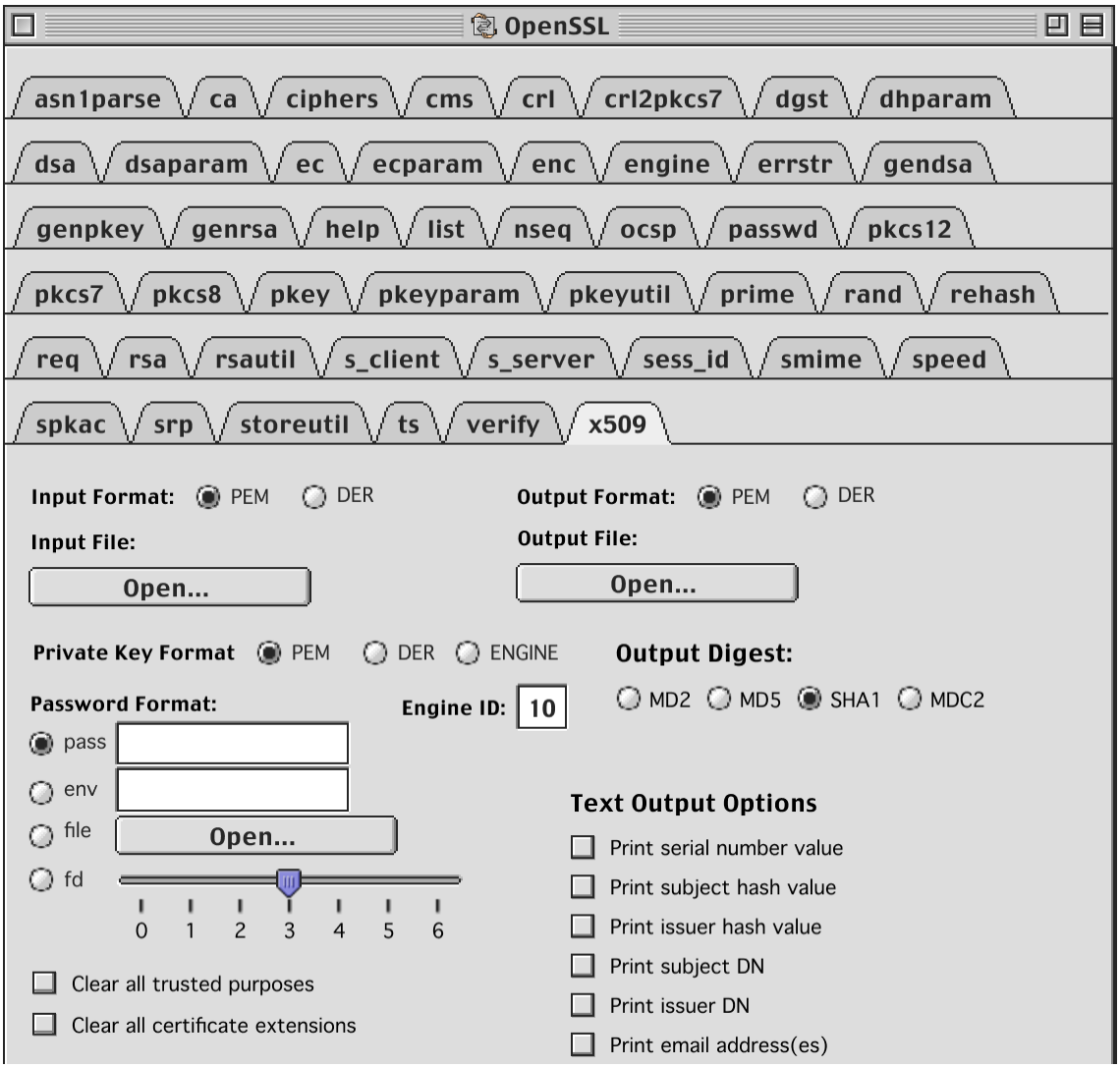
Security Now! #877 - 06-28-22

The “Hertzbleed” Attack

This week on Security Now!

This week, after dealing with a major piece of errata from last week, we look at Germany's reaction to the EU's proposed “let's monitor everyone and privacy be damned” legislation. The Conti gang finally pulls the last plug. We have an update on the status of Log4J and Log4Shell and a weird proposal for a "311" cyberattack reporting number, and a sweeping 56 new vulnerabilities were found and reported across the proprietary technologies of major industrial control technology providers. And this week we have a piece of miscellany, followed by ten interesting items of closing-the-loop feedback to share from our listeners. We will then take a deep dive into the latest “HertzBleed Attack” which leverages the dynamic speed scaling present in today's modern processors. We’ll examine another effective side-channel attack — which is even effective against carefully-written post-quantum crypto — and can be used to reveal its secret keys.

If OpenSSL’s command-line options had a GUI...



Subject & Issuer Name Display Options

- ☐ Use the old format
- ☐ compatible with RFC2253
- ☐ more readable than RFC2253
- ☐ multiline format
- ☐ escape the "special" characters required by RFC2253 in a field
- ☐ escape control characters
- ☐ escape characters with the MSB set
- ☐ quote the string and escape inner quotes
- ☐ convert all strings to UTF8 format first
- ☐ do not attempt to interpret multibyte characters in any way
- ☐ show the type of the ASN1 character string
- ☐ DER hexdump some fields
- ☐ dump any field whose OID is not recognised by OpenSSL
- ☐ reverse the fields of the DN
- ☐ change how the field name is displayed
  - ○ none ○ short ○ long ○ OID
- ☐ places spaces round the = character which follows the field name

- ☐ Print certificate purpose
- ☐ Print the RSA key modulus
- ☐ Print the certificate fingerprint
- ☐ Print OCSP hash values for the subject name and public key
- ☐ Print OCSP responder URL(s)
- ☐ Print not before field
- ☐ Print not after field
- ☐ Print both before and after dates
- ☐ Output the public key
- ☐ Output the certificate in the form of a C source file

Trust Settings

- ☐ Output a trusted certificate
- ☐ Set a certificate alias: [ ]
- ☐ Output the certificate alias
- ☐ Clear permitted uses of the certificate
- ☐ Clear prohibited uses of the certificate

Add trusted certificate uses:

serverAuth	○ permit	○ prohibit
clientAuth	○ permit	○ prohibit
emailProtection	○ permit	○ prohibit

- ☐ Test certificate extensions and output results

Signing Options ("mini CA")

- ☐ self-sign the input file using this private key

Open...

Private Key Format ◉ PEM ○ DER

- ☐ this file is a CSR
- ☐ delete any extensions from the certificate

Validity days

1 5 10 30 90 365 3650

- ☐ convert the input certificate into a CSR

Signing CA File

Open...

Signing CA Private Key File

Open...

Signing CA Serial Number File

Open...

- ☐ create serial number file if it doesn't exist

Subject Key ID

○ RFC3280 hash ○ [ ]

Authority Key ID

☐ keyId ☐ issuer

Certificate Extensions File

Open...

Basic Constraints

- ☐ create a CA
  - ☐ critical
  - ☐ specify path length

0 1 2 3 4 5 6

Key Usage

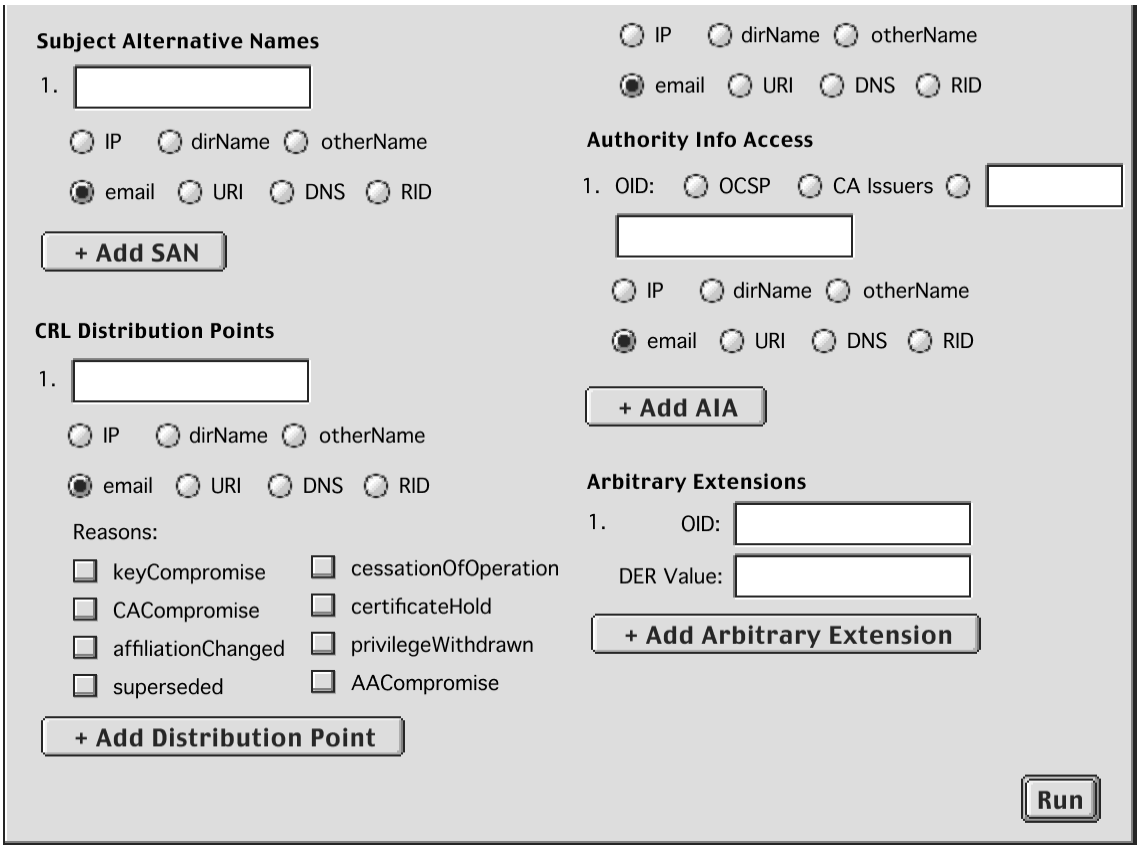
- ☐ digitalSignature
- ☐ nonRepudiation
- ☐ keyEncipherment
- ☐ dataEncipherment
- ☐ keyAgreement
- ☐ keyCertSign
- ☐ cRLSign
- ☐ encipherOnly
- ☐ decipherOnly

Extended Key Usage

- ☐ serverAuth
- ☐ clientAuth
- ☐ codeSigning
- ☐ emailProtection
- ☐ timeStamping
- ☐ msCodeInd
- ☐ msCodeCom
- ☐ msCTLSign
- ☐ msSGC
- ☐ msEFS
- ☐ nsSGC

Issuer Alternative Name

☐ [ ]



Note that even THIS is incomplete. It covers about 80% of one corner of OpenSSL's functionality. The certificate policy options have a lot more knobs that were not included.
(https://smallstep.com/blog/if-openssl-were-a-gui/)

Errata:

Firefox's "Total Cookie Protection"

I have to begin this week by fixing a piece of nonsense from last week: GRC's Cookie Forensics page is working perfectly and just the way it should. I designed that system to detect the presence and freshness of 3rd-party cookies. And it does that just as it should. And Firefox's Total Cookie Protection is also working just as it should.

As I clearly explained last week, before I got myself all tangled up in the operation of my own Cookie Forensics page, the brilliance of Mozilla's approach is that 3rd-party cookies **do still work** just as they always did. What no longer works, and what GRC's Cookie Forensics page **does not test for**, is whether a 3rd-party cookie that's set while at grc.com would be readable through Firefox when the browser is anywhere else. **In other words: Duh!!!**

As I mentioned last week, I maintain a separate www.grctech.com domain just for this purpose. So, in order for me to create a dynamic test for domain-locked 3rd-party cookie sequestration, I would need to have a third test domain which would be used to attempt to read the cookies set for the grctech.com domain while at grc.com. But, as with everything else in my life at the moment, SpinRite v6.1 comes first. So hopefully someone else will do this for us.

Security News

3rd Party FIDO2 Authenticators

I attended the LastPass webinar last Thursday and not much was said that wasn't already known or assumed. I snagged two sides from the presentation.:

The first slide titled *"Also, it's a journey..."* Says under *"Real talk:"* that *"truly removing passwords from the login equation could take years."* And it answers its own question *"Why?"*, Saying *"Universal passwordless access requires FIDO2 support from devices, operating systems (OS), browsers, and authenticators, but also from each website individuals need to access."* So this all had a feel of *"yeah, FIDO2 would be great, but we're not holding our breath and neither should you."*

The second slide is titled "Going Passwordless with LastPass" with the subtitle "Our pathway to achieving open, simple, secure passwordless for all." And this pathway is shown to have three phases:

Phase 1 is Available Now! (exclamation point) "Passwordless login to the vault using LastPass Authenticator."

Phase 2 is "Available Later This Year" (no exclamation point) "Adding FIDO2-supported Authentication (security keys, biometrics) for additional security and flexibility when using passwordless login to your vault." And I'll remind everyone from my digging around in Bitwarden, as I mentioned last week, that Bitwarden already offers the use of FIDO2 login for more secure access to Bitwarden's vault.

Phase 3 of LastPass's three phase plan is "Coming in 2023" and says: "Adding secure storage of passkeys in addition to passwords, to enable consistent passwordless login to all apps and sites." Of course, that's what Apple, Google and Microsoft have all just announced their support for. And neither LastPass nor Bitwarden have solutions for that today. LastPass is saying that we'll be waiting for it from them until sometime in 2023. If we assume for the sake of argument the middle of 2023, that would be one year from now.

And that timing probably works well overall since what website is going to support WebAuthn when no one has any way of logging in with it? It's a chicken and egg dilemma. But I recall when the Internet was viewed the same way: The argument back then was "no one is on the Internet, so why would anyone spend resources creating an Internet presence when no one is there to be present for?" But history shows that it happened anyway. I suspect that FIDO2 will be the same.

Germany's not buying the EU's proposal which subverts encryption
https://www.politico.eu/article/germany-eu-damage-control-encryption-abuse-online/

Politico's headline was: "Germany forces EU into damage control over encryption fears", adding that "Berlin has criticized the EU's plan to fight child sexual abuse material as a threat to privacy and fundamental rights."

As we covered less than a month ago, the European Commission proposed a new law to crack down on sexual abuse of children online. That proposal is clearly facing some serious headwind from the bloc's largest member country. Since the proposal was first aired, the German government has repeatedly slammed the proposed legislation as an attack on privacy and fundamental rights until finally, last week, Germany's digital minister Volker Wissing warned that the draft law "crosses a line."

Berlin's opposition prompted the EU's Home Affairs Commissioner to step in last week in an effort to limit the damage with some appeasement. The EU's Johansson defended the proposal during an impromptu press conference in Luxembourg on Friday afternoon, saying that the legislation *"is much more targeted"* than the current regime to scan for illegal images and *"will allow only companies to do detection after a court decision or another independent authority have decided so after consultation with data protection authorities and with specific technologies that have been approved."*

Okay, wait, what? Those were direct quotes. So the legislation *"will allow only companies to do detection after a court decision or another independent authority have decided so after*

consultation with data protection authorities and with specific technologies that have been approved." Whoever wrote that is apparently describing some very different legislation than the text we examined a month ago. The intent of that legislation was quite clear. As was the breach of end-to-end encryption that would be needed to make it work.

And this is devolving into the sort of mess that it was bound to. Sweden's EU commissioner spoke alongside Germany's Interior Minister Nancy Faeser, insisting upon Faeser's "strong support" as a mother, an adult and a politician. And for her part, Faeser said that "the initiative, from the German point of view, we support this," but added that "for us, it's important to find the balance" between the right to confidential communication and cracking down on child sexual abuse material — CSAM. But it's just not possible to provide both at the same time. It's not.

To remind everyone, as it was presented last month, the revised rulebook wants to force all tech companies — including messaging apps like Whatsapp, Apple's iMessage, Instagram and Telegram — to scan, remove and report illegal photos and videos of SCAM material. Courts could also order digital companies to hunt down manipulative conversations between potential sex offenders and children, known as grooming.

No one wants our technology to be used for criminal child sexual abuse. But there's no way around the fact that examining photo and video content, and scanning and interpreting text messaging proactively looking for suggestive "grooming language patterns" can only be accomplished by defeating both the spirit and the act of end-to-end communications privacy.

So, while welcoming stronger action to protect victims of online abuse, a swath of German government ministers has nevertheless piled on to lament that the European Commission's proposal would effectively result in mass surveillance of people's private messages and thus undermine encryption. One vocal German minister said that the solution to protect kids wasn't to "check every private message." Unfortunately, if you want to actually protect kids from unknown predators, that's exactly what needs to be done, and that's what the legislation proposes to do.

We know that crypto technology is obedient. It will do anything we want. But what we apparently want is to invade everyone's privacy on the off-chance of detecting that their conduct might be criminal, while at the same time invading no one's privacy. Good luck with that.

The Conti Gang have finally pulled the last plug

Recall that we previously noted the clear indications that the too prolific for it's own good Conti ransomware gang had boxed itself into a corner by clearly and forcibly siding with Russia in their very unpopular unprovoked war against Ukraine. The sanctions that the West and most of the rest of the free world leveled against Russia choked off Conti's victim's ability to pay ransoms, even if they wished to.

So, Conti set up a shell game. All of the members of Conti abandoned ship except for one last member who remained behind to keep the fires lit. This member continued leaking data and taunting Costa Rica to create a facade of a running operation while all of the rest of Conti's members quietly moved to other ransomware gangs.

Last month's report by Advanced Intel stated that the only goal Conti had for this final attack was to use the platform as a tool for publicity, to keep the spotlight on Conti as they performed and faked their own death, keeping their multiple rebirths off of anyone's radar. Even though they were pretending to still be active, the ransomware operation was not performing any further attacks, and the data being leaked by this remaining Conti member was from earlier attacks. To confuse researchers and law enforcement even more, this Conti member released the same victim's data on both their site and Hive's data leak site, where he is also an affiliate. But this was all a charade with the rest of the Conti ransomware crew infiltrating or even taking over other ransomware operations.

So now, finally, the masquerade is over and the Conti ransomware operation has finally shut down its last public-facing infrastructure, consisting of two Tor servers used to leak data and negotiate with victims, of which none remain.

So the Conti name has been retired, but its crew have spread out and around and have taken up operations under many other names.

Log4J and Log4Shell is alive and well

Last Thursday, CISA posted a useful reminder report that I thought was interesting on several levels. Its title was: *"Malicious Cyber Actors Continue to Exploit Log4Shell in VMware Horizon Systems"*

One of the reasons I thought this was important is because it's easy for us to become complacent. For a few weeks late last year, Log4J and Log4Shell were big news and everyone was scurrying around fearing that this might result in the meltdown of the Internet, or another big worm, or the end of civilization as we know it. Okay, probably no one seriously believed that. But it was initially grabbing all of the tech press headlines. And it certainly had our attention on this podcast.

Then we learned an important lesson: because it did not enable super-simple drop and go exploitation, despite the fact that the presence of the Log4J vulnerability remained quite widespread — remember that we talked about how long it would take for all of the JAVA packages that were dependent upon it to be updated, if ever — the mass of the world's script kiddies did not pick it up and run with it. Potentially widespread though it was, actually exploiting the Log4J vulnerability required significant per-instance work. It wasn't the lowest hanging fruit, so lower hanging fruit continued to be preferentially exploited.

But, as we also said several months ago when the Internet was still here, neither did the Log4J threat go away. Log4J would be quietly added to the toolkits of the world's sophisticated threat actors for judicious deployment when and where its presence had been overlooked on a system where it might offer a way in. And that's what CISA's posting last Thursday was intended to remind us of, and to document.

CISA wrote:

The Cybersecurity and Infrastructure Security Agency (CISA) and United States Coast Guard Cyber Command (CGCYBER) are releasing this joint Cybersecurity Advisory (CSA) to warn network defenders that cyber threat actors, including state-sponsored advanced persistent threat (APT) actors, have continued to exploit CVE-2021-44228 (Log4Shell) in VMware Horizon® and Unified Access Gateway (UAG) servers to obtain initial access to organizations that did not apply available patches or workarounds.

Since December 2021, multiple threat actor groups have exploited Log4Shell on unpatched, public-facing VMware Horizon and Unified Access Gateway servers. As part of this exploitation, suspected APT actors implanted loader malware on compromised systems with embedded executables enabling remote command and control (C2). In one confirmed compromise, these APT actors were able to move laterally inside the network, gain access to a disaster recovery network, and collect and exfiltrate sensitive data.

This Cybersecurity Advisory provides the suspected APT actors' tactics, techniques, and procedures (TTPs), information on the loader malware, and indicators of compromise (IOCs). The information is derived from two related incident response engagements and malware analysis of samples discovered on the victims' networks.

I have a link in the show notes to the entire Advisory for anyone who wants more detail. But the moral of our story is an important reminder, that exploits which fall off the radar don't also cease to exist. Just because old problems are no longer being actively discussed should not be any source of comfort for anyone: https://www.cisa.gov/uscert/ncas/alerts/aa22-174a

The '311' emergency number proposal

Last Wednesday, in their third such meeting, a group of cybersecurity company experts met in Austin, Texas to provide industry recommendations to CISA. The group was founded in June of last year and held its first meeting last December. The group is split into six subcommittees each focused upon different issues which cover cyber workforce, information dissemination, cyber hygiene (whatever that is), technical advisories, critical infrastructure and misinformation.

The cyber hygiene subcommittee, led by Apple's vice president of corporate information security suggested that CISA *"launch a '311' national campaign, to provide an emergency call line and clinics for assistance following cyber incidents for small and medium businesses."* The measure was also floated by the communications subcommittee, led by a member of Tenable's board.

A spokesperson for Check Point Software said that the idea for a "311" emergency line is "smart and timely." The Check Point executive noted: *"Right now, we're seeing on average organizations in the United States being attacked 868 times per week. The emergency line can make for a faster path towards incident response."* During this month of June, Check Point said that they were seeing an average of more at least 27 cyberattacks against small and medium size businesses each week, and that this was an increase of 72% compared to last year.

Meanwhile, CISA executives and others continue to push for more robust and reliable incident reporting. As we covered at the time, cyber incident reporting legislation was passed and signed

into law earlier this year, but it only covers critical infrastructure organizations. All qualifying organizations are now required by law to report breaches to CISA within 72 hours and report ransomware payments within 24 hours.

Three weeks ago, during the annual RSA security conference, Eric Goldstein, CISA's executive assistant director for cybersecurity spoke at length about how damaging the lack of data on ransomware attacks in the U.S. is for organizations like his. (I assume that more attacks means more funding, though CISA has recently been getting a great deal of funding.) Eric told attendees that: *"Only a small fraction of ransomware victims are reported to the government, and the problem is getting worse. We have no idea what that actual number is. We have no idea what level of ransomware instructions are occurring across the country on any given day."*

I hadn't really stopped to think about it, but that would really be true, right? I mean, if you aren't required by law to report a breach of your organization, and if you don't have cyberattack insurance, as I would imagine most small to medium size businesses do not, so you're able to keep the fact of an attack closely held, then why would small businesses do anything more than arrange payment and hope to obtain a recovery key?

I can see how that lack of information flow to CISA would be frustrating. But, even so, it's unclear what good calling '311' would do for someone. It's not as if the government is going to pay the ransom. And you can imagine that the bad guys are going to take the standard action of saying *"don't get law enforcement involved or you'll never see your data again!"* So I suppose it's unsurprising that small to medium size organizations that are attacked are not reaching out.

56 Insecure-By-Design Vulnerabilities

Some reports are just demoralizing and depressing. One such, is a recently published report by Forescout which details a collection of 56 vulnerabilities which they have found in the so-called "OT" or Operational Technology category of devices. OT is the newer term which we once used for SCADA systems. So they're the technologies used to monitor and control oil and gas, chemical and nuclear power generation and distribution, manufacturing, water treatment and distribution, mining, and building automation. What's interesting is that these products are sold under the rubric "secure by design" and even carry certifications for OT security standards. But Forescout's report was titled: "OT:ICEFALL - A Decade of **Insecure**-by-Design Practices in OT"

The vulnerabilities affect Siemens, Motorola, Honeywell, Yokogawa, ProConOS, Emerson, Phoenix Contract, Bentley Nevada, Omron and JTEKT. And the disclosure of this mess was coordinated with CISA and other relevant government agencies around the world. Summarizing what they found, the vulnerabilities could be divided into four categories:

1. Insecure engineering protocols,
2. Weak cryptography or broken authentication schemes
3. Insecure firmware updates, and
4. Remote code execution via native functionality.

38% of the 56 vulnerabilities allowed for compromise of credentials,
21% allowed for firmware manipulation, and
14% allowed remote code execution.

And despite that, three quarters of the affected product families carry some form of feel good, yet apparently worthless, security certification. Forescout explained their intentions and why they felt this work was important by writing:

> *"With OT:ICEFALL, we wanted to disclose and provide a quantitative overview of Operational Technology insecure-by-design vulnerabilities rather than rely on the periodic bursts of CVEs for a single product or a small set of public, real-world incidents that are often brushed off as a particular vendor or asset owner being at fault. These issues range from persistent insecure-by-design practices in security-certified products, to subpar attempts to move away from them. The goal is to illustrate how the opaque and proprietary nature of these systems, the suboptimal vulnerability management surrounding them and the often-false sense of security offered by certifications significantly complicate OT risk management efforts."*

This report highlights another of the recurring themes of this podcast: Something is very wrong with the model we currently have for proprietary technology being blindly applied in critical infrastructure such as power generation and water treatment without any truly effective oversight over the security of these proprietary systems. We need to move this away from a model and a culture where the security status of products is allowed to be obscured by their proprietary intellectual content. This is the classic "Voting Machine" problem. Until that happens our infrastructure is going to be fragile and at risk.

Miscellany

"Long Story Short"

This is totally random and off topic. Okay, so it has a bit of time travel, but it's not Sci-Fi. It's a low-budget, independent, 90-minute, well-written romantic comedy on Netflix called "Long Story Short." It was released a year ago on July 2nd. It isn't any sort of blockbuster and it only scored 6.5 on IMDB, which is weird because all of the reviews I saw came up as 8's, 9's and 10's. In any event, this little gem carries and beautifully delivers an important message about the conduct of our lives, and not in a preachy way.

So, I offer no guarantee since I know that people's tastes vary widely. But I know that **I** would have been glad to have been told about it. So I didn't want to pass up the opportunity to share this little discovery. Again, it's "Long Story Short" on Netflix. And if you have someone to watch it with, so much the better. :)

And we have TEN pieces of interesting feedback from our amazing listeners:

Closing The Loop

Lawn_dart / @lawn_dart

Just listened to this week's SN and the encryption problem still isn't quite complete. If an attacker knows a significant amount about the plain text then encrypting plain text guesses might be more efficient. The puzzle only truly works if the plain text is also indistinguishable from entropy.

Tiemo Kieft @TiemoKieft

Routers are IOT devices, they interact with the physical world through LEDs. Either that or smart light bulbs are not IOT either.

Arvind Narayanan / @random_walker

I mis-clicked on one of my 150 open tabs and it happened to be a tab that's been open since 2019 with a paper that has a solution to the exact research problem I've been puzzling over today. This is the moment I've been waiting for and I've decided to never close any tabs again.

LDizzy / @Ldizzy

Hello Steve. I've been listening to a lot of the talks lately about passkeys and how it locks you into the ecosystem you start with more or less. But I think there's a big chunk we're missing here. While the public/private key pair does function like a much more sophisticated password, where things will get better is sites can accept more than one passkey. Instead of having a single password that's reset via email when lost, sites will most likely allow for the addition of several passkeys. If I am an iOS mobile user that also runs Windows on my desktops (which I am), then I could add an iOS-based passkey on my phone that's synced via iCloud. Then, whenever I get around to logging in to the site from my computer, I could use my iPhone to authenticate and add a second passkey from the desktop that would then by synced within Microsoft's ecosystem. This makes sense to me and seems like a new paradigm. There's no real reason for sites not to allow multiple passkeys; although this wouldn't have been logical with the old username/password combo. The downside is that IF you use multiple ecosystems then you'd have to add a passkey within each. BUT they don't have to be synced between ecosystems and it's a one-time thing for each. I doubt many people use more than 2. I hope I'm not being overly optimistic and I'd appreciate your take on this. Love your work.

Joseph Fienberg / @fienberg

My wife and I each received an email about the Facebook tracking class action settlement. When I went to click on the link to file the class action claim, Ublock origin blocked lzzgcc5d.r.us-east-1.awstrack.me which is not secure and is a tracking link for a tracking settlement where we'll each probably get 50¢. I thought Security Now listeners would get a laugh out of the irony of this.

Thomas Martin / @toskp10

Hi Steve! I wanted to weigh in on your solution to the double encryption dilemma in SN-876. Your position was that an attacker cannot know the intermediate text, and therefore the two encryptions cannot be attacked separately. The conclusion was then that double encryption was substantially more secure than single encryption: if you can brute force a single 256-bit cipher (e.g. AES-256) in one day, then it does not take 2 days to brute force double AES-256, it would take (on the order of) 2^256 days.

Unfortunately, this line of reasoning is incorrect as it does not take into consideration another type of attack: a meet in the middle attack. The attacker needs one known ciphertext-plaintext pair. They encrypt the plaintext with every possible key and store the (key, text) pairs in a lookup table (requires O(2^256) operations and O(2^256) storage). They also decrypt the ciphertext with every possible key. Each resulting text is checked against the lookup table. If it matches a stored text, then the attacker has a pair of keys that decrypt the known ciphertext to the known plaintext. This would need to be confirmed with other plaintext/ciphertext pairs as there is the possibility of a coincidental match.

Douglas Nichols / @TheNickleMan

Double encrypting can be divide and conquer if you are using an authenticating encryption like aes gcm or ccm modes on at least the outer encryption since the integrity check will fail.

Sean OBrien / @sobrien60

Many years ago you gave almost the opposite answer to the divide and conquer attack. Back then you said the encryptions add because of authentication. If the 1st encryption is signed then the 1st decryption is obvious. It's in the case without authentication that the 1st decryption results in noise.

Rando / @therandomviking

A better proof that we live in a simulation is murphy's law. Because statistically it is unlikely for everything possible to go wrong all the time without exception, but in my experience that is exactly what happens.

Barnacules Nerdgasm / @Barnacules

Hey Steve, it was really cool hearing you talk about my Microsoft career as a Senior SDET. I love sharing my experiences with the WHQL & ESC teams responsible for testing WINMAIN branch & talking about why the current system post 2015 layoffs is so bad!

The “Hertzbleed” Attack

It has a snappy name, its own website and a memorable logo. Everything the modern vulnerability needs. And its name is an obvious play on 2014’s “HeartBleed” attack, for good reason. Eight years ago, “HeartBleed” was one of those rare vulnerabilities, like Dan Kaminsky’s realization of DNS spoofing vulnerabilities, that truly moved the industry to action. It was CVE-2014-0160 — Ah, those good old days of 4-digit CVE’s. HeartBleed’s summary reminds us:

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

We have tested some of our own services from attacker's perspective. We attacked ourselves from outside, without leaving a trace. Without using any privileged information or credentials we were able to steal from ourselves the secret keys used for X.509 certificates, usernames and passwords, instant messages, emails, business critical documents and communications.

That was eight year ago with HeartBleed. Today we have HertzBleed:

https://www.hertzbleed.com/ https://www.hertzbleed.com/hertzbleed.pdf

Here’s how today’s HertzBleed describes itself:

*Hertzbleed is a new family of side-channel attacks: **frequency side channels**. In the worst case, these attacks can allow an attacker to extract cryptographic keys from remote servers that were previously believed to be secure.*

Hertzbleed takes advantage of our experiments showing that, under certain circumstances, the dynamic frequency scaling of modern x86 processors depends on the data being processed. This means that, on modern processors, the same program can run at a different CPU frequency (and therefore take a different wall time) when computing, for example, 2022 + 23823 compared to 2022 + 24436.

Hertzbleed is a real and practical threat to the security of cryptographic software. We have demonstrated how a clever attacker can use a novel chosen-ciphertext attack against SIKE to perform full key extraction via remote timing, despite SIKE being implemented as “constant time”.

"SIKE" stands for "Supersingular Isogeny Key Encapsulation" and all we really need to know is that it's some state–of-the-art post-quantum crypto. But what it is is not really that important. The point is that regardless of how secure something is, if its keys can be extracted, the game is over.

I should say a little bit about power consumption in modern state-of-the-art semiconductor systems. Today's CPUs still use transistors. But whereas the first transistors invented were current amplifiers, known as bipolar transistors, today's transistors use electrostatic charge rather than active current to open and close their switches. They are metal oxide semiconductor field effect transistors. That's M. O. S. F. E. T. or MOSFET.

If a MOSFET transistor is either on or off, no power is consumed. In order to "flip the switch" the controlling gate of a field effect transistor must be charged up with electrons, or drained of its electrons. But once that's done, the gate will remain charged up or drained and the switch will remain open or closed without consuming any power.

So, the key thing to appreciate is that in a large array of interconnected MOSFET transistor switches — like any modern CPU — power is only consumed when the states of those MOSFET switches are changed. And the amount of power consumed is proportional to how many switches are changed and how often they are changed.

This explains why we can hear our laptop fans spin up when our machines get busy and why they eventually spin back down some time after they have been idle. When a CPU is idling it is doing less work because it has less work to do. Since switching transistors on and off is what consumes power, batteries can be made to last longer and systems can operate more "green" by slowing down the clock speed of CPUs which don't have much work to do. A reduced clock speed means fewer transistors switching per second, which means less power needed and consumed, which means less power lost as heat, and fans don't need to spin as fast since there's less heat needing to be removed from the CPU.

The dynamic speed scaling of today's CPUs has become an art form, with the CPU tightly and instantly changing its speed based upon the instantaneous demands placed upon it. And if you have guessed that this dynamic speed changing represents a new form of side-channel information leakage, you get an "A" and move to the head of the class.

Looking again at that the researcher's summary:

> *Hertzbleed takes advantage of our experiments showing that, under certain circumstances, the dynamic frequency scaling of modern x86 processors depends on the data being processed. This means that on modern processors the same program can run at a different CPU frequency (and therefore take a different wall time) when computing, for example, 2022 + 23823 compared to 2022 + 24436.*

This work was done through the collaboration of a team of six researchers from Universities in Illinois, Texas and Washington. Their paper, titled *"Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86"* will will appear in the 31st USENIX Security Symposium being held in Boston from August 10th through the 12th this summer.

So what does this mean for us?

Am I affected by Hertzbleed?

Likely, yes. Intel's security advisory states that all Intel processors are affected. We experimentally confirmed that several Intel processors are affected, including desktop and laptop models from the 8th to the 11th generation Core microarchitecture.

AMD's security advisory states that several of their desktop, mobile and server processors are affected. We experimentally confirmed that AMD Ryzen processors are affected, including desktop and laptop models from the Zen 2 and Zen 3 microarchitectures.

Other processor vendors (e.g., ARM) also implement frequency scaling in their products and were made aware of Hertzbleed. However, we have not confirmed if they are, or are not, affected by Hertzbleed.

What is the impact of Hertzbleed?

First, Hertzbleed shows that on modern x86 CPUs, power side-channel attacks can be turned into (even remote!) timing attacks—lifting the need for any power measurement interface. The cause is that, under certain circumstances, periodic CPU frequency adjustments depend on the current CPU power consumption, and these adjustments directly translate to execution time differences (as 1 hertz = 1 cycle per second).

Second, Hertzbleed shows that, even when implemented correctly as constant time, cryptographic code can still leak via remote timing analysis. The result is that current industry guidelines for how to write constant-time code (such as Intel's) are insufficient to guarantee constant-time execution on modern processors.

This is the brilliance of what these guys have done.

We talked a long time ago about the need for constant-time execution of cryptographic algorithms. Or being a bit more specific, never have any secrets — like the cipher's key material — directly controlling the execution path. As we know, in modern processors, all execution paths leave bread crumb trails in the underlying microarchitecture. Things like the processor modifying its future branch predictions based upon the past.

So, constant-time code is carefully designed to take the same path and to execute in the same number of CPU cycles specifically so as to give attackers who may be watching carefully from the outside, what specific data was being processed.

What these guys realized was that there's another form of information leakage occurring from Intel's x86 and AMD Ryzen processors: Even though the number of CPU cycles may be constant and the execution path may be invariant, the exact number of INDIVIDUAL TRANSISTORS whose on and off states were changed during the computations will almost necessarily differ depending upon the data that was processed, and since CPUs dynamically scale their speed, the ACTUAL time required to perform the computation — not in CPU cycles but in actual passage of world

time — will be subtly altered... and that this can leak secret key information.

Should I be worried?

If you are an ordinary user and not a cryptography engineer, probably not: you don't need to apply a patch or change any configurations right now. If you are a cryptography engineer, read on. Also, if you are running a SIKE decapsulation server, make sure to deploy the mitigation described below.

Is there an assigned CVE for Hertzbleed?

Yes. Hertzbleed is tracked under CVE-2022-23823 and CVE-2022-24436 in the Common Vulnerabilities and Exposures (CVE) system.

Is Hertzbleed a bug?

No. The root cause of Hertzbleed is dynamic frequency scaling, a feature of modern processors, used to reduce power consumption (during low CPU loads) and to ensure that the system stays below power and thermal limits (during high CPU loads).

When did you disclose Hertzbleed?

We disclosed our findings, together with proof-of-concept code, to Intel, Cloudflare and Microsoft in Q3 2021 and to AMD in Q1 2022. Intel originally requested our findings be held under embargo until May 10, 2022. Later, Intel requested a significant extension of that embargo, and we coordinated with them on publicly disclosing our findings on June 14, 2022.

Do Intel and AMD plan to release microcode patches to mitigate Hertzbleed?

No. To our knowledge, Intel and AMD do not plan to deploy any microcode patches to mitigate Hertzbleed. However, Intel provides guidance to mitigate Hertzbleed in software. Cryptographic developers may choose to follow Intel's guidance to harden their libraries and applications against Hertzbleed. For more information, we refer to the official security advisories (Intel and AMD).

Why did Intel ask for a long embargo, considering they are not deploying patches?

Ask Intel.

Is there a workaround?

Technically, yes. However, it has a significant system-wide performance impact.

In most cases, a workload-independent workaround to mitigate Hertzbleed is to disable frequency boost. Intel calls this feature "Turbo Boost", and AMD calls it "Turbo Core" or "Precision Boost". Disabling frequency boost can be done either through the BIOS or at runtime via the frequency scaling driver. In our experiments, when frequency boost was disabled, the frequency stayed fixed at the base frequency during workload execution, preventing leakage via Hertzbleed. However, this is not a recommended mitigation strategy as it will significantly impact performance. Moreover, on some custom system configurations (with reduced power limits), data-dependent frequency updates may occur even when frequency boost is disabled.

In other words, as with Spectre and Meltdown, this is another instance where a CPU optimization must be discarded if its exploitation is to be completely eliminated. Nothing is safe.

What is SIKE?

SIKE (Supersingular Isogeny Key Encapsulation) is a decade old, widely studied key encapsulation mechanism. It is currently a finalist in NIST's Post-Quantum Cryptography competition. It has multiple industrial implementations and was the subject of an in-the-wild deployment experiment. Among its claimed advantages are a "well-understood" side channel posture. You can find author names, implementations, talks, studies, articles, security analyses and more about SIKE on its official website.

What is a key encapsulation mechanism?

A key encapsulation mechanism is a protocol used to securely exchange a symmetric key using asymmetric (public-key) cryptography.

Is my constant-time cryptographic library affected?

Affected? Likely yes. Vulnerable? Maybe.

Your constant-time cryptographic library might be vulnerable if is susceptible to secret-dependent power leakage, and this leakage extends to enough operations to induce secret-dependent changes in CPU frequency. Future work is needed to systematically study what cryptosystems can be exploited via the new Hertzbleed side channel.

Did you release the source code of the Hertzbleed attack?

Yes, for full reproducibility. You can find the source code of all the experiments from our paper at the link: https://github.com/FPSG-UIUC/hertzbleed