Security Now! #943 - 10-10-23

The Top 10 Cybersecurity Misconfigurations

This week on Security Now!

How many people have downloaded GRC's latest freeware so far? Do we believe what 23andMe have told the world about the leak of their customers' personal and private data? What are the stats regarding all aspects of cyberattacks? How's the Brave Browser doing? Where and when is Google surreptitiously embedding tracking links into Google Docs exports? What high profile enterprise was also compromised by the Progress Software MOVEit SQL injection? What additional web browser just added and announced its support for Encrypted ClientHello? What change did Google just make with the release of their Pixel 8 family of smartphones? What cyber initiative did the U.S. Congress just overwhelming pass? What's "DwellTime" and why do we care? And that's just the news. We'll also be entertaining many of our listeners' questions, then starting into the first part of our examination of a really terrific document that was just published by the NSA and CISA.

4000 years ago man built the pyramids...

It's been downhill ever since.

ValiDrive

Since I know many of our listeners have been waiting for it, I wanted to start off by noting that GRC's latest freeware utility, ValiDrive – which was born during this podcast when Leo was taken aback by my description of the surprising problem that exists with fraudulent drives – has been published and it's being downloaded briskly. As we know, users are purchasing drives which they believe are a certain size, but those drives are deliberately – not mistakenly – declaring themselves to be far larger than they really are. I've heard from several users of ValiDrive who have been surprised. I purchased twelve 1 or 2 terabyte drives from Amazon during the work on ValiDrive... and **every single one of them was fraudulent**:

... even a few that arrived in quite elaborate packaging were lying.

ValiDrive spot-checks 576 evenly-space regions across the drive's declared size. At each region it reads what's there for safe keeping, then writes a unique pattern of noise, then reads that region back and verifies that the data that was read back matches what was written. Then in either case it restores the original data so it can be safely used on drives already containing data.

ValiDrive also measures the time required to perform all of the reading and writing and once the testing is completed it shows statistics about the drive being tested. One of the most interesting things that has been revealed is that drives vary widely in their ability to switch back and forth between reading and writing. Or more specifically, in their ability to begin writing when they've been reading. As I've mentioned before, NAND style flash storage, uses a high voltage, typically around 20 volts, to break down barrier insulation during erasing and writing. Since only 5 volts is available to USB devices, circuitry known as a charge pump is used to boost the lower 5 volts up to 20 volts. But this boosting takes time. Essentially, some capacitors in the NAND circuitry need to be charged up to 20 volts before any erasing and rewriting can be performed.

A significant pre-write delay doesn't cause trouble in normal use because entire files or multiple files are often written all at once. So only one charging cycle is required at the start of writing what might be many megabytes of storage. This means that the time required to begin writing is

amortized across all of the writing that's done at once. But since ValiDrive only writes or reads 4k bytes at a time, no amortization takes place.

Anyway, ValiDrive is about 95 Kbytes and needs no installation. It's finished, ready for the world, and I expect it's going to have a long and happy life. And as for me, I'm getting back to finishing up SpinRite v6.1 so that before long we can also be celebrating its release.

But first we have some news and listener feedback to share...

Security News

23andMe accounts breached

Late last week, after internal private customer data was found circulating on hacker forums, the well known DNA aggregation, analysis and profiling service 23andMe announced that the accounts of some of its users had been accessed through credential stuffing attacks. The first three paragraphs of their announcement said:

> *We recently learned that certain 23andMe customer profile information that they opted into sharing through our DNA Relatives feature, was compiled from individual 23andMe.com accounts without the account users' authorization.*
>
> *After learning of suspicious activity, we immediately began an investigation. While we are continuing to investigate this matter, we believe threat actors were able to access certain accounts in instances where users recycled login credentials – that is, usernames and passwords that were used on 23andMe.com were the same as those used on other websites that have been previously hacked.*
>
> *We believe that the threat actor may have then, in violation of our Terms of Service, accessed 23andMe.com accounts without authorization and obtained information from certain accounts, including information about users' DNA Relatives profiles, to the extent a user opted into that service.*

I don't believe them. The more I've thought about this, the less it appears to pass the smell test. My problem is the sheer quantity of customer records that were apparently retrieved – the attacker claims to have 7 million records which is half of 23andMe's user base. And the nature of the data fields leaked look very much like internal raw database records. The initial data leak on hacker forums consisted of 1 million records data for Ashkenazi Jewish people. And millions more are, we're told by the hackers, available for purchase.

So, if we're to believe 23andMe, they're saying that all the attackers did was login as some other **valid** customers, and that they were then able to obtain apparently complete records for millions of other 23andMe users? In other words, any 23andMe user can login as themselves then do the same thing? Really. That's what we're being asked to believe?

It's not my business to pursue this. But many people are quite sensitive to the disclosure of their very personal and private, and especially genetic, information. So I hope that someone in a position of authority digs a bit deeper and asks 23andMe to further explain what, to me, from what we've seen so far, looks like a P.R. cover-up of a far larger problem; and they should be grown up enough to know better.

Microsoft Digital Defense Report 2023

Microsoft just published their 2023 Digital Defense report. It contains a number of important facts that are worth sharing:

- Of the 78% of IoT devices with known vulnerabilities on customer networks, 46% are not patchable.
- Since 2019, attacks targeting open-source software have grown on average 742%.
- Fewer than 15% of non-governmental organizations (NGOs) have cybersecurity experts on staff.
- Coin-mining activity was found in 4.2% of all incident response engagements.
- 17% of intrusions involved known remote monitoring and management (RMM) tools.
- Adversary in the middle (AitM) phishing domains grew from 2,000 active domains in June 2022 to more than 9,000 by April 2023.
- 156,000 daily business eMail compromise attempts were observed between April 2022 and April 2023.
- 41% of the threat notifications Microsoft sent to online services customers between July 2022 and June 2023 went to critical infrastructure organizations.
- The first quarter of 2023 saw a dramatic surge in password-based attacks against cloud identities.
- Microsoft blocked an average of 4,000 password attacks per second over the past year.
- Approximately 6,000 MFA fatigue attempts were observed per day.
- The number of token replay attacks has doubled since last year, with an average of 11 detections per 100,000 active users in Azure Active Directory Identity Protection.
- DDoS attacks are on the rise, with around 1,700 attacks taking place each day, cumulating at up to 90 terabits of data per second (Tbps).
- State-sponsored activity pivoted away from high-volume destructive attacks in favor of espionage campaigns.
- 50% of destructive Russian attacks observed against Ukrainian networks occurred in the first six weeks of the war.
- Ghostwriter continues to conduct influence campaigns attempting to sow distrust between Ukrainian populations and European partners who support Kyiv—both governmental and civilian.
- Iranian operations have expanded from Israel and the US to target Western democracies and NATO.

This is not an industry in which we would like to see such growth. It would be far better if things were pretty much as bad this year as they were last year. But, unfortunately, that's not what's happening. We're seeing real growth in cybercrime activity – and success – across the spectrum of criminal activities.

Brave lays off 9% of its workforce
It's unclear what, if anything, this might mean for the future of the Brave browser, but Brave Software confirmed that it has laid off 9% of its staff across departments. The company didn't indicate how many people were affected, but it said the decision was driven by the tough economic climate saying: "Brave eliminated some positions as part of our cost management in this challenging economic environment. Several departments were affected, amounting to 9% of our staff."

Brave had already been taking steps to bolster its revenue sources this year. In April, Brave Search dropped Bing's index to start relying on its own indexing solution. In May, the company released its own search API for clients, with plans starting from $3 per 1,000 queries. The API also offers different plans for AI data model training, data with storage rights, spellcheck, and autosuggest. Last month, Brave introduced image, news, and video results as part of its Search API.

And, Leo, Brave has named their forthcoming AI assistant after you... or at least they are calling it "Leo". While the plan is for Leo to be available to all users, Leo will also have a premium tier to offer features like higher rate limits and access to more conversation models. That paid premium tier will help pay for the cost of API access and hosting.

From someone named "Joe" posted on FOSSTODON:

> *"Today I found out that google docs infects HTML exports with spyware, no scripts, but links in your document are replaced with invisible google tracking redirects. I was using their software because a friend wanted me to work with him on a google doc. He's a pretty big fan of their software, but we were both absolutely shocked that they would go that far."*

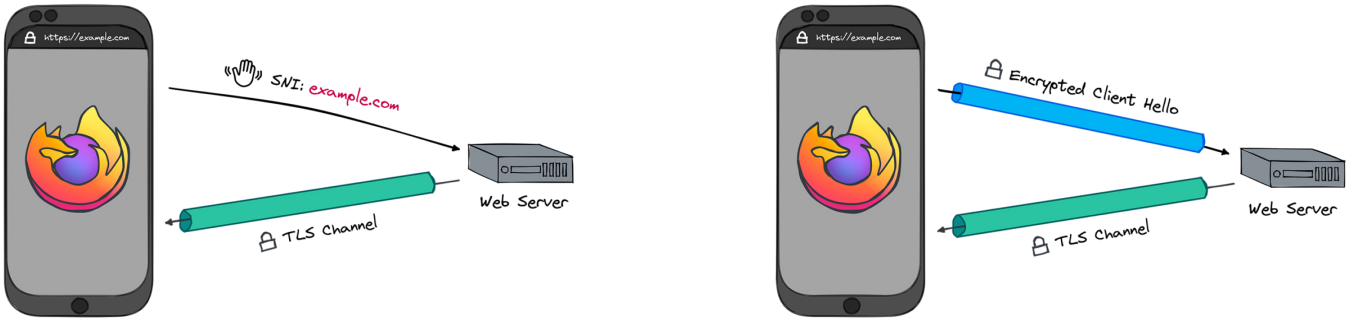
I was curious so I first exported these show notes as a PDF and as far as I could tell their embedded links were clean. But I wanted to see whether I could substantiate Joe's claim. So I then exported the show notes as HTML and, sure enough, the embedded URLs all point to Google with the original, visible URL as a parameter, along with a bunch of tracking gibberish:
<a class="c18"
href="https://www.google.com/url?q=https://grc.sc/842&sa=D&source=editors&ust=1696905303796344&usg=AOvVaw3CKwjT50ji8eSwiJXJGs1W">https://grc.sc/842

I wouldn't go so far as to call this spyware, but this is clearly deliberate and it sure doesn't seem like any of Google's business what links are clicked in the future on in an HTML export of something created by Google Docs. So I thought everyone should know.

More MOVEit Mess
Our podcast #928 of June 20th was titled "The Massive MOVEit Maelstrom". It just came to light due to a disclosure Sony filed with US authorities that Sony Interactive Entertainment was among the now more than **2,300 companies** who were impacted by the exploitation of the SQL injection vulnerability in Progress Software's MOVEit file transfer system. Sony said that the hackers stole data on current and former employees and their families and that nearly 7,000 people were impacted.

Firefox 118 gets ECH

While we were talking about Encrypted ClientHello last week, Mozilla was busy announcing it in Firefox 118 which, I just checked, is what I'm sitting in front of right now. Mozilla's announcement page has a very clean and simple graphic to highlight the difference between non-encrypted ClientHello and encrypted ClientHello. I've grabbed the graphics for the show notes:



Mozilla summed up last week's deep dive quite succinctly. They wrote: "ECH uses a public key fetched over the Domain Name System (DNS) to encrypt the first message between a browser and a website, protecting the name of the visited website from prying eyes and dramatically improving user privacy." Later they explain that ECH must be paired with DNS over HTTPS (DoH) to secure and hide that initial public key fetch over DNS.

As we know, Cloudflare's web proxy frontend has switched on ECH support for all of their free tier client sites. Now we need the other stand alone web servers to catch up and offer support. An experimental implementation for OpenSSL exists, so that will help to get the Unix and Linux based servers onboard the ECH train. And in time it seems clear that the rest will join, since user privacy is dependent upon the site they are connecting to offering its support for the privacy of their incoming connections.

The Pixel 8 & Pro, now with two additional years of updates!

Last Wednesday, Google announced that its newest Pixel 8 and Pixel 8 Pro smartphones will now receive seven years of software and security updates. That's a two-year bump from the previous 5-year support they had been providing for their Pixel devices. We understand the importance of long-term security updates for consumer devices. Updates really do need to extend throughout the useful life of consumer devices. And smartphones are certainly among those devices needing the most protection. No other device is more prone to attack than smartphones, and few other devices are as complex or as exposed to the multiple channels of external material.

The MACE Act Passed.

The MACE Act Passed. What's the MACE Act you're likely wondering? MACE stands for Nancy Mace who was coincidentally one of the bill's co-sponsors. Nancy is a Republican representative from South Carolina. But the acronym "MACE" aside from being Nancy's last name, stands for "Modernizing the Acquisition of Cybersecurity Experts" and the US House of representatives, which is presently having a difficult time agreeing on the time of day, is apparently in wild agreement over this idea, since the Act passed through the house on a vote of 394 to 1 – which

really does make me curious to know who the holdout was? Anyway, the other co-sponsor of this bipartisan bill was California Democrat Katie Porter, who is just about as far to the left as Nancy is to the right. But on this they agree.

The MACE Act is aimed at addressing shortages in federal cybersecurity positions by expanding the pool of eligible applicants by lowering the education requirements. Under the legislation, agencies would be allowed to consider an applicant's education only if their education directly reflects the competencies required for the position. The bill would also require the Office of Personnel Management to publish annual reports detailing changes to minimum qualifications for cybersecurity positions and data on the education level of people in those positions.

The bill is now headed to the Senate where it is expected to pass as well.

Dwell time plummets

Those who track, monitor and remediate the consequences of ransomware attacks use the term "dwell time" to reflect the time from the initial network intrusion until the ransomware encryption event is triggered. So those monitoring this dwell time are reporting that attackers are deploying ransomware on breached networks faster than ever before. In just 12 months, the median dwell time of ransomware groups on hacked networks has fallen from 4.5 days to less than one day. According to the security firm Secureworks, ransomware is now being deployed within one day of initial access in more than half of all engagements and in as few as five hours of initial network penetration in 10% of all cases.

One of the reasons for this is that the percentage of human-driven attacks has risen dramatically. At the time, several years ago, we talked about how bad guys would have a pending inventory of victims that they would "get around to" when they were able. This victim pool came from automated scanning and attack malware that would get into a host network, settle down and phone home, logging in with its command and control network.

But that was then. These days the majority of attacks are human driven in real time; so a live human attacker is now doing the penetrating. They quickly get the lay of the land, see what's what, and if the network appears to be a useful victim they will immediately set about exfiltrating what they can find and prepare to encrypt all of the systems within reach... all in 10% of cases in fewer than five hours.

Closing the Loop

David Sherman / @davidhsherman13

@SGgrc / Which is better - Passkeys via Bitwarden or Passkeys via browser? How to sync passkeys among different browsers? Thanks

I think the best answer is that Passkey management is still too new and too much in flux for any answer to necessarily hold for long. But there's an essential point I don't think I've made clear enough in the past. Sadly, passkey management has been shrouded in hocus pocus because the purveyors of passkeys have, from all appearances, desperately hoped to be able to use the mysticism surrounding this promising new technology to retain and corral their users into their proprietary environments. The universally missing feature of simple passkey export and import is astonishing to me.

We all know what a password is. We make them up and use password managers or our browsers to remember and regurgitate them on demand. But no one really knows what a passkey is because no one has ever actually seen one. It's all just mysterious "don't worry we got this" hocus pocus behind the scenes and we're told that it's all super wonderful. So here it is: A passkey is nothing more than a private key. Period. And as such it's not some impossible-to-represent mystical token. It's just a relatively short blob of binary data. As such, it could easily be turned into a QR code, or into a short and manageable Base64 encoded string of text. And at that point it could be moved from place to place, printed out and stored in a safe place. All of a user's current passkeys could be exported as a CSV file for safe keeping.

But for some bizarre reason end users are not allowed any of those freedoms today. Expert users want it, and everyone could have it. But no. However, I don't expect this mystical barrier to survive in the long term because it is trivial to export and import passkeys for backup and cross-platform sharing. So someone will be the first to create a simple QR code, textual or CSV file passkey export and import. Then everyone will want it and it will become a required feature for any passkey system. But we're certainly not there today because the system is still so new and passkey support hasn't yet become the commodity that it eventually will.

Henrik Lexow / @LexowHenrik *Offers us a view from inside an ISP...*

Hi Steve, After your deep dive into encrypting the Client Hello, an old internal dilemma has resurfaced. I've spent many years at a European ISP, where ideas about selling content-filtered access control or insights would occasionally come up. Some of the veteran business folks, who had experienced the unencrypted internet era, demanded similar services here. I had to explain that HTTPS Everywhere was happening, but we often circled back to the possibility of inspecting TLS packets for user insights. This usually boiled down to two "customer value areas": child content protection and enterprise content filtering, with the underlying goal of monetizing data from blanket internet use inspection. (We eventually did not offer these services during my tenure.)

My ongoing struggle revolves around balancing parental concerns for children's online safety with the goal of internet privacy. ECH, despite its complexity, is a step in the right direction, as I'm aware of the ISP efforts to tap into this revenue stream while publicly championing

privacy. ECH can put an end to this hypocrisy. But, as is often asked in privacy, "what about the children?"

My question to you, Steve, relates to those two "customer value areas" in the context of ECH becoming a reality:

1. Who should take responsibility for child protection services now that ECH is coming? Not all children have parents who can actively monitor their online experiences. And how could it be done correctly?

2. Are there valid reasons for enterprises to engage in content filtering, be it for security or other purposes? I personally struggle to see the value in this. In the case of spending time on non-work activities on the internet, it seems more like a cultural issue within the company rather than an internet access problem.

Best regards, Henrik

In answer to Henrik's first question: *"Who should take responsibility for child protection in a world where TLS packet filtering will finally be thwarted?"* The first thought that comes to mind is local DNS service filtering. I think that's the best solution by far. If a family's home is using, for example, the free OpenDNS Family Shield Service, which is as easy as configuring the residential router to use a specific pair of IP addresses, (208.67.222.123 & 208.67.220.123) then there's no need to see into any aspect of the TLS connections once they have left the family's router since unwanted domains will never have their IP addresses resolved.

And Henrik's second question: *"Are there valid reasons for enterprises to engage in content filtering, be it for security or other purposes?"* I doubt that the addition of Encrypted ClientHello will change the enterprise environment much. As we've talked about in the past, an enterprise's network is owned by the enterprise and all of its employees should be informed and aware that the content of the enterprise's network is not private and that nothing they do over that network should be considered private. We've talked about placing a strip of paper to constantly remind everyone of that fact along the top of every company-owned computer monitor, and of having the Human Resources department remind every employee of that every year during their annual review.

So that's the enterprise policy view. The practical implementation of such technology within the enterprise is already well established. All non-proxied TLS connections will be blocked from leaving the enterprise's network and any system wishing to connect to the external Internet will need to have the enterprise's proxy server's root certificate added into its root store. When that's been done, all outbound TLS connections will be intercepted and accepted by the enterprise proxy middlebox which will, in turn, connect on behalf of each user to the remote resource while being able to fully examine and filter every network connection. So for the enterprise, the addition of Encrypted ClientHello won't change anything. The enterprise will need to update their middlebox's firmware to add support for ECH, but otherwise life goes on.

Matthew Cowgur / @mattcowgur

Does having a very good password (mine is ~140 bits of entropy) have any effect on encryption with a low number of iterations?

It's interesting that this is not a question that we've directly addressed during our focus upon the question of iteration count. We've exclusively focused upon iteration count as the way to slow down any brute force password guessing attack. But the only reason such an attack needs to be slowed down is because we're assuming that the user has not chosen a super-strong password. That means that given a sufficient number of guesses, the attacker will eventually obtain the not-very-strong password.

Wikipedia reminds us that in one study of half a million users, the average password entropy was estimated at 40.54 bits. And the medium to strong password threshold is regarded as around 50 bits of entropy. So that's nine and a half more bits than the study quoted by Wikipedia. Thanks to the power of powers of two, '2' (because they are binary bits) raised to the power of 9.5 is 724. So the difference in brute force resistance between that study-average password and one that's on the lower-border of strong, is 724 times.

So now let's look at Matthew's question. Matthew claims to be using what he calls a "very good password" having around 140 bits of entropy. Now that we've created some context, it should be clear that if Matthew is correct, and his password truly contains around 140 bits of entropy, it's not "a very good password" it's an insanely good password. If 50 bits of entropy is the lower bound of a good password, then Matthew's has added 90 bits to that. Once again, turning to the power of powers of two, 90 additional bits of entropy results in a password that's 1.238 x 10^27. To help us understand the size of that number, 27 is 3 x 9 and 9 zeros is a billion which we have three times. So Matthew's 140-bit entropy password is 1.238 billion billion billion times stronger than a strong password.

So Mattew's question was: *"Does having a very good password (mine is ~140 bits of entropy) have any effect on encryption with a low number of iterations?"* And the answer is yes. If you have such a password you'd be fine with ZERO iterations. Since the only reason we iterate is to protect weak passwords, if you really have a strong password you have nothing to worry about. When we change the iteration count from 5,000 to 600,000, we've increased the attacking difficulty by a factor of 120. But just 7 binary bits is 128. So a password having just 7 additional bits of entropy would provide more cracking resistance than jump the iteration count from 5,000 to 600,000.

The key takeaway is that increasing iteration counts is a linear increase whereas adding bits of entropy is exponential. Every bit of entropy added doubles the cracking difficulty. So, anyone who has a really strong high-entropy password, even if it's not as ridiculously strong as Matthew's, has nothing to worry about regarding PBKDF iterations. Increasing iteration counts is far weaker protection than using a truly strong password.

IcyvRan TocVuc / @IcyvRan

Hi Steve, according to this TorrentFreak article "Cloudflare has enabled Encrypted Client Hello for all customers on free plans, which includes many pirate sites. The new privacy feature makes it impossible for Internet providers to track which websites subscribers visit. As a result, it also renders pirate site-blocking efforts useless, if both the site and the visitor have ECH enabled."
https://torrentfreak.com/encrypted-client-hello-ech-effectively-defeats-pirate-site-blocking-231006/

This is inevitable, and it's analogous to the encryption debate, right? We want to enhance privacy, but we're unable to enhance only the good guy's privacy. Everyone gets their privacy enhanced, even those who will criminally abuse that privacy. The question the world has been struggling with is whether our inability to restrict who gets more privacy means that no one should have more. But it's looking like the world is going to agree that giving more to everyone is the best solution.

The Torrentfreak article that our listener linked to was interesting and it shed a different light onto the emerging presence of Encrypted ClientHello connections. I've trimmed it down a bit to remove the stuff we already know, but here's what Torrentfreak observed:

Cloudflare has enabled Encrypted Client Hello for all customers on free plans, which includes many pirate sites. The new privacy feature makes it impossible for Internet providers to track which websites subscribers visit. As a result, it also renders pirate site-blocking efforts useless, if both the site and the visitor have ECH enabled.

Website blocking has become the go-to anti-piracy measure for the entertainment industries when tackling pirate sites on the internet. The practice has been around for well over 15 years and has gradually expanded to more than forty countries around the world. The actual blocking is done by Internet providers, often following a court order. These measures can range from simple DNS blocks to more elaborate schemes involving Server Name Indication (SNI) eavesdropping, or a combination of both.

Thus far, the more thorough blocking efforts have worked relatively well. However, as privacy concerns grew, new interfering technologies have emerged. Encrypted DNS and SNI, for example, made blocking efforts much harder, although not impossible.

A few days ago, Internet infrastructure company Cloudflare implemented widespread support for Encrypted Client Hello (ECH), a privacy technology that aims to render web traffic surveillance futile. This means that site blocking implemented by ISPs will be rendered useless in most, if not all cases. ECH is a newly proposed privacy standard that's been in the making for a few years. The goal is to increase privacy for Internet users and it has already gained support from Chrome, Firefox, Edge, and other browsers. Users can enable it in the settings, which may still be experimental in some cases.

The main barrier to widespread adoption is that this privacy technology requires support from both ends – websites have to support it as well. Cloudflare has made a huge leap forward on that front by enabling it by default on all free plans, which currently serve millions of sites. Other subscribers can apply to have it enabled.

The push for increased privacy is well-intended, but for rightsholders it represents a major drawback too; when correctly configured, ECH defeats site-blocking efforts. Tests conducted by TorrentFreak show that ISP blocking measures in the UK, the Netherlands, and Spain were rendered ineffective.

This doesn't automatically apply to all blocked sites, as the sites must have ECH enabled too. We have seen mixed results for The Pirate Bay, perhaps because it has a paid Cloudflare plan, but other pirate sites are easily unblocked.

This new privacy feature hasn't gone unnoticed by pirate site operators. The people behind the Spanish torrent site DonTorrent, which had dozens of domains blocked locally, are encouraging users to try ECH.

DonTorrent notes: "Before ECH, your online privacy was like a secret whispered in the wind, easily picked up by prying ears. But now, with ECH by your side, your data is like hidden treasure on a remote island, inaccessible to anyone trying to get there without the right key. This feature encrypts your data so that neither ISPs nor organizations like ACE and MPA [can] censor, persecute and intimidate websites that they consider 'illegal'."

Cloudflare and other tech companies are not supporting ECH to make site-blocking efforts obsolete. However, this privacy progress likely won't be welcomed by rightsholders, who've repeatedly criticized Cloudflare for hiding the hosting locations of pirate sites. TorrentFreak reached out to a major anti-piracy organization for a comment on these new developments, but we have yet to receive an on-the-record response. It wouldn't be unthinkable, however, that we will see more blocking lawsuits against Cloudflare in the future.

We touched upon the content filtering thwarting aspect of ECH last week, but we didn't explore the real-world consequences to those, like the Motion Pictures Association, who have been using legal means to enforce the blocking of sites containing pirated copyrighted content.

As I said at the top of this, everyone getting their privacy increased means that the bad guys do, too!

SKYNET / @fairlane32

This ECH stuff is going to mess with my public wifi content filtering isn't it. I JUST got my vendor to figure out what was blocking patrons ebooks from being downloaded onto their Kindles and now this. Thanks Steve. Thanks a lot 😑 🤣🤣

Okay, first of all, this is not my doing. I'm just reporting the facts! <grin> But as for it messing with someone's public WiFi content filtering, uhhhhh, yep, it's going to do that.

I presume that ECH is somehow protecting itself from protocol downgrade attacks. We've talked a lot about them in the past because they can be tricky to prevent. The original downgrades were HTTPS to HTTP. On an HTTP connection that was attempting to switch to HTTPS, since the connection was not encrypted yet, something would simply change all HTTPS URLs into HTTP, keeping the connection unencrypted and leading each other to assume that the other end had a problem with encryption when that was not true.

In the case of ECH, for example, it only works when both ends support it. And at this early point in time, support is probably more surprising than not. So, if an outgoing or incoming initial TLS handshake packet were to be tweaked to show that ECH is not supported, the other end would shrug and not be surprised. Meanwhile the domain name would be exposed to anyone watching the traffic. Hopefully, ECH's designers were aware of this and came up with some means of preventing middlemen from removing ECH support from connections before they get started. Otherwise ECH may only be a temporary inconvenience.

In the case of Skynet's public WiFi, any ECH-using clients would also be using DNS over HTTPS (DoH) and thus not his local access point's DNS. So filtering DNS wouldn't work, either.

Anonymized by Steve – He didn't ask me to, but I feel as though this person's tweet should be anonymous.

Hey Steve. You mentioned in a recent episode that the linux kernel has fixed the epoch time issue, in kernel build 5.10. I feel like that might have put too many people at ease. Not only are old devices still running out of date kernels but modern stuff does too.

As I have mentioned before, I work for Check Point, a company with firewalls protecting a massive portion of the internet. We serve millions of businesses, including all of the Fortune 500 companies, which account for decent chunks of the packets moving across the internet, all having to pass through one of our firewalls.

After many years of fighting with R&D, Check Point finally upgraded its OS to move away from Linux Kernel 2.6.18... to 3.10 as of 2020. This is the latest anyone can run. Still with the epoch code issues. I shudder at the thought that we still have customers running code that has passed end of life over a decade ago. Even if R&D begins working on a migration to kernel 5.10 today, it would take years to finish and decades to move everyone off the older releases. All of that on current, modern, state of the art key infrastructure that the world relies on. I think people should still expect a huge mess at the end of epoch time.

Thank you for the wonderful show. Looking forward to more episodes to 999 and beyond. And to not having to use Twitter anymore. All the best, [XXX]

That note speaks for itself and I thought it was a valuable look inside the reality of the commercial use of Linux-based appliances. There really is an "if it's not broken let's not break it" mentality. And, really, after Check Point built a robust firewall architecture on top of an old OS kernel... if it's working, why change it? If it's primarily being used to boot one's own code, and important things like the OpenSSL library can be kept current, then why mess with the boot loader? The problem is, the earlier Linux file system timestamps are using the signed 32-bit time... so any reliance upon file times is going to go berserk in 2038. But that's still 15 years off.

Matt organ / @Slater450413

Hey Steve, There's something that still bothers me about that recent Microsoft hack. There were far too many coincidences where the attacker "happened to know" various flaws.

Just to interject here, Matt's referring to the fact that not only was the secret key resident in RAM and then captured by a system crash dump, but then that a chain of no fewer than every one of five flaws were required for that crash dump image to finally reach an attacker. Anyway, Matt continues...

Unlike regular flaws in a desktop OS where you can continually poke at it with a debugger attached, most of the surface on a cloud instance being attacked also happens to be blind to outcome observation (like the modern version of blind SQL injection), yet pinpoint accuracy was achieved anyway multiple times.

It's almost as if the attacker had source code to examine, sort of like the access and download that was achieved during the solarwinds attack 2 years ago. I don't like conspiracy theories and I'm aware we will never know, but this seems far more likely than an attacker guessing their way through this much obscure and highly technical knowledge.

I remember back on Security Now episode #800 there was a passing comment Microsoft made publicly that they do not rely on source code being kept secret as a security feature.
But I wonder if this accidental visibility is now coming home to roost.

As Matt says, we'll never know. But it was, if nothing more, a surprisingly glaring sequence of successive failures that brought the key into the hands of someone who knew what to do with it.

Errata:

I mis-numbered last week's shortcut of the week **842** instead of **942**. So, technically, that would have made it a shortcut of the week from about two years ago, which was not what I intended. Elaine who transcribed the podcast and several of our listeners picked up on my mistake. That shortcut, just to remind everyone, was to the very nice YouTube video explaining how to set up SyncThing. So, for anyone who may have thought "Hey, that's great, Steve created a shortcut for this week's podcast #942!", and then found that it didn't work, the link I created was for https://grc.sc/842. Sorry for the confusion.

And, when I was setting up **this** week's properly-numbered shortcut of the week I noticed that 555 of our listeners took me literally and followed the #842 link to learn more about SyncThing ... so I'm glad for that.

The Top 10 Cybersecurity Misconfigurations

https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a

Last Thursday, the U.S. National Security Agency (our NSA) and the Cybersecurity and Infrastructure Security Agency (our CISA) jointly published a cybersecurity advisory. Of course, everything has initials, so it's the "CSA." This advisory was the result of NSA and CISA red team and blue team activities as well as the activities of both agencies' Hunt and Incident Response (HIR) teams. The advisory identifies and highlights the most common cybersecurity misconfigurations which they continually uncover within organizations and the report details the tactics, techniques, and procedures (TTPs) which the bad guys use to exploit these misconfigurations.

Checklists such as this can be very useful because, over and over and over in this podcast we encounter the many consequences of the tyranny of the default. You add multifactor authentication to your network. Great. Good move. But did you carefully read through all of the cautions that its publisher included? Has this new facility been fully configured correctly? Or did it take longer than you expected to get it going at all, so now you're late for a meeting and had to run off leaving it – probably forever – just as it came out of the box?

Reading through this advisory, I feel as though this podcast has been serving its listeners well, since nothing here will surprise anyone who's been listening for long. These are the topics we often focus on. In fact, #1 on the list is "Default configurations of software and applications."

So today, we're going to hit the high points on this advisory. But this work drills down into very useful and actionable specifics – which I'm certain was their intention. This makes it too long to cover at that level of detail here. But every listener who's responsible for their enterprise's network security would do well to spend some time with it on their own. It's available as both a web page and as a very nicely formatted 44-page PDF. So I've made the PDF edition, this week's carefully-numbered GRC shortcut of the week. So grc.sc/943 **will** redirect you to the PDF which resides at defense.gov.

Okay, so without further ado, I'm going to quickly enumerate these top-10 most common and most troublesome cybersecurity network-related misconfiguration issues, then we'll dig into each one a bit deeper. The top 10 are:

1. **Default** configurations of software and applications
2. Improper separation of user/administrator **privilege**
3. Insufficient internal network **monitoring**
4. Lack of network **segmentation**
5. Poor **patch** management (in other words, not keeping up with updates)
6. Bypass of system access controls
7. Weak or misconfigured multifactor authentication (MFA) methods
8. Insufficient access control lists (ACLs) on network shares and services
9. Poor credential hygiene
10. Unrestricted code execution

The NSA and CISA then elaborate upon these a bit, writing:

These misconfigurations illustrate (1) a trend of systemic weaknesses in many large organizations, including those with mature cyber postures, and (2) the importance of software manufacturers embracing secure-by-design principles to reduce the burden on network defenders:

- *Properly trained, staffed, and funded network security teams can implement the known mitigations for these weaknesses.*
- *Software manufacturers must reduce the prevalence of these misconfigurations—thus strengthening the security posture for customers—by incorporating secure-by-design and -default principles and tactics into their software development practices.[1]*

NSA and CISA encourage network defenders to implement the recommendations found within the Mitigations section of this advisory—including the following—to reduce the risk of malicious actors exploiting the identified misconfigurations.

- *Remove default credentials and harden configurations.*
- *Disable unused services and implement access controls*
- *Update regularly and automate patching, prioritizing patching of known exploited vulnerabilities.[2]*
- *Reduce, restrict, audit, and monitor administrative accounts and privileges.*

NSA and CISA urge software manufacturers to take ownership of improving security outcomes of their customers by embracing secure-by-design and-default tactics, including:

- *Embedding security controls into product architecture from the start of development and throughout the entire software development lifecycle (SDLC).*
- *Eliminating default passwords.*
- *Providing high-quality audit logs to customers at no extra charge.*
- *Mandating MFA, ideally phishing-resistant, for privileged users and making MFA a default rather than opt-in feature.[3]*

In other words, both parties – the software manufacturer and the software user – have responsibilities. A perfect example that they mention is default passwords. The users of any system MUST change the default passwords when they are first setting up their software, but the creators of that software should also absolutely come up with some way to avoid ever having a default password in the first place.

Just how serious is this simple-seeming problem of default credentials? It's a bit shocking when you look at how many different ways these "defaults" can be abused. They explain:

Many software manufacturers release commercial off-the-shelf (COTS) network devices —which provide user access via applications or web portals—containing predefined default credentials for their built-in administrative accounts. Malicious actors and assessment teams regularly abuse default credentials by:

- *Finding credentials with a simple web search and using them to gain authenticated access to a device.*
- *Resetting built-in administrative accounts via predictable forgotten passwords questions.*
- *Leveraging default virtual private network (VPN) credentials for internal network access.*
- *Leveraging publicly available setup information to identify built-in administrative credentials for web applications and gaining access to the application and its underlying database.*
- *Leveraging default credentials on software deployment toolsfor code execution and lateral movement.*

In addition to devices that provide network access, printers, scanners, security cameras, conference room audiovisual (AV) equipment, voice over internet protocol (VoIP) phones, and internet of things (IoT) devices commonly contain default credentials that can be used for easy unauthorized access to these devices as well. Further compounding this problem, printers and scanners may have privileged domain accounts loaded so that users can easily scan documents and upload them to a shared drive or email them. Malicious actors who gain access to a printer or scanner using default credentials can use the loaded privileged domain accounts to move laterally from the device and compromise the domain.

My feeling is, that the awareness of the danger posed by dangerous defaults of any kind has been very well known for decades. So any manufacturer who's still shipping products with dangerous default settings which they expect their customers to know to change, is just being lazy. Many if not most users – even, obviously, at the enterprise level – presume that the way things come from the factory are intended to be the way they should be – thus the tyranny of the default. If this was not the case, it wouldn't be the "tyranny of the default", it would be the "blessing of the default."

There's the occasional sighting of a manufacturer who gets it. Who **requires** their user to invent their own admin password right off the bat, during the initial setup and configuration of a device. These devices have no defaults so they are inherently far more secure. But even today, such sightings are the exception rather than the rule. So the responsibility rests upon those of us who use these things.

The document notes, also, some interesting specifics, writing:

Certain services may have overly permissive access controls or vulnerable configurations by default. Additionally, even if the providers do not enable these services by default, malicious actors can easily abuse these services if users or administrators enable them. Assessment teams regularly find the following:

- *Insecure Active Directory Certificate Services*
- *Insecure legacy protocols/services*
- *Insecure Server Message Block (SMB) service*

Looking more closely at legacy protocol and services and insecure server message block (SMB) services, they note:

> *Many vulnerable network services are enabled by default, and assessment teams have observed them enabled in production environments. Specifically, assessment teams have observed Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), which are Microsoft Windows components that serve as alternate methods of host identification. If these services are enabled in a network, actors can use spoofing, poisoning, and relay techniques to obtain domain hashes, system access, and potential administrative system sessions.*
>
> *Malicious actors frequently exploit these protocols to compromise entire Windows' environments. Malicious actors can spoof an authoritative source for name resolution on a target network by responding to passing traffic, effectively poisoning the service so that target computers will communicate with an actor-controlled system instead of the intended one. If the requested system requires identification/authentication, the target computer will send the user's username and hash to the actor-controlled system. The actors then collect the hash and crack it offline to obtain the plain text password.*
>
> *The Server Message Block service is a Windows component primarily for file sharing. Its default configuration,* ***including in the latest version of Windows, does not require signing network messages to ensure authenticity and integrity.*** *If SMB servers do not enforce SMB signing, malicious actors can use machine-in-the-middle techniques, such as NTLM relay. Further, malicious actors can combine a lack of SMB signing with the name resolution poisoning issue above to gain access to remote systems without needing to capture and crack any hashes.*

As I'm reading that, another thing occurs to me: There's an aspect of asymmetrical warfare here. These systems have grown to be insanely complex and they're dragging along a growing encrustation of legacy protocol crap so that nothing from the past ever breaks and everything that someone might have continues to work – and this is true even if they don't have any of that stuff – it's all just in case they might.

So, the beleaguered I.T. professional just wants things to work. Sure, he or she also wants them to be secure. But first they have to work. But the bad guys have an entirely different agenda. And I understand that this is obvious, but I think that it's still critically important. The bad guys are living off of this debris. Off of all of this "what if maybe we'll need this someday" legacy crap. They've learned, and know, the in's and out's of how to abuse these retired or retiring systems that persist. And this is the asymmetric aspect of this. As we know, security is about the weakest link. So it literally does no good to have super security on the latest spiffy new network layers if the ancient networking protocols are still left lying around. The enterprise may not be using them, but that doesn't mean that the bad guys will not be abusing them.

We're at page 18 of the show notes, which usually means we're out of time. But there's still so much really good meat here to discuss: Segmentation of user and admin privileges, lack of network monitoring and lack of network segmentation. Poor patch management, bypass of access controls, weak or misconfigured multifactor authentication, and more. So, next week I plan to continue digging into some of the remaining high points of this very important document.