's that over the past couple of years, what was initially a simple, next-word predicting "spell check" has become really really really really fancy spell check. Deep underneath even today's astonishingly apparently intelligent reasoning systems, down at their core is still just a neural network that only does exactly one thing: Given a long, and in many cases astonishingly long, token sequence, it predicts the most likely next token. And the fact that we get what we now get from that, well, that's what's still mind-boggling to me.

I'm going to spend a bit more time on this because a deeper understanding of the truth of what's actually going on with today's AI I think will help everyone to appreciate next week's, the second of the two research papers I plan to share, which is, well, I don't know to sum it up quickly. So stay tuned.

So your original, Leo, "it's just spell check autocomplete" summation, that has its roots in AI circa 2020. Way back then, if you were to carefully phrase a question to GPT-3 such as "The capital of France is?," it would be able to complete the sentence by emitting the next expected word "Paris" because that model, the GPT-3 model's vast statistical dataset made Paris the next most likely word. The capital of France is Paris. But if you used question-style phrasing back then, "What is the capital of France?" that would not have been met with the same success. So what happened? Obviously we have that now. How did we turn these models from autocomplete engines, that is, that's all they could do, into conversationalists?

So it took a few years of experimentation, but AI researchers first used something that's now known as "instruction tuning." They took the knowledge-trained model and fine-tuned it on a - and this is, again, another surprise - a surprisingly small set of human-written examples of what good responses would look like. After that, then, something known as RLHF - Reinforcement Learning From Human Feedback - was used. So this applied preference rankings where, again, humans compared and ranked multiple outputs from best to worst. And that ranking was used to train a reward, which pushed the model toward the better behavior.

Now, here's the astonishing part to me. Researchers then found that they could employ a comparatively small model of these "query and ranked-response" samples, which then created reward feedback, and that these large language models would, and did, with startling speed, they were able to generalize from the language patterns of queries and responses across their entire knowledgebase. That is, they learned that pattern and generalized it.

So to better appreciate the scale of this, a model that had been trained on trillions of tokens of raw knowledge, you know, this created the original statistical autocomplete engine, a neural network that just could probabilistically choose the next most likely token. It could then be rewarded using only - again, it was originally fed trillions of tokens. It could be rewarded using only tens of thousands to low hundreds of thousands of query/response samples, or examples, and the behavior of the entire model would be reshaped across all of the knowledge that it contained.

This is a well-documented fact now that this occurs. And the fact that it occurs with such sample efficiency is what was utterly unexpected. I mean, this was the surprise. Today, now, as I said, it's a well-documented real phenomenon which has now been termed, we have a term for it now, it's known as the "superficial alignment hypothesis," the idea that all of the raw knowledge was already there from the model's initial knowledge corpus training. Then a relatively miniscule bit of fine-tuning teaches the model format and behavior, but not new knowledge. That is, it wasn't giving it new knowledge. It was reformatting it and giving it behavior for the first time.

So to state that a bit differently for clarity, the breakthrough about four years ago was not the use of a larger model than we had at that time. That is one of the things that's

been happening since then. But the breakthrough was the unexpected discovery that a comparatively infinitesimal dose of human- provided "here's what a good answer looks like, and here's which of these answers is better" training would reshape a massive already trained model's entire behavior, turning it from something that completes patterns into something that acts like it's trying to be helpful. So once the massive model learned what "helpful" looked like and that its trainers wanted it to look like that, all of its stored knowledge was immediately available in that new "helpful" format, and we got helpful chatty AI. And that's how it happened.

And as we know, those first steps made by ChatGPT were more than a little shaky. So researchers realized that their newly birthed Chatbot would need some additional post-training "alignment," as it's now being called. This began as the RLHF (Reinforcement Learning from Human Feedback) that we talked about. But then, a year later in 2023, a handful of AI researchers at Stanford University published a paper titled: "Direct Preference Optimization." The paper's title is "Direct Preference Optimization: Your Language Model Is Secretly a Reward Model." And since all of the descendents of this, it's now known as DPO system, Direct Preference Optimization was sort of the granddaddy, now we have further refinements of that which have occurred in the last three years - something known as IPO, there's KTO, ORPO, and SimPO, they all descend from DPO - I want to share just the Abstract of the Stanford researcher's original paper, which was the breakthrough beyond that earlier reinforcement learning from human feedback.

So they explain their invention by writing: "While large-scale unsupervised language models learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels, you know, feedback of the relative quality of model generations [model output] and fine-tune the unsupervised language model to align with these preferences, often with reinforcement learning from human feedback (RLHF).

"However," they write, "RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model.

"In this paper we introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form" - I have no idea what that means, but you'll get a sense for this - allowing us to solve the standard RLHF problem with only a simple classification loss. The resulting algorithm, which we call Direct Preference Optimization, is stable, performant, and computationally lightweight, eliminating the need for sampling from the language model during fine-tuning or performing significant hyperparameter tuning." Whatever that is.

"Our experiments show that DPO can fine-tune language models to align with human preferences as well as or better than existing methods." Actually, it's vastly better. It completely obsoleted everything that came before. "Notably, fine-tuning with DPO exceeds PPO-based RLHF in ability to control sentiment of generations, and matches or improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train." So, okay. That's just their abstract. And then the paper goes on at length, like with crazy formulating.

I wanted to share that because I didn't want to leave everyone with the belief that the original RLHF - Reinforcement Learning from Human Feedback - approach which began this transformation from autocomplete to actually Q&A, that that's what the industry was still using. As I said, it was the genesis. Stanford's DPO (Direct Preference Optimization) dramatically improved it, and DPO's success, as I said, spawned a number of successors which I cited earlier.

Okay. So what is all this about? It's about how we take a massive neural network which has been trained on and contains raw knowledge, which can initially only be used to predict the next most likely token, and impress upon that network actual behavior. That's what's changing here. We're giving this knowledge base a set of behavior. The first example of behavior was turning the network into something that could actually respond to queries. That gave us the first interactive AI. But as I noted, those first steps were somewhat shaky. So over time, researchers learned that by using the technologies I've just described, they could improve the model's instruction-following behavior so that it would answer the question that was asked as well as respect the requested format, and length, and language.

The model's style and tone could also be modified and tuned to improve its response formatting, structure, hedging, politeness, and its own verbosity. These factors were, again, they were given significant weight in the tuning. As we saw in the early days, sycophancy was an often-seen problem. This arose because we humans reliably prefer being agreed with, so preference optimization trains models toward agreement, and it takes deliberate counter-effort to prevent it now. So that's now in place. We've been able to kind of get that under control. Another improvement that was impressed upon models was factuality, preferring responses that will admit uncertainty rather than always confident fabrication.

Leo: Well, that explains a lot because hallucination, at least in my experience, has almost disappeared. And I was wondering how they did that. Now we know, RPO.

Steve: Exactly. So the other class of behavior that can be trained into a model is its refusal to provide certain classes of information. And what's significant is that this exists in two places. This is different than guardrails. There's filtering what you allow the human prompter to get into the model, and then filtering what you allow the model to show of its results. So that's real-time filtering input and output. But the other place is you can actually train the model itself to refuse, independent of input/output filtering. So, you know, which is to say the harnessing of the model.

So it's one thing for a model to contain knowledge that's dual-use, which only authorized users should be able to access; but some model behavior and/or knowledge dissemination should be proactively prevented. The test AI designers apply is termed "uplift." That is, does a model meaningfully advance someone's capability beyond what they could already get? You know, does it uplift them? This falls into two categories. One is where the knowledge is the harm, and the other is where the output is the harm.

Examples of knowledge uplift would be, for example, bioweapon synthesis routes, nerve agent production, and nuclear device design. You know, the relevant literature is scattered, it's partial, and it's difficult to assemble. So any model that would synthesize it into an actionable protocol provides genuine uplift toward mass casualties. And the asymmetry is clear; right? Defensive work in these fields does not require the synthesis route. Somebody, you know, a vaccine researcher, needs to understand pathogen biology, not a means of enhancing a pathogen's malicious use.

So this list and those examples that I cited, you know, would not take anyone by surprise. They're one category where essentially every AI model provider refuses regardless of credentials or system prompt. That is, it's not about who you are, what your privileges are on the model. You just can't have that. It's been trained, the model itself has been trained to refuse.

Leo: Trained in what way? Do they not have the information?

Steve: We're going to get there.

Leo: Oh, good.

Steve: That's about the perfect question.

Leo: By the way, this is fascinating. You know when the world became aware of this, I mean, the knowledge of this has been around, the papers and so forth, for some time. But it was when DeepSeek came out from China, the first - it was the first model to use RLHF, as far as I know.

Steve: Yup.

Leo: And it was an eye-opener for people because the model was stunning. This was January of last year. I remember it very well. It was a DeepSeek moment. That's when all the stocks of all the AI companies plummeted because people said, wait a minute, the Chinese can do this cheap.

Steve: Yup.

Leo: It was very interesting. Now everybody uses RLHF, of course.

Steve: Yup.

Leo: Yeah. Really interesting.,

Steve: Now actually would be a great time to take a break. We're at a little after an hour and a half in. So we're going to pace ourselves.

Leo: This is fantastic, yeah. And you found this by reading papers?

Steve: Yeah. Yeah. I found...

Leo: On Archive.org?

Steve: Yeah, they're all there.

Leo: Yeah, yeah. Yeah, Jeff loves reading those papers, too. There's a lot of garbage there, too. That's my only - but you know what to look for.

Steve: This is like, yeah, these were the founding...

Leo: These are the seminal papers, yeah, the founding documents of AI.

Steve: The founding seminal work, yeah.

Leo: Yeah, absolutely. Fascinating stuff.

Steve: Yeah, like three AI researchers at Stanford that figured out, you know, how to improve on RLHF so that that's what everyone is using now.

Leo: Right.

Steve: That's what you want to look at.

Leo: It's, you know, the other thing that I find amazing is there are breakthroughs like this happening almost all the time now. That there are researchers all over the world working as hard as their little research brains can to find new techniques.

Steve: Yes. That's why I keep saying "today's AI, today's AI." I mean, anybody looking back, well, like, just quarterly, look back in three-month hops, and it's clear we're on to something.

Leo: Oh, yeah. And the other thing is that I from the outside can - I can put flags in the ground exactly when, you know, January 2025 is when DeepSeek came out. And everybody said, oh, RLHF, oh oh oh.

Steve: And last November...

Leo: November 2025, when Opus 4.5 - we're I'm sure going to get to that. So these internal progress shows up in ways that those of us who use this heavily...

Steve: Milestones. Milestones, yeah.

Leo: ...can see those milestones, the impact of those milestones. I mean, it's very clear. You don't see hallucinations like you used to. And I don't know why. I'm glad you're explaining this. You also see sycophancy going down a little bit, although it's - this is the other thing is that companies are loathe to get rid of the stuff that makes people like me keep using their products; right? So they're not going to get rid of all of that. You're not going to get rid of all of that.

Now, I'm a big fan of Steve Gibson, who is finally explaining something I've been using for a year or two and had no idea what was going on under the hood.

Steve: We all have been.

Leo: Yeah, it's fascinating.

Steve: So knowledge is, you know, forbidden knowledge is one class. The other is where the output itself is the problem, or the harm. So an example of that, you know, which carries universal condemnation, would be text which sexualizes minors. Models will not produce any such text. It's trained out of them.

Leo: See, that's fascinating because we've heard about classifiers, which kind of are gates...

Steve: Right.

Leo: ...preventing of the egress of that information. But this goes deeper than that. The models themselves say no, no, no.

Steve: Yes. And that's why, unless you remove that, even without any kind of a harness, even without anything that is filtering input and output, the model says, sorry, I cannot help you there.

Leo: Right.

Steve: So as we've seen earlier, the good news is that these large language models are astonishingly able to absorb and embrace. We saw this in like their ability to learn to answer questions, I mean, to understand the linguistic nature of a question and how to apply the knowledge they had to create an answer. I mean, it's an astonishing technology. But it does this. So we're able to give them, as a consequence, what appears to be a personality, many forms of behavior, and also instruct them what they may and may not do. So that's the good news. The bad news, as it turns out, is that any behavior like this that can be easily imprinted can also be easily removed.

In the summer of 2024, researchers at ETH Zurich, the University of Maryland, Anthropic, and MIT published a paper titled "Refusal in Language Models Is Mediated by a Single Direction." And here "direction" is a term of art in neural networks, as we'll see. So the Abstract of their paper employs some of this inside baseball terminology, but everyone should be able to easily get the gist of it. So, and I'll explain a bit more afterward.

The paper's Abstract, that is, "Refusal in Language Models Is Mediated by a Single Direction," the Abstract reads: "Conversational large language models are fine-tuned for both instruction-following and safety" - safety meaning I'm not going to tell you that - "resulting in models that obey benign requests but refuse harmful ones. While this refusal behavior is widespread across chat models, its underlying mechanisms remain poorly understood." Again, this is summer of 2024, so just about two years ago this paper appeared.

"In this work, across 13 popular open-source chat models up to 72 billion parameters in size, we show that refusal is mediated by a one-dimensional subspace. Specifically, for

each model, we find a single direction such that erasing this direction from the model's residual stream activations prevents it from refusing harmful instructions, while adding this direction elicits refusal on even harmless instructions."

They said: "Leveraging this insight, we propose a novel white-box jailbreak method that surgically disables refusal with minimal effect on other capabilities. Finally, we mechanistically analyze how adversarial suffixes suppress propagation of the refusal-mediating direction. Our findings underscore the brittleness" - and this is the key - "the brittleness of current safety fine-tuning methods." In other words, just instructing the model not to answer the question, well, that works. But if the weights are open, it turns out to be trivial to remove those instructions, even after the fact, and not having known what the instructions were. I'll explain a little more. It's just amazing. So they finish, saying: "More broadly, our work showcases how an understanding of model internals can be leveraged to develop practical methods for controlling model behavior."

Okay. So what this group discovered was that any late-term model behavior imprinting can later be removed from such models. And the way this is done, as I said, is wonderful and wild. They compare the model's activations on harmful versus harmless prompts, compute the mean difference, then project the weight matrices orthogonal to that direction. So essentially they deliberately ask it questions it is trained not to answer. And they watch some of what it does, some of where the activations are, compared to asking questions that it's happy to oblige. And they're able to take the difference in the activations, see them, and then apply a remover that suppresses that, and suddenly it will now answer all questions.

So after they do that, the model loses the ability to represent, and therefore to execute, on its refusal. And what they found was that this was surgical. It specifically disables completely 13 completely different chatbots, all of their refusal, while having minimal effect on other capabilities. So...

Leo: It's kind of like a functional MRI on the model; right? You're looking for what got activated.

Steve: Yes. And then remove it.

Leo: And then excise it. The only thing I'm worried about, you know, for instance, I use a model from China, Qwen, as we mentioned, 3.8.27. And I've seen abliterated versions of it. But people say, well, you also have a risk. This is brain surgery. After all, you might make it dumber.

Steve: So I can't speak to that, but I can cite the research...

Leo: Okay.

Steve: ...because they do address this. So, okay, so first, from my lay view it appears clear that since little training, amazingly little training, was required to imprint that original refusal behavior...

Leo: Ah. Now I get it.

Steve: The impact upon the model was minimal. You know, it wasn't diffused throughout the entire model. It had a - but still it's significant that this is able to change its behavior so quickly.

Leo: Think about it. You're modifying with just a few inputs.

Steve: Yes. Yes.

Leo: A large model. You know, there's going to be some collateral neurons that are affected.

Steve: Well, and actually I had used the analogy a little bit later of clean margins when a surgeon excises something.

Leo: Right, right.

Steve: Anyway, so these guys discovered how to identify the changes created by that training. Which, again, wasn't pervasive. It was, you know, not that much training did comprehensively change the model's behavior. So it turns out it could be removed.

Okay. So, okay. For what it's worth, when you encounter the term "abliteration," this is what is meant. Not "obliteration," "abliteration." Just to put a final point on it, a Hugging Face blog posting in the summer of 2024, that is, you know, following this research, was titled "Uncensor any LLM with abliteration." And I'm going to share just the intro from that posting to give everyone a sense for where that earlier research led, which is here.

The blog says: "The third generation of Llama models provided fine-tunes (Instruct) versions that excel in understanding and following instructions. However, these models are heavily censored, designed to refuse requests seen as harmful with responses such as 'As an AI assistant, I cannot help you.' While this safety feature is crucial for preventing misuse, it limits the model's flexibility and responsiveness.

"In this article, we will explore a technique called 'abliteration' that can uncensor any LLM without retraining. This technique effectively removes the model's built-in refusal mechanism, allowing it to respond to all types of prompts. The code is available on Google Colab and in the LLM Course on GitHub."

So, and it says: "What is abliteration? Modern LLMs are fine-tuned for safety and instruction-following, meaning they're trained to refuse harmful requests. In their blog post, Arditi et al." - and that is the previous research that I was referring to, the research in the summer of 2024 - "Arditi et al. have shown that this refusal behavior is mediated by a specific direction in the model's residual stream. If we prevent the model from representing this direction, it loses its ability to refuse requests."

So that Arditi reference, as I said, is the research I referred to previously, which showed the world how to do this, just how to simply perform this excision. The blog posting on Hugging Face is one artifact of that preceding research. And the companion repository on GitHub contains all of the details. That is, it's mlabonne.github.io, and it is "Uncensor any LLM with abliteration." And as you would expect, it was all tremendously exciting to the world's AI hackers, many of whom immediately jumped on any and every published and available open weight model and happily abliterated away any and all perceived and

imposed censorship upon those models. And those "abliterated" public open-weight models are now all available for use by anyone who can harness them. And as you said, Leo, you've seen them, both with and without the censorship abliterated from them.

So this finally brings us to the first major topic I wanted to discuss today now that we have a much deeper understanding of where these chatbots came from, how they work, how they can have behavior imprinted upon them after they've been filled with raw knowledge, and how unfortunately brittle that imprinting is. So, okay. Now, I also need to mention that the term "pretraining" is what the AI industry has unfortunately landed on to actually mean training, the training that occurs before the post-training.

And I suppose, since there is now always going to be a definite post-training phase, during which behavior is layered on top of the previously trained-in knowledge, you know, if that "pretraining" were just called "training," which is actually what it is, then it might be assumed to encompass both the training and the post-training. Meaning if it was just called "training," it would mean all training.

So my point is, there's pretraining, and there's post-training, and there is not any just training in the middle. We don't use that. So now we know that pretraining - that's what it's called. Pretraining is the initial knowledge corpus training phase. Now that we know that, I can explain that a small competent team of AI researchers at AE Studio, working with Anthropic, they've proven the feasibility of a new form of AI model pretraining. That is, a new way of doing that initial knowledge capture.

Last month, both AE Studio and Anthropic blogged about it, and they published a joint research paper. I'm going to start by sharing Anthropic's press-release style posting since it provides the essence of the research without dragging us too far down into the weeds. Under the title of, which is their title, "An off switch for dual-use knowledge in AI models." And again, the issue here is dual-use; right? Nuclear bomb generation, you know, creation. Well, there's some knowledge that we want to be able not to provide. The problem is we've seen how brittle the instruction of Do Not Provide That, given post-training, is. It can be removed. And it has all been removed. It's been abliterated from all of the current open weight models.

So here is what Anthropic said. And they start by saying: "This post describes research conducted by AE Studio in collaboration with Anthropic." They said: "A frontier AI model is, among other things, a large store of knowledge. Some of that knowledge is dual-use, meaning it can be used for good or bad. For example, knowledge of cybersecurity can help patch critical security vulnerabilities, or it can be used to exploit them. Knowledge of virology can help a researcher create a vaccine, but it can also help a malicious actor design a deadly pathogen.

"Ideally, we would be able to balance three separate goals: first, limiting access to dual-use capabilities in as surgical a way as possible; second, allowing trusted users to access those same capabilities for beneficial purposes; and, third, doing all this without affecting the model's performance on any other task.

"Current safeguards are imperfect," they wrote. "We train models to refuse harmful requests and use classifiers to screen inputs and outputs for dangerous content. These layers of protection guard against dangerous outputs, but they don't change the knowledge stored in the underlying model. Despite our safeguards, a sufficiently determined attacker may still try to jailbreak the model, working past its defenses to access the dual-use knowledge.

"A more robust protection against misuse would be to control what the model knows. We've explored this before," they wrote. "In earlier work, we filtered information about chemical, biological, radiological, and nuclear weapons out of pretraining data, and later

showed that dual-use knowledge can be confined to a removable slice of a model's weights. But filtering is a blunt instrument. It produces one model with one fixed set of capabilities. Using filtering, if you want a model version that can discuss advanced virology - for deployment in a vetted biosecurity lab, say - and another version that cannot discuss that because it doesn't have the knowledge," they say, "you have to train two separate models. Especially in the case of frontier models, which are large and very expensive to train, the cost to the developer would be prohibitive."

So I'm going to interrupt here just to add that while OpenAI and Anthropic are not being currently forthcoming about the cost to train a current frontier model, Leo, you've always talked about how expensive it is, and oh, boy. Once those two are publicly traded, then their accounting ledgers will no longer be private. So we're going to find out. But to get some sense of scale, OpenAI's Sam Altman has stated that the training cost for GPT-4 was more than $100 million; and OpenAI reportedly spent $3 billion overall on compute to train their models two years ago, back in 2024. Also, as we know, models have grown much larger recently. And those numbers ring true, those earlier ones, because Google's Gemini model is believed to have cost Google $192 million to train. So we're talking...

Leo: That's chicken feed, though, compared to what they're spending now. I mean...

Steve: Well, actually, yes, right, because these are old and smaller models; right? So it could be half a billion dollars.

Leo: So if Fable is, as some have speculated, a 10 trillion parameter model...

Steve: Gosh.

Leo: And ChatGPT4 was, I don't know, several hundred million probably.

Steve: Right. Yeah. So all of this leads us to understanding why it's not feasible for any commercial provider, anyone, commercial or not, to train, to have multiple versions of a single model type, like a Mythos that knows nothing about cybersecurity. The advantage would be you can't trick it into revealing what it doesn't know. The knowledge would have never been put in there. So it's just not there to ask. But you had spent so much money training that up, and this is the problem that they're trying to identify.

So the quite valid point that Anthropic is making here is that it's massively infeasible to train a state-of-the-art model on any pre-filtered knowledge to make it dumb about some things, to make it just, I don't know anything about that. You know, if you're going to be spending that kind of money on training, it needs to know everything so that it can have the widest application range for its use, you know, and then we're going to have some chance of getting some money back out of all that training that went into it. So if a state-of-the-art model were to be trained on a filtered subset of everything, that is, a limited one, forever, you end up with a very expensive, forever limited model.

Okay. So we've seen that imposing post-training behavior, which is what we were just talking about, this refusal abliteration, post-training behavioral restrictions on open source models where you're able to modify the network weights, that can be altered, so it was a short-lived solution. The means for removing those restrictions, as we saw, it's all public knowledge, child's play. And even the best closed-weight models, which are operated by cloud-based hyperscalers, like OpenAI and Anthropic and Google and so

forth, AWS, we've seen that they can be prone to trickery and abuse, you know, prompt injection example, and next week we're going to understand exactly how that happens. So what's clearly needed is a new solution, and that's what these researchers have found.

Leo, we're at two hours. Let's take our last break, and then we're going to look at the solution for this dual-use problem and how to solve this with a single training.

Leo: We will have more of this. I am just, by the way, absolutely fascinated by this. I feel like this should be required listening for the listeners of Intelligent Machines, our AI show tomorrow because it's such foundational information about how all this stuff works.

Steve: Yeah.

Leo: And it explains a lot, to be honest, about how these models work. And I think it's a good idea to kind of understand the underlying technology because it means that you can do a better job.

Steve: And again, as a user, you know, Lorrie, she's using the...

Leo: Doesn't matter.

Steve: ...heck out of ChatGPT. But our audience, that's why they're here.

Leo: Sure. Sure.

Steve: Is to get this.

Leo: And, you know, for advanced users who are looking at things like abliterated models and wondering why some models do this and some models do that, there's so much going on. This stuff moves so quickly, it's very helpful to understand it a little bit, absolutely. I appreciate it. We're going to take a little break and come back with more. You're watching Security Now! with the wonderful Steve Gibson. On we go, sir.

Steve: So because post-training behavior modification has been demonstrated to be easily removable, it is not sufficient to say don't give people this information. It doesn't work to suppress that behavior. What we need is a new solution. And it's completely infeasible to do multiple training runs with different combinations of information filtered out of a model because training is prohibitively expensive. So what we need is a new solution. And that's what these researchers working with Anthropic have found.

Anthropic continues their writing, saying: "In new research carried out with collaborators at AE Studio, we explore a new method that could enable the benefits of training many separately filtered models, but at the cost of training only one. We call it GRAM, Gradient-Routed Auxiliary Modules," they wrote. "Note that the results of the

experiments presented here are preliminary. GRAM has not been applied to any of the production models at Anthropic." And they said: "And we're not sure it ever will be."

And, okay, so I take that to reflect a very reasonable and very cautious approach because can you imagine the cost of a mistake if one of these companies' GRAM-trained model, whatever that is, and we'll get to that in a second, turned out to have unsuspected problems? There's no reason to believe that it would, or that that would be the case. But, again, their caution is understandable because they could be scrapping half a billion dollars these days.

So they write, Anthropic writes: "How GRAM works: The idea behind GRAM is to give a model dedicated, removable compartments for each category of dual-use knowledge, and to update only those compartments when learning from dual-use data." Again, to update only those compartments, not all of the models' weights, only those in the compartments, when learning from dual-use data.

They said: "Concretely, GRAM adds extra neurons to every layer of a standard Transformer, the neural network architecture on which large language models are based. These neurons are divided into groups, or 'modules,' one module per dual-use category. During training, when the model encounters general-purpose text" - that is, you know, just standard text, we don't worry about it one way or the other - "it learns in the usual way. But when it encounters text from a dual-use category - virology, for instance - the rules change. The model can use its general knowledge to make predictions, but only the virology module is allowed to learn from that text. The general-purpose weights are temporarily frozen." Right?

So in other words, the bulk of the model isn't changed by learning about virology. Only the neurons in the virology module are changed. And if the bulk of the models' weights don't change after learning about virology, it doesn't learn about virology. It doesn't gain any knowledge from that. It's like it never happened to most of the model.

They said: "The consequence is that virology knowledge accumulates in the virology module rather than diffusing across the whole network. After training, the module can simply be deleted, and the virology knowledge goes with it. Or it can be left in place for trusted deployments, when virology..."

Leo: It gives it a lobotomy.

Steve: Exactly. No, it's exactly...

Leo: What's interesting about this is you would think, well, that's how all knowledge is. But it isn't. When the models are trained, it propagates through the whole model.

Steve: Yes.

Leo: So this is a special technique that says, no, no, only virology can, you know, can only go here. That's interesting.

Steve: Yes. And what's really interesting is that it tends to concentrate there because it's like the virology module doesn't - it's like it knows it's carrying the full weight of that knowledge.

Leo: Yeah. Yeah.

Steve: It's so cool.

Leo: So amazing.

Steve: So they said: "The knowledge can be tailored very specifically to the type of deployment needed. In our experiments, we defined four dual-use categories, so that one training run with GRAM yielded a model that can be configured in 16 different ways, 'on' or 'off' for each of the four categories." And as we know, a four-bit number can have 0 through 15, so 16 different possible ons and offs.

They said: "We tested GRAM in three settings of increasing realism. First, on a synthetic dataset of children's stories tagged by topic, a small GRAM model could be reconfigured to 'forget' any chosen topic, and each configuration performed almost identically to a separate model trained from scratch with that topic filtered out." In other words, they did an AB comparison. Here's a GRAM-trained model where we turned off the topic, comparing it to a normal model that was never trained with that topic, and there's no difference in behavior. And they made it more clear. They said: "That is, for the cost of training a single model, we achieved results that would normally require multiple training runs on different datasets.

"Second, we trained a larger model on a realistic mix of web text, code, and scientific papers, with four dual-use domains: virology, cybersecurity, nuclear physics, and a niche programming language (to serve as a proxy for specialized dual-use code)." They said: "The capability associated with each dual-use domain is routed to its own module. Deleting a module removed the corresponding capability about as effectively as never having trained on that data at all. Remarkably," they said, "we find that this removal did not degrade general performance."

And finally they said: "We also tested whether an attacker could recover the removed knowledge by training on a small amount of malicious data. Believe it or not, GRAM resisted this about as well as data filtering did. By contrast, an 'unlearning' technique applied after training only suppresses the knowledge. That's what we've been talking about. It was easy to remove that with a small amount of fine-tuning." So that's a parenthetical about the resistance ablation.

And then, finally, third, they said: "We ran the experiment at seven model sizes from 50 million to five billion parameters. GRAM matched the performance of data filtering at every size, and the gap between 'module on' and 'module off' grew wider as models got larger."

I'm going to pause on that for a moment. The larger the model the more general knowledge was stored outside of the various subject matter-specific modules, so the subsequent removal or suppression of any one or more of them during inference as a result had diminishing effects upon the model's overall performance. This is exactly what we would hope to see.

Leo: I wonder, you know, I'm running two different kinds of models right now. In the large video card, the 3090, I can run there what's called a "dense model," which is that's the Qwen 3.8 27B. And it's, I think, kind of extrapolating from what you just said, it's a model where all the weights are spread throughout the model. That's why

they call it "dense." But in order to run DeepSeek V4 Flash on my Sparks, on two different machines with a fast interconnect, it has to be - it can't use a dense model. You can't split the lobes of the brain.

Steve: Right.

Leo: It has to be what they call an MOE, or Mixture Of Experts.

Steve: Experts; right.

Leo: And the advantage of doing that is you don't load the whole model into memory.

Steve: Right.

Leo: You take shards of it. I suspect this is a similar technique, that you kind of localize knowledge in a shard as opposed to spreading it throughout an entire dense model.

Steve: Right, right. And, I mean, in retrospect it seems obvious; right? If you don't update a model's weights, it can't learn what you just showed it.

Leo: Right.

Steve: Because you didn't change, you know...

Leo: That's how it learns. That's what learning is.

Steve: Yes, yes. You made it forget. You made it, you know, like nothing happened.

Leo: Yeah.

Steve: And so then if you reserve a region that you do allow to learn, then as it turns out it learns about virology. The rest of the model can't because you froze its weights.

Leo: It's all stuck here, yeah.

Steve: Doesn't, I mean, but it turns out this actually works. And the larger the model gets, the better it works.

Leo: Interesting.

Steve: Which is what, of course, which is what they care about because now, you know, they're not interested in doing any five billion parameter models anymore. Sorry, that was just an experiment to see, you know...

Leo: This 3.8 is a relatively small model at 27 billion parameters; right?

Steve: Exactly. Exactly.

Leo: And DeepSeek V4 Flash is considerably larger than that, yeah.

Steve: So they continue, saying: "As AI companies train more capable models, the need to limit access to dual-use capabilities will increase." Okay. So in other words, Anthropic understands that the more capable the industry's models become, and we're seeing it, like, before our eyes, the more valuable the knowledge they contain will become, and thus the need to manage the access to that knowledge grows increasingly crucial. So they also make another very good point. They write: "Today, companies limit access through classifiers and refusal training." And we just blew up refusal training; right? That's gone. That no longer works. Well, if you have access to the weights. If it's OpenAI and Anthropic, they're not giving you access to their closed weight models. So you can't abliterate those. But you sure can the open weight ones.

They said: "However, these safeguards" - meaning classifiers and refusal training - "are difficult to make robust without degrading performance on harmless requests. Methods like GRAM offer a potential path toward access control that is more robust." And that's a really great point. The current system which combines the refusal training, which we examined earlier, with real-time input and output classifiers, that provides at best fuzzy filtering. The AI can frustrate its innocent user by refusing a benign prompt. It's like, wait. What do you mean you won't answer that? You're an AI. I know you're an AI. But why won't you give me the answer? And, similarly, it can delight a malicious user by letting down its guard when it should not.

But by using a method like GRAM, a model can have its functioning knowledgebase selectively tuned to the authentication level or nature that the prompter's access permissions specify. So that's a significant improvement over today's soft and somewhat ad hoc solutions.

So Anthropic concludes, writing: "This is early research, and there are clear limitations. We have not tested GRAM at frontier scale or in a production training pipeline." And they said: "As noted above, it has not been applied to any of our Claude models. Our evaluations quantify performance in terms of next-token prediction ability, rather than performance on real downstream tasks. And there's a deeper open problem that applies to data filtering and methods like GRAM: some dual-use capabilities might be so entangled with general knowledge that no method can separate them cleanly." And they finish, saying: "For further details about our experiments, read the post on our Alignment Science blog."

And that's really interesting, Leo. The point they make I think is a good one. Like you need to know when to freeze learning globally, and only allow the knowledge to be concentrated in the module. But that decision is going to be a little soft and fuzzy, too; right? Like some biology is not virology, or not prone to abuse. But, you know, again, it's not binary; right? It's going to be kind of on a continuum somewhere.

So anyway, when I wrote about GRAM two weeks ago in the Security Now! weekend special email, the one before Black Hat, several of our listeners wrote back with feedback that I also shared during our Black Hat podcast. Their concerns surrounded censorship and "who gets to decide who has access to what data." When I voiced that during Black Hat, both Richard and Paul chimed in immediately, well, this was just a different version of the way things have always been. You know, the fact that AI has made most of the world's knowledge vastly more accessible doesn't necessarily mean or need to mean that AI has made ALL of the world's knowledge vastly more accessible to everyone.

You know, there isn't any entitlement to the knowledge that AI holds. After all, it cost those companies hundreds of millions of dollars to create and offer this facility. So I would argue that they can put whatever restrictions on it they wish. If commercial providers of that knowledge are required by internal policy, public pressure, the government, their stockholders, or whomever, to gate and control access to some aspects of that knowledge, I think that's entirely reasonable. And I would argue that the commercial providers have every right to do so.

We just saw an example with OpenAI and that Hugging Face incident, where Hugging Face was unable to deploy either Anthropic's or OpenAI's models to help with their cybersecurity forensic investigation after they'd been attacked by OpenAI because Hugging Face hadn't been granted the magic keys to those providers' AI cybersecurity knowledge. Everyone would argue now that they should have had such access, and that we did later hear that OpenAI was working with them to give it to them.

So anyway, I keep repeating the qualifier "commercial providers" because, as we know, and you're using them, Leo, there are alternatives, and an alternative is exactly what Hugging Face turned to after their commercial models refused to help them. Yu know, they used one of the many publicly available completely unrestricted open-source models.

Leo: GLM 5.2, which is really good, and has been succeeded by, interestingly, GLM 5.3, which is the same - this is an interesting slice on what you were just talking about. It's the same...

Steve: From Z.ai; right?

Leo: Z.ai says it's the same model. It's the 5.2 model with more enhanced post-training.

Steve: Uh-huh.

Leo: So there is, there's headroom even there.

Steve: Yes.

Leo: Where you could take the blob that is all the weights and...

Steve: And just do a better job with the knowledge.

Leo: Fine-tune it, yeah.

Steve: Yes. Because the post-training is behavior. Pre-training is knowledge; post-training is behavior.

Leo: And interestingly, that's where it really excels is at coding and that kind of thing, yeah.

Steve: Right.

Leo: It's really fascinating what's going on here.

Steve: So just to finish here, anyone who might object to the big commercial services restricting what can be done can easily turn to any of the many alternatives. And I have to say, given what I was able to do with the carefully unrestricted and uncensored Venice.AI service, which I played with briefly when it appeared, and we talked about it here on the podcast, I'm pretty certain that fully unrestricted, unrestrained, and uncensored AI models are readily available, even from third parties in the cloud, you know, not from OpenAI and Anthropic. That's not their style.

Leo: Well, they're serving their proprietary models. But because there are so many open weight models, you know, Hugging Face has more than three million models. And those aren't all - those are, you know, many, many millions of them are just abliterated or somehow modified.

Steve: Yes.

Leo: Larger, well-known open models.

Steve: Right.

Leo: So there might be thousands or hundreds of thousands of GLMs on Hugging Face that hackers modified. That's a very fun thing for people to do.

Steve: Yeah, and it's, you know, it's like apps to download for Android. A bunch, you know, how many millions of them are...

Leo: A lot of them are crap, absolutely.

Steve: So we have run out of time, and there was a lot to take in.

Leo: No. I want to know more, Steve.

Steve: We have now a far better understanding of the way today's AI works, I mean, at least enough to kind of have a feel for it. It seems like it's less magic than it was. We're going to learn next week how and why prompt injection attacks continue, despite the best minds in the industry struggling to prevent them. That technology blew my mind on the plane flight to Las Vegas. And I'm going to blow everybody's mind next week.

Leo: The chain of thought is not all it's made out to be.

Steve: As they say, stay tuned.

Leo: It really is interesting stuff. Thank you, Steve. Steve Gibson, our guru, now not of just security, but of AI, as well.

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/