Security Now! #911 - 02-21-23

A Clever Regurgitator

This week on Security Now!

For how long were bad guys inside GoDaddy's networks? What important oral arguments is the US Supreme Court hearing today and tomorrow? What's Elon done now? What's Bitwarden's welcome news? What's Meta going to begin charging for? Should we abandon all hope for unattended IoT devices? Are all of our repositories infested with malware? How'd last Tuesday's monthly patchfest turn out? Why would anyone sandbox an image? What can you learn from TikTok that upsets Hyundai and KIA? And are there any limits to what ChatGPT can do, if any? We're going to find out by the end of today's 911 emergency podcast.

Security News

GoneDaddy

Last Friday GoDaddy revealed a rather astonishing bit of news: Its network and organization had suffered a multi-year security compromise that allowed attackers who remain unidentified to exfiltrate the company's source code, customer and employee login credentials, and install malware that redirected customer websites to malicious sites.

With nearly 21 million customers and last year revenue of nearly $4 billion, GoDaddy is one of the world's largest domain registrars. Many years ago, when I was making my move away from Network Solutions, I considered GoDaddy. It's the choice of a very techie friend of mine, Mark Thompson. But for me it was too bubblegum and commercial. What I want from my domain registrar is staid, stodgy and stoic. I don't want a domain registrar that looks like Romper Room. I don't need entertainment and up-selling. So I chose Hover and I've been very happy – and just to be clear, my choice was made years before Hover became a TWiT sponsor.

So, in a filing Thursday with the S.E.C. (the U.S. Securities and Exchange Commission), GoDaddy admitted that three serious security events, the first three years ago 2020 and somehow lasting through 2022 were all carried out by the same intruder. They wrote:

"Based on our investigation, we believe these incidents are part of a multi-year campaign by a sophisticated threat actor group that, among other things, installed malware on our systems and obtained pieces of code related to some services within GoDaddy." and GoDaddy said that their investigation was ongoing.

The most recent event occurred last December when the threat actor gained access to the hosting servers GoDaddy's customers use to manage websites hosted by GoDaddy. The threat actor installed malware on the servers that *"intermittently redirected random customer websites to malicious sites."* GoDaddy was unaware of the presence of this malware and learned of it from their customers who were complaining that visitors to their sites were occasionally being redirected elsewhere.

GoDaddy said: *"We have evidence, and law enforcement has confirmed, that this incident was carried out by a sophisticated and organized group targeting hosting services like GoDaddy. According to information we have received, their apparent goal is to infect websites and servers with malware for phishing campaigns, malware distribution, and other malicious activities."*

Saying *"hosting services **like** GoDaddy"* begs the question whether other hosting services have been similarly affected and, if so, which ones and by whom? Those questions remain unanswered.

It appears that the first of the several intrusions took place in March 2020, when a threat actor obtained login credentials that gave it access to employee accounts and the hosting accounts of roughly 28,000 customers. Fortunately, those hosting login credentials did not also provide access to the customers' main GoDaddy account. That first breach was disclosed two months later in May 2020 in a notification letter sent to affected customers. The company said on Thursday it's responding to subpoenas related to the incident that the Federal Trade Commission issued in July 2020 and October 2021.

GoDaddy then discovered another incident in November 2021 two months after the threat actor obtained a password that gave access to source code for GoDaddy's Managed WordPress service. So, beginning two months earlier in September of 2021, the unauthorized party used their access to obtain login credentials for WordPress admin accounts, FTP accounts, and email addresses for 1.2 million current and inactive Managed WordPress customers.

And these were not the first of GoDaddy's many problems. Through the years, security lapses and vulnerabilities have led to a series of suspicious events involving large numbers of sites hosted by GoDaddy. For example, in 2019 a misconfigured domain name server at GoDaddy allowed hackers to hijack dozens of websites owned by Expedia, Yelp, Mozilla, and others and use them to publish a ransom note threatening to blow up buildings and schools. The DNS vulnerability which was exploited by the hackers had come to light three years earlier yet GoDaddy didn't take any action to mitigate the risk.

Also in 2019, a researcher uncovered a campaign that used hundreds of compromised GoDaddy customer accounts to create 15,000 websites that published spam promoting weight-loss products and other goods promising miraculous results.

One question I had was how it was that GoDaddy could assert, through the more recent three attacks spanning the same number of years, that they had been repeatedly plagued by a **single** threat actor, yet they somehow have no idea who this individual or group is? So I did a bit more digging and I found that in their 10-K filing with the S.E.C. they stated that the most recent, December 2022 incident, is connected to the two other security events they suffered in March 2020 and November 2021.

This reminded me of what we recently saw from LastPass, where we were told that the **second** attack – the one where all of our backed-up LastPass vaults were stolen – was enabled by the **initial** intrusion. That was worrisome since it suggested that LastPass had not fully cleaned up after the first intrusion. In the GoDaddy case, they appear to be stating that they know that it's the same threat actor because information obtained during the initial intrusion, three years ago back in 2020, was subsequently used in both 2021 and 2022. This suggests, as with LastPass, that post-intrusion cleanup may have been minimized, but in any event, was ineffective. A full post-intrusion clean-up means that nothing that an intruder could possibly have obtained remains valuable once the clean-up is concluded. We know that didn't happen in the case of LastPass and that also appears to have been the case for GoDaddy.

As we've had occasion to note on this podcast, once malware has had access to a system, you can never fully trust it again. And I should really remove the qualifier "fully" — you cannot trust any system after it's been compromised. These days, when we have malware burrowing into our motherboard firmware to maintain persistence, the only course of action is to reflash firmware, wipe drives and rebuild from scratch – and change everyone's access credentials!

Yes, this is a huge nightmare in the case of a large sprawling enterprise, but there's really no choice. After GoDaddy's initial 2020 breach, either something lingered in a system that was never found, or they failed to rotate all of the keys and logon credentials across the enterprise. **Something remained —** either malware tucked away in an unexamined corner, or someone's credentials that were never changed.

Section 230
Today and tomorrow, the US Supreme Court will be hearing initial oral arguments in a pair of cases which will open the door to allow the Court to reexamine the famous – and infamous – Section 230 of the Communications Decency Act which was passed into law by Congress 27 years ago, back in 1996. A crucial 26 words from Sector 230 of that law are what enable our Internet's media companies to remain un-responsible – and some would say irresponsible – for the content that their users post online for consumption by others.

Those 26 words are: *"No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider."* In other words, this blanket protection provides that none of today's media companies can be held responsible for the content that is being served by their technologies. It serves as powerful and now crucial protection. But many wonder whether it might have been taken too far.

The specific question that the cases address, focuses upon the content promotion algorithms used by Google for YouTube, Facebook, Twitter and others, to provide their users with *"more relevant"* content. So the question may be whether our social media companies **have** actually become publishers of this content the moment they involve themselves in that content's deliberate selection and promotion – even if that involvement is entirely algorithmic. The argument, then, is that they are no longer acting as passive repositories of user-provided content and that the selections made by their algorithms are ultimately motivated by profit.

Jeff Kosseff, a cybersecurity law professor at the U.S. Naval Academy wrote an entire book on Section 230, titled *"The Twenty-Six Words That Created the Internet."* In reporting by the Washington Post early last October, when the Supreme Court decided that they would hear the two cases which are now before them, they quoted Professor Kosseff saying: *"The entire scope of Section 230 could be at stake, depending on what the Supreme Court wants to do."*

Although the stakes could not be much higher, the way these things go we won't have a decision anytime soon. Likely, not until much later in the year, at the earliest. But this will be one to watch.

For their part, the plaintiff's attorneys say that applying the sweeping civil immunities created by Section 230 to algorithmic recommendations incentivizes the promotion of harmful content, and that Section 230 denies the victims of such content any opportunity to seek redress when they can show those recommendations caused injuries or even death. So this will be an interesting pair of cases to keep an eye on.

No Blue, No SMS-based 2FA
The Verge's headline was: [It's] *"Official: Twitter will now charge for SMS two-factor authentication. Only Twitter Blue subscribers will get the privilege of using the least secure form of two-factor authentication."*

They continued:

> *Now, it's official: You have to pay for the privilege of using Twitter's worst form of*

authentication. In fact, if you don't start paying for Twitter Blue ($8 a month on Android; $11 a month on iOS) or switch your account to use a far more reliable authenticator app or physical security key, Twitter will simply turn off your 2FA after March 20th.

[The writer adds] I know which one I would choose.

Good riddance to SMS is my feeling, given how common SIM swap hacks are these days. Heck, Twitter's own Jack Dorsey was successfully targeted by the technique four years ago. You don't want someone to get access to your accounts by proving they are you simply because they've stolen your phone number.

That's how Twitter is trying to justify this change, too, but I wouldn't be surprised if there's a simpler reason: it costs money to send SMS messages, and Twitter does not have a lot of money right now. The company had been phasing out SMS even before Elon Musk took over.

Twitter's own transparency data shows that as of December 2021, only 2.6 percent of Twitter users had 2FA turned on, and 74 percent of those users were using SMS as their 2FA method.

Here's what Twitter posted and explained last Wednesday...

Titled: ***An update on two-factor authentication using SMS on Twitter By Twitter Inc.***

We continue to be committed to keeping people safe and secure on Twitter, and a primary security tool we offer to keep your account secure is two-factor authentication (2FA). Instead of only entering a password to log in, 2FA requires you to also enter a code or use a security key. This additional step helps make sure that you, and only you, can access your account. To date, we have offered three methods of 2FA: text message, authentication app, and security key.

While historically a popular form of 2FA, unfortunately we have seen phone-number based 2FA be used - and abused - by bad actors. So starting today, we will no longer allow accounts to enroll in the text message/SMS method of 2FA unless they are Twitter Blue subscribers. The availability of text message 2FA for Twitter Blue may vary by country and carrier.

Non-Twitter Blue subscribers that are already enrolled will have 30 days to disable this method and enroll in another. After 20 March 2023, we will no longer permit non-Twitter Blue subscribers to use text messages as a 2FA method. At that time, accounts with text message 2FA still enabled will have it disabled. Disabling text message 2FA does not automatically disassociate your phone number from your Twitter account. If you would like to do so, instructions to update your account phone number are available on our Help Center.

We encourage non-Twitter Blue subscribers to consider using an authentication app or security key method instead. These methods require you to have physical possession of the authentication method and are a great way to ensure your account is secure.

Some other reporting I found stated that Twitter took this step because SMS 2FA was being

abused by fraudsters who would establish accounts using application to person – or A2P – premium telephone numbers. Then, when Twitter would send 2FA texts to these numbers the fraudsters would get paid. Estimated losses are claimed to be around $60 million a year.

Everyone is piling on Elon these days. And his decisions at Twitter have been a source of controversy. 74% of 2.6% is 1.95%. So at the end of 2021, 1.95% of all Twitter account holders were using SMS-based 2FA. On the other hand, 3 out of every 4 of the Twitter users who use **any** form of 2FA were using SMS and the use of **any** form of 2FA certainly prevents some amount of abuse. And even though SMS is not the best solution, it's still better than having none, and using it doesn't create any new vulnerability where none existed before. It's just not something that can be relied upon as much as one-time passcodes or security keys.

I don't think this is great news because it seems to me that it might end up causing Twitter users to simply disable all use of 2FA without upgrading their existing SMS authentication to one-time passcodes or a security key.

At around 450 million monthly users of Twitter, that 1.95% who have been using SMS-based 2FA is 8.25 million SMS users. So that likely adds up and I can see Elon wanting to cut that cost. And if there's no way for Twitter to determine whether the phone numbers being registered are "pay to send" numbers then I suppose he doesn't have much choice. But a great many other large social media organizations offer SMS-based 2FA and they don't appear to have similar problems.

In any event, I hope that those who need some form of authentication will move to passcodes rather than just turning off all extra authentication steps.

Bitwarden gets Argon

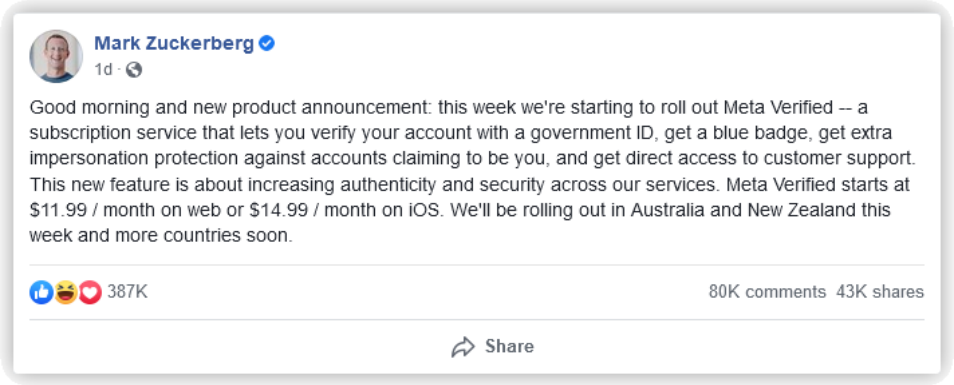
The Argon2 memory hard PDKDF which promises to be far more resistant to brute forcing attacks is now available from Bitwarden and is present on some Bitwarden clients. But before switching to it, since the switch must be made system wide per user, you'll need to wait until all of the platform clients you use have been upgraded to support Argon2.

Six days ago, "Ryan_BW" a Bitwarden Employee posted to Reddit:

> *For those curious as to why not everything is rolled out at once, each browser extension and mobile app needs to go through an approval process with their respective app stores. Please be patient - usually the approval process takes about a week.*

The version to wait for is: 2023.2.0.

"Meta Verified"



Facebook adds paid identity verification and more. They wrote:

> *Some of the top requests we get from creators are for broader access to verification and account support, in addition to more features to increase visibility and reach. Since last year, we've been thinking about how to unlock access to these features through a paid offering.*
>
> *With Meta Verified, you'll get:*
>
> - *A verified badge, confirming you're the real you and that your account has been authenticated with a government ID.*
> - *More protection from impersonation with proactive account monitoring for impersonators who might target people with growing online audiences.*
> - *Help when you need it with access to a real person for common account issues.*
> - *Increased visibility and reach with prominence in some areas of the platform– like search, comments and recommendations.*
> - *Exclusive features to express yourself in unique ways.*

First of all, $12/month on the web and $15/month on iOS strikes me as expensive. It's not a one-time verification fee, which would seem reasonable. This is an ongoing cost of $144 or $180 per year. I suppose that it's not for everyone. If someone uses Facebook as a major platform then I could see how it makes sense to pay something to obtain spoofing prevention and higher visibility in search results.

Emsisoft Fake Code Signing

In a reminder why simply having code signed is not, and should not be, sufficient to have anti-virus and download protection warning silenced, The anti-virus publisher, Emsisoft has put out a public service announcement warning that threat actors are currently using fake Emsisoft code-signing certificates to sign their malware. This results in attacks appearing to come from Emsisoft's products as well as to slip past anything that refuses to run unsigned software. At some point codesigning will become necessary but not sufficient. At the moment it's entirely optional and mostly for user assurance.

Attacks breaking records
DDoS attacks are always resource depletion or resource consumption of one kind or another. Today's modern DDoS attacks are no longer floods of TCP SYN packets. Those now seem quaint. No. Modern attacks are aimed less at consuming or clogging raw bandwidth than at asking web servers to generate more pages per second than they possibly can. Since modern sites are generally the front-facing surface of a complex content management system which is driven by some form of SQL database back end, individual HTTPS queries have become much more computational than yesterday's static web pages.

The previous modern-style DDoS attack-blocking record was set by Google Cloud which, last June, reported blocking an attack rate of 46 million HTTPS requests per second. But that was then. Now, Cloudflare has reported that it successfully fended off an attack that was 35% greater than that, mitigating a record-breaking HTTPS DDoS attack of 71 million requests per second. That's a lot of bots spread around the world all concentrating their fire onto a single target.

There are a growing number of strong website DDoS defenders. Today they include:

- Akamai DDoS Mitigation,
- AWS Shield,
- Cloudflare DDoS Protection,
- Google Cloud,
- F5 DDos Hybrid Defender,
- Imperva DDoS Protection and
- Microsoft’s Azure DDoS Protection.

Websites that pay to be located behind them are able to remain online even during an attack of such scale. That alone is somewhat astonishing. And an attack of this scale would utterly obliterate any other site that's simply “on the Internet”.

The mitigation of attacks of such scale, while avoiding collateral damage to nearby resources, requires carriers of the attacking traffic bound for a site under an attack to block all traffic as far away upstream from the target as possible to prevent that traffic’s aggregation as it moves from router to router approaching its destination. If we picture the Internet as a highly interconnected global network of individual routers, each one forwarding traffic toward its destination, a useful overlay for this is the image of a funnel, where incoming traffic is being funneled toward its target. In the model of a funnel, the closer we approach the funnel’s neck, the greater the traffic burden becomes. Since the physical implementation of this traffic movement are individual routers, the best defense against “too much traffic” is to cause attacking traffic packets to be dropped far out at the funnel’s mouth.

But doing this effectively inherently requires a large traffic provider. If the provider’s network was not sufficiently large to allow the incoming traffic to be blocked before it has the opportunity to concentrate, then the provider’s aggregation routers would be swamped and many other of that provider’s customers would be impacted by the collateral damage caused by a failure of the packet transport fabric.

An organization of Cloudflare's size, to name just one, has the advantage of operating at global scale. And when we're talking about handling attacks of this size, the network's size is not only an advantage, it's a necessity. Since attacking bots are also globally spread, traffic bound for one customer's website will be entering the network of a global carrier such as Cloudflare at many peering points across the globe. So the moment an attack is detected, all of the provider's edge routing infrastructure can be informed and switched into an attack mitigation stance.

We talked many years ago about the sheer brilliance of the Internet's design with the original concept of autonomous packet routing. That the original concept has withstood the tests of time, insane growth in usage and application, stands as a testament to those who created this system. But, its great weakness is that it was never designed to withstand deliberate abuse. The idea that someone would flood the network with attack traffic was something that this system's gifted designers could never have anticipated. Even so, the Internet's basic architecture has been adaptable to incorporate such protections over time.

More Mirai

Speaking of DDoS attacks, I've often worried out loud here, for at least the past several years, about what would happen when malicious actors finally got around to focusing their evil intent upon, and commandeering for their nefarious ends, the truly countless number of Internet-connected low end IoT devices. Well, those worries are beginning to manifest.

Last year, from July through December 2022, Palo Alto Networks Unit 42 researchers observed a Mirai Botnet variant known as V3G4 predominantly leveraging IoT vulnerabilities to spread. V3G4 targets 13 separate vulnerabilities in Linux-based servers and IoT devices. The devices are commandeered for use in DDoS attacks. The malware spreads both by brute-forcing weak or default telnet/SSH credentials, and by exploiting known, but unpatched, firmware coding flaws to perform remote code execution on the targeted devices. Once a device is breached, the malware infects the device and recruits it into its botnet tribe.

This is exactly what I've been warning of, for years. Though it makes no rational sense at all, we know how difficult it is even to update big iron systems that need to be kept current, where there's a well established notification and patching infrastructure in place to support that. Just look at the recent VMware ESXi fiasco. Those systems should have been readily updated.

Compare that to some random IP camera long ago installed and since forgotten. What about patching it? Good luck with that. We can't even keep our servers patched. Today, as I've often lamented, we have a literally uncountable number of gizmos and gadgets attached to the Internet. Why? Because we can. And while many of those in our homes are safely tucked away behind the one-way valve of our NAT routers (and are also, hopefully, on their own isolated network where possible), a great many, due to their role and application, have deliberately been given access to the public Internet.

In the present case of V3G4, Unit 42 tracked three distinct campaigns. Unit 42 believes all three attack waves originated from the same malicious actor because the hardcoded command and control domains contain the same string, the shell script downloads are similar, and the botnet clients used in all attacks feature identical functions.

So what does V3G4 attack? It exploits one of the 13 vulnerabilities:

- CVE-2012-4869: FreePBX Elastix remote command execution
- Gitorious remote command execution
- CVE-2014-9727: FRITZ!Box Webcam remote command execution
- Mitel AWC remote command execution
- CVE-2017-5173: Geutebruck IP Cameras remote command execution
- CVE-2019-15107: Webmin command injection
- Spree Commerce arbitrary command execution
- FLIR Thermal Camera remote command execution
- CVE-2020-8515: DrayTek Vigor remote command execution
- CVE-2020-15415: DrayTek Vigor remote command execution
- CVE-2022-36267: Airspan AirSpot remote command execution
- CVE-2022-26134: Atlassian Confluence remote command execution
- CVE-2022-4257: C-Data Web Management System command injection

Notable is that some of these CVEs date from 2012, 14, 17, and 19. There is no reason to imagine that any of these problems will ever be repaired. Why would they? The device is working just fine. And who even knows whether the company that created it still even exists? A new trend we've observed is that companies are formed on the fly by pulling together the individual required resources, devices are designed, manufactured and sold, then the entire briefly assembled organization dissolves, returning to its original component parts. There is no one to call for updates. There is no follow-up. There is no accountability. Yet an Internet-connected gadget can now harbor hostile code and be used, probably throughout the rest of its long service life, as one more tiny cog in a massive and untraceable global attack-launching platform.

Again, in the case of V3G4, after compromising the target device, a Mirai-based payload is dropped onto the system and attempts to connect to the hardcoded command and control address. Once running, the bot terminates a large number of known processes from a hardcoded list, which includes other competing botnet malware families. There's a new king of the hill.

A characteristic that differentiates V3G4 from most other Mirai variants is that it interlaces the use of four different malware XOR encryption keys rather than just one. This was an attempt to make static analysis reverse engineering of the malware’s code and decoding its functions more challenging. As I briefly noted earlier, when spreading to other devices, the botnet uses a telnet/SSH brute-forcer that tries to connect using default or weak credentials and those 13 known vulnerabilities.

Once setup and running with a connection to the Botnet's command and control, the compromised devices are given DDoS commands directing their attacks. This variant offers TCP, UDP, SYN, and HTTP flooding methods.

The Unit 42 guys suspect that V3G4 sells DDoS services to clients who want to cause service disruption to specific websites or online services, though the front-end DDoSing service associated with this botnet had not been identified at the time of Unit 42's report.

NPM malware

Week after week I encounter news of malware stashes being found on this or that, or sometimes all, popular code repositories. An example of such a piece of news from last week is that Check Point's research team detected 16 malicious JavaScript packages uploaded on the official npm registry. The researchers said that all packages were created by the same author, and were designed to download and run a hidden cryptominer on a developer's system. All 16 of those related packages have now been removed.

So it's like that... week after week. Endlessly. I'm mentioning it this week because I don't mention that all of this is happening ... every ... single ... week ... in one form of another. Sometimes it's npm. Sometimes it's PyPI. Sometimes something else. Basically, wherever security firms look they are now finding malicious packages. So, I just wanted everyone to be aware of this constant flux of malware dribbling into the open source ecosystem. It's today's reality.

Patch Tuesday

Last Tuesday was, of course, February's patch Tuesday and many well-known software publishers got in on the action. The industry was made aware of security updates released by Apple, Adobe, Git, Microsoft, and SAP. The Android project, OpenSSL, and VMWare also released security updates last week. Microsoft patched 80 vulnerabilities, including three 0-days and Apple released security updates that included a patch for an actively exploited Safari (WebKit) 0-day vulnerability. We know that the sometimes crucial mistakes many large and small organizations make is ignoring these fixes. If everyone kept their software patched we'd be seeing many fewer widespread problems, such as that VMware ESXi debacle — which is still ongoing, by the way with more than 500 newly compromised systems just last week.

But it turns out that it wasn't all smooth sailing with this month's security updates. Microsoft has stated that some Windows Server 2022 virtual machines may no longer boot after installing the updates released last week. This issue only impacts VMs with Secure Boot enabled and running on VMware's vSphere ESXi 6.7 U2/U3 or vSphere ESXi 7.0.x. The culprit is patch KB5022842 which, if installed on guest virtual machines (VMs) running Windows Server 2022 may no longer start up. VMware and Microsoft are working to determine the cause. Interestingly, even though Microsoft says that only VMware ESXi VMs are affected, some admin reports point to other hypervisor platforms (including bare metal) also being impacted by this issue.

Samsung announces "Message Guard"

Last Friday, Samsung announced a new feature for, at the moment only its Galaxy S23 series smartphones, called Message Guard. The details are sketchy, and it sounds like it resembles Apple's "Blast Door" technology that was introduced in iOS 14. Both technologies, Message Guard and Blast Door, are image rendering sandboxes. We've often talked about the difficulty of safely and securely rendering images because image compression encodes images into a description that must later be read and interpreted in order to recover a close approximation of the original image. It's those image decompressing and rendering interpreters that have historically harbored subtle flaws that malicious parties have leveraged to create so-called zero-click exploits, meaning that all the phone needs to do is display an image in order to have it taken over by a remotely located malicious party.

So, Samsung now has this technology for its S23 series and has said that it plans to expand it to other Galaxy smartphones and tablets later this year that are running on One UI 5.1 or higher.

The addition of these technologies represent a maturation of our understanding of the problems we face. It's so easy to imagine that any problem that is found is the last one that will ever be found. And that's of course true... right up until the next problem is discovered. Experience shows that we're not running out of such problems.

The Hyundai & Kia mess

So it turns out that millions of Hyundai and KIA autos, which is to say approximately 3.8 million Hyundai vehicles and 4.5 million KIAs are vulnerable to being stolen using just a bit of technology. And that, indeed, once the method of doing so became common knowledge in some circles, Los Angeles reported an 85% increase in car thefts of those two brands and not to be outdone in the car theft category, Chicago saw a nine-fold increase — 900% — in theft of those cars.

So first, how was the news of this spread? Believe it or not, by something being called a "challenge" which has been heavily promoted on TikTok since last summer, July 2022. TikTok presented instructional videos showing how to remove the steering column cover to reveal a USB-A format connector that can be used to hotwire the car.

Hyundai's and KIA's first low-tech response, which began last November, was to work with law enforcement agencies across the United States to provide tens of thousands of steering wheel locks. A big red steering wheel locking bar has the advantage of letting TikTok-watching car thieves know that even if they're able to enter and start the car... aiming it will still present a problem.

The fundamental problem surrounds a coding logic flaw that allows the "turn-key-to-start" system to bypass the engine immobilizer which is supposed to verify the authenticity of the code in the key's transponder to the car's ECU. In other words, no key is needed. This allows car thieves to activate the ignition cylinder using any USB cable to start and then drive off with the vehicle.

Hyundai wrote: *"In response to increasing thefts targeting its vehicles without push-button ignitions and immobilizing anti-theft devices in the United States, Hyundai is introducing a free anti-theft software upgrade* [oh how nice of them] *to prevent the vehicles from starting during a method of theft popularized on TikTok and other social media channels."*

So, the software upgrade will be provided at no charge [you better believe it!] for all impacted vehicles, with a rollout which began last Monday, initially to more than 1 million 2017-2020 Elantra, 2015-2019 Sonata, and 2020-2021 Venue cars. All the rest of the affected autos – and there are too many to list here – will be upgraded through the summer of this year. The upgrade will be installed by Hyundai's official dealers and service network in the U.S., and is expected to take less than an hour. Eligible car owners will be individually notified.
Hyundai's announcement explained that the upgrade modifies the "turn-key-to-start" logic to kill

the ignition when the car owner locks the doors using the genuine key fob. After the upgrade, the ignition will only activate after the key fob is first used to unlock the vehicle. That was the missing interlock which facilitated this hack.

The question remains, though, without a big red steering wheel locking bar, how would thieves without wheels know that your particular Hyundai or KIA is no longer vulnerable? Hyundai is solving this dilemma by supplying its customers with a convenient window stickers [I would LOVE to see what that sticker says!] which makes it clear to aspiring thieves that the car's software has been upgraded to neutralize the TikTok USB hack... so don’t bother.

Unfortunately, there are some models which completely lack the engine immobilizer technology and so cannot receive the software fix which updates the missing immobilizer logic. To address that problem, Hyundai will cover the cost of steering wheel locks for their owners. That's the definition of a kludge.

So far, all of this talk has been about Hyundai. But as noted, KIA has a similar problem. KIA has promised to start the rollout of its software upgrade soon, but hasn't yet announced any specific dates or details.

The US Department of Transportation was the source of those stats about the number of affected vehicles, and also noted that these hacks have resulted in at least 14 confirmed car crashes and eight fatalities. What do you want to bet that product liability and personal injury law firms are already rubbing their hands together over this significant screw-up.

A Clever Regurgitator

The astonishing success and the equally surprising performance of OpenAI's ChatGPT-3 Large Language Model AI, means that a new phenomenon will soon be entering mainstream use. Right here on this podcast, thanks to Rob Woodruff's inspiration to enlist ChatGPT in assisting him with authoring that LastPass vault-deobfuscating PowerShell script, we've all witnessed first hand just how significant these coming changes may be. And anyone who has been following the news of this may have continued to be somewhat astounded by what this technology appears to be capable of accomplishing.

I think that the most accurate and succinct way of describing what we're witnessing is that it is astonishing to see the degree to which a neural network using large language modeling, as exemplified by ChatGPT, is able to **simulate intelligence**. And I think that is the key concept to hold on to. ChatGPT is not, itself, in **any** way intelligent; it is a clever regurgitator of intelligence.

One of the dangers, which we can feel present, is that this turns out to be a surprisingly subtle yet crucial distinction which is guaranteed to confuse many, if not most people who casually interact with this mindless bot.

After absorbing the historical global output of a truly intelligent species – namely, man – we have an automaton that's able to take our entire historical production, all at once as a whole and quickly select from that massive corpus the right thing to say. It's able to choose it because that right thing has been expressed before, by man, in thousands of different contexts. So it appears intelligent because it's mimicking an intelligent species. A parrot in a cage who says "Polly wants a cracker" is more intelligent because it really does want a cracker. Although ChatGPT may be induced to express a desire, that's still nothing more than mimicry since it has previously absorbed all of humanity's past expressions of desire. It doesn't ever actually want anything because there's not actually any "it" there at all to do any wanting.

Again I come back to "yes, what it does is astonishing." But that's only because it is the first thing we've ever encountered that's able to convincingly sound like us. But that's all it's doing. It's sounding like us. The parrot in its cage is extremely limited in its ability to sound like us. A sufficiently large language model neural network is potentially **unlimited** in its ability to sound like us. And if we can be certain of anything, it's that this simulation will be improving over time; especially now that this technology has left the lab and that capitalistic forces of commerce will be driving and funding further advancement. But nevertheless, in no way should "sounding like us" ever be confused with "being like us." A high fidelity recording of Pavarotti may sound exactly like Pavarotti, but it isn't Pavarotti. It's just a recording.

Okay, so what got me started on this? It was an interesting experiment by some researchers at the company "**any.run**" who wanted to explore an aspect of ChatGPT's limitations. They wanted to see whether ChatGPT's otherwise impressive capabilities might extend to analyzing real-world malware. If so, it might make security researcher's lives more productive by allowing them to dump a load of code into ChatGPT to figure it out..

Their blog posting begins:

If ChatGPT is an excellent assistant in building malware, can it help analyze it too? The team of ANY.RUN malware sandbox decided to put this to the test and see if AI can help us perform malware analysis. Lately, there's been a great deal of discussion about malicious actors using ChatGPT — the latest conversational AI — to create malware.

*Malware analysts, researchers, and IT specialists agree that **writing code** is one of GPT's strongest sides, and it's especially good at mutating it. By leveraging this capability even want-to-be-hackers can build polymorphic malware simply by feeding text prompts to the bot, and it will spit back working malicious code.*

OpenAI released ChatGPT in November 2022, and at the time of writing this article, the chatbot already has over 600 million monthly visits. It's scary to think how many people are now armed with the tools to develop advanced malware.

So, going into this, our hopes were high, but unfortunately, the results weren't that great. We fed the chatbot malicious scripts of varying complexity and asked it to explain the purpose behind the code. We used simple prompts such as "explain what this code does" or "analyze this code".

Okay, so the short version of what they discovered is that ChatGPT did remarkably well when the researchers gave it toy code to examine. And it really did surprisingly well. But as the complexity of the testing code increased there was a sort of "complexity cliff" they ended up going over, after which ChatGPT collapsed completely. And knowing what we know now, isn't that exactly what we would expect?

As a Large Language Model neural network, ChatGPT is not in any way even the tiniest bit sentient. Our limited-language Parrot is more sentient. So ChatGPT is unable to "understand" anything at all. That means it's not going to be great at the true problem solving that reverse-engineering complex malware code, or any code, requires. But reverse-engineering code is very different from writing code. Thanks to the explosion of open source software, ChatGPT has previously ingested all of the source code on the Internet. That's a massive amount of real working code. And as we understand, it **is** able to select, regurgitate and rearrange the code that it has previously encountered. But when it's asked to produce code that it hasn't previously seen, that's where things start to become fuzzy and where it starts making mistakes, since, again, it's not really understanding anything about what it's doing – it's simply searching for a matching context amid all of the world's previously written code.

Last week I was corresponding with two of the sharpest minds I've ever had the privilege of knowing. And I was talking about the idea I've previously shared here, which is that I think one of the things ChatGPT's surprising success at mimicry teaches us is that a good portion of the vaunted human intelligence we make such a big deal about is mostly just repeating what we've previously encountered, and anticipating what's going to come next based upon what came next in the past. Here's what I wrote to these two friends:

If I look back over my creative life, there have been a few moments that I would say were truly inspired invention, where I created something from nothing -- something that was actually new. But far and away 99.99999% of everything I do and have done has been wholly derivative. As it happens, I obtain immense satisfaction and even some joy from endlessly solving combinatorial puzzles -- thus I love electronics and coding.

So, to wrap up, I thought it was interesting and not at all surprising that whereas ChatGPT **can** perform quite well at recombining what it's seen in the past to produce new and nearly functional code in the future, it is not going to be able to understand and explain the detailed operation of some piece of purpose-written malware that it has never encountered before.

Though ChatGPT was initially a surprise, and though I'm sure that this technology is going to continue to improve over time, I believe that we now have a good foundation for understanding what it can and cannot do. At least for the foreseeable future, it is, at most, a very clever regurgitator.