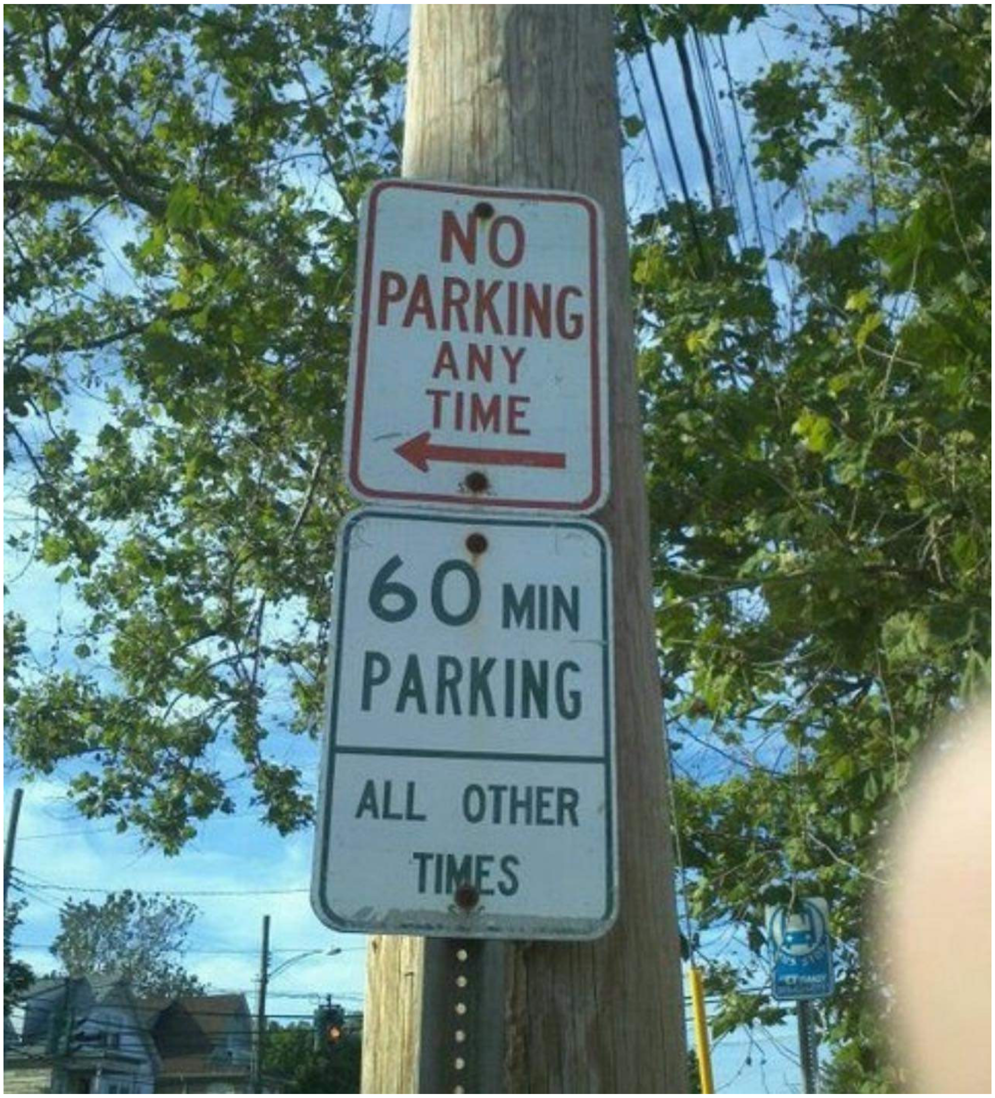
Security Now! #947 - 11-07-23

Article 45

This week on Security Now!

Where was Microsoft storing their Azure keys? What four new 0-day flaws has Microsoft declined to repair? and what happens next? What's this week's latest mass-casualty event for publicly-exposed Internet servers? And do we have any news on last week's Citrix Bleed fiasco? What comes after CVSSv3.1 and why? What happened to Google's WebDRM proposal? And what about the earlier Cisco IOS XE mass-casualty mess? And what's the new Security Now! podcast slogan to emerge from it? Our favorite password manager just announced their support for Passkeys! Now what? That guy with the badly messed-up SSD shared the results of using SpinRite 6.1. I'll share and explain what happened. And then, after entertaining some great feedback from our listeners, we're going to look into the next big looming battle between conservative tech and rapacious governments. All that and more during this week's Security Now! podcast #947 ... and counting.

To park, or not to park? THAT is the question.



Security News

Microsoft announced their move to an HSM

Last Thursday, Microsoft posted under the headline “Announcing Microsoft Secure Future Initiative to advance security engineering” which was written by Charlie Bell, Executive Vice President, for Microsoft Security. He opened this posting with this introduction:

> *Today Microsoft’s Vice Chair and President Brad Smith shared insight on the global cybersecurity landscape and introduced our Secure Future Initiative. These engineering advances anticipate future cyberthreats, such as increasing digital attacks on identity systems. They also address how we will continue to build secure foundations necessary for the AI era and beyond. In the spirit of transparency and to emphasize the importance of this moment, we are sharing the internal email sent earlier about our Secure Future Initiative’s strategy and objectives.*

https://www.microsoft.com/en-us/security/blog/2023/11/02/announcing-microsoft-secure-future-initiative-to-advance-security-engineering/

Mostly it’s just a marketing piece “blah blah blah – we’re leading the way toward a more secure future, improving the lives of our customers in the face of rapidly growing cyber threats – blah blah blah”. I have a link to the piece in the show notes just for the record in case anyone’s curious. And I normally wouldn’t have given this a second thought nor be mentioning it here, except that about 80% of the way down something appeared that did seem worth sharing. They wrote, as part of their “Secure Future Initiative'':

> *To stay ahead of bad actors, we are moving identity signing keys to an integrated, hardened Azure HSM* [you know, a Hardware Security Module] *& confidential computing infrastructure. In this architecture, signing keys are not only encrypted at rest and in transit, but also during computational processes as well.* [Ah! What a concept!] *Key rotation will also be automated allowing high-frequency key replacement with no potential for human access, whatsoever.*

So, in short, Microsoft will be leading the way into a Secure Future by working to catch up with what everyone else who cares about security has been doing all along for many years already. As I observed a couple of months ago, the **only possible way** they could have lost control of that private signing key during a system crash and subsequent RAM snapshot, was if that private key was in RAM at the time of the crash. And the only way it could have **ever** been in RAM was if signing was being done outside of an HSM.

During last week’s “Citrix Bleed” podcast we examined a crisp and clear example of a bug which allowed for the exfiltration of RAM. And, as it happened, that RAM contained active and valid authentication tokens. We don’t know whether they needed to be lying around in RAM. It’s quite possible that they did need to be there in order to remain valid. But it’s very common to make the mistake of leaving sensitive information lying around even after it’s no longer needed. The problem is that our current programming languages are still not secure by default or design; so they must be made secure by deliberate act.

I also noted last week that while I was coding the SQRL client I was in a more or less constant state of terror. And I would submit that that's the state you want coders of secure systems to be in.

There's a great and useful concept and phrase: It's the notion of "multi-layer security." The idea is that there is no single point of failure that would result in a security compromise. In order for security to be compromised, many things would need to go wrong all at once. In the case of Citrix Bleed, if the system's RAM could have been swept clean of valid tokens – and we don't know whether or not that would have been feasible – but if it could have been, then even in the face of that very clear coding error, valid tokens would not have been available for exfiltration. In other words, wipe RAM not because you know you need to, but because doing so would add an additional layer of security. And additional layers of security are never a bad thing for a secure system to have.

In that almost hard to believe case of Microsoft's loss of their private signing key, they explained that they **did** already have multiple layers of security in place – like, 5 of them! – yet in a bizarre and quite unlikely seeming chain of failures, every one of those layers failed when it was actually needed. If they had the **one** additional layer of using a Hardware Security Module to their system, which they're boasting about doing now, then none of those high-value governmental eMail accounts would have been breached as a result of the failure of every other layer of security.

So our takeaway lesson here is, it's not possible to have too many layers of security. You never know which one will be the layer that stops the bad guys.

A Quartet of new 0-days in Exchange Server (that Microsoft won't fix.)

Last Friday, the day following Microsoft's big "Secure Future Initiative to advance security engineering" announcement, we learn from BleepingComputer that Trend Micro's Zero-Day Initiative informed Microsoft on September 7th and 8th of four new 0-day vulnerabilities they had discovered in Exchange Server, one which allowed remote code execution. Microsoft acknowledged the reports, but decided that the flaws were not severe enough to warrant immediate attention and decided to put off the fixes until some later unspecified date. In other words "thank you, now go away." Since ZDI strongly disagreed with this response, they decided to publish the rough descriptions of the four vulnerabilities under their own tracking IDs in order to at least warn Exchange admins about the security risks – even though there's not much for Exchange admins to do at this point.

Now, if something about this overall scenario seems familiar, where security researchers inform Microsoft of flaws they have found somewhere, which they believe are important, and after presumably examining those reports Microsoft decides that the problem is not worthy of their attention... you would be correct. We've been right here before and if history continues to repeat itself, we also know what lies ahead: Microsoft will leave this unpatched, some bad guy somewhere will pick up on the possibility of an outstanding remote code execution vulnerability in Exchange Server, and will go "a hunting." Some time later, Exchange Servers will start being compromised in some mysterious new way that no one ever saw before – except that, whoops, Trend Micro and Microsoft both saw it in September of 2023 and one of the two of them who could have done something to prevent it chose not to. Like I said, we've seen this before, and it's a shame.

> *ZDI-23-1578 is a remote code execution (RCE) flaw in the 'ChainedSerializationBinder' class, where user data isn't adequately validated, allowing attackers to deserialize untrusted data. Successful exploitation enables an attacker to execute arbitrary code as 'SYSTEM,' the highest level of privileges on Windows.*

So now our would-be bad guy also knows right where to look in Exchange Server. And surprise surprise, the problem is in deserialization. We've talked several times about the inherent difficulty of deserializing data. The process of "serializing" data takes some sort of formatted data structure and turns it into a "blob" for storage or transmission. The reverse process of "deserializing" that blob requires – wait for it... the interpretation of the data that the serializer produced. So we have some flaw in an interpreter in the ChainedSerializationBinder. Probably wouldn't be too difficult to find.

Just for the record, though they are not remote code execution flaws, the remaining three of the four are:

> *ZDI-23-1579 – Located in the 'DownloadDataFromUri' method, this flaw is due to insufficient validation of a URI before resource access. Attackers can exploit it to access sensitive information from Exchange servers.*
>
> *ZDI-23-1580 – This vulnerability, in the 'DownloadDataFromOfficeMarketPlace' method, also stems from improper URI validation, potentially leading to unauthorized information disclosure.*
>
> *ZDI-23-1581 – Present in the CreateAttachmentFromUri method, this flaw resembles the previous bugs with inadequate URI validation, again, risking sensitive data exposure.*

So they allow for some sort of unspecified information disclosure. While it's not running the attacker's code, information leaks can still be very valuable to attackers as part of a larger campaign.

The mitigating factor behind all four of these vulnerabilities is that they all require authentication. This may be the basis for Microsoft's dismissal of this as anything to worry about. But cybercriminals have repeatedly shown that there are many ways to obtain Exchange credentials, including brute-forcing weak passwords, phishing attacks, purchasing them, or acquiring them from info-stealer logs. So once the bugs are found, the need for a credential for a specific Exchange Server might not pose an insurmountable problem.

Trend Micro's Zero-Day Initiative folks said that the only salient mitigation strategy would be to restrict interaction with Exchange apps. But many businesses and organizations cannot operate without access to Exchange apps.

So, let's see whether we're returning to this in the future. Will history repeat itself?

Apache ActiveMQ Server Attacks

Apache ActiveMQ servers have been having trouble recently, too. And the attacks are neither theoretical nor latent. Apache ActiveMQ Server is a standalone message broker server which facilitates reliable high-available messaging among clusters of computers. It's written in Java and has been around and evolving since 2004.

The flaw, tracked as CVE-2023-46604, is a maximum severity bug in the ActiveMQ scalable open-source message broker that enables unauthenticated attackers to execute arbitrary shell commands on vulnerable servers.

It's unclear when the Apache Foundation became aware of the attacks on ActiveMQ, but two security firms, ArcticWolf and Huntress Labs, found that threat actors had been exploiting the flaw as a 0-day to deploy SparkRAT malware since at least October 10th. Apache released security updates to fix the vulnerability seventeen days later, on October 27. But we know that a patch being available and patches being applied are two very different things.

In addition to the deployment of the SparkRAT, remote access Trojan, ActiveMQ servers exposed to the Internet are being targeted in "HelloKitty" and "TellYouThePass" ransomware attacks. Those two pieces of ransomware share a common infrastructure, eMail addresses, cryptocurrency addresses, and so forth.

And in terms of its spread, data from the threat monitoring service ShadowServer found that there are currently more than 9,200 Apache ActiveMQ servers exposed online, with over 4,770 – so more than half – currently vulnerable to exploitation. If the good guys can scan the public Internet to obtain a count of victims, bad guys can scan for potential victims to target. Needless to say, if your organization is using an Apache ActiveMQ Server with Internet exposure you'll want to update it immediately and look around carefully for any indication that the bad guys might have already entered your network.

Citrix Bleed Update

Given the sweeping scope of the mess with Citrix Bleed, I spent some time looking for any update since last week's Halloween status. The team at Mandiant, which is now a Google property, appears to be more on the ball and current than everyone else who're just citing Mandiant's research. Their latest update from last Thursday adds some interesting new pieces of information and helps to bring home the reality of the situation that's facing those 20,000 plus Citrix users whose network appliances have been compromised. On Thursday, they wrote:

> *Mandiant has identified zero-day exploitation of this vulnerability in the wild beginning in late August 2023 as well as n-day exploitation after Citrix's publication. Mandiant is investigating multiple instances of successful exploitation of CVE-2023-4966 that resulted in the takeover of legitimate user sessions on NetScaler ADC and Gateway appliances. The session takeovers bypassed password and multi-factor authentication. In this blog post, we will discuss artifacts that can be used to identify exploitation activity and highlight some of the post exploitation techniques we observed during the incident response investigations.*

They lay out what's already known about the vulnerability of the Citrix endpoints, the challenges of investigating vulnerable devices because the webserver running on the appliance doesn't record requests (or errors) to the vulnerable endpoint. And they note that they're not aware of any configuration change that can be made to force request logging for these endpoints.

But what was most interesting, and the reason I wanted to share this update, was what they had to say about the post-exploitation activity they have observed. In other words, what are some of the things the bad guys do once they bypass the system's authentication and gain access? Mandiant explained:

> *Following the successful exploitation of CVE-2023-4966, Mandiant has observed a variety of post-exploitation tactics, techniques, and procedures (TTPs). Once an actor was able to successfully achieve session hijacking, the threat actor performed actions including host and network reconnaissance of the victim environment, credential harvesting, and lateral movement via RDP. Mandiant identified evidence of Active Directory reconnaissance using living-off-the-land binaries such as net.exe. Additionally, Mandiant has observed the use of the SoftPerfect network scanner (netscan.exe) to perform internal network enumeration.*
>
> *In several cases, the threat actor used 7-zip to create an encrypted segmented archive to compress the reconnaissance results. The threat actor then used the built-in `certutil` utility to Base64 encode the segments.*
>
> *In one case, `certutil` was used to decode multiple files related to credential theft. Mandiant observed the threat actor use `e.exe` to load `d.dll` into `lsass` process memory. When run, the utility creates a memory dump file located at `%temp%\1.png` and prints `success` to the console when done. The memory dump file can be processed offline by the threat actor to extract credentials. Mandiant identified `sh3.exe` as a utility suspected to run the Mimikatz `LSADUMP` command.*
>
> *In another instance, a threat actor used `certutil` to decode a file that Mandiant identified as a newly tracked backdoor that uses Slack as its command and control. Tracked by Mandiant as FREEFIRE, it is a lightweight backdoor written for .NET. FREEFIRE communicates to a hard-coded channel to retrieve commands and upload responses. It supports loading arbitrary .NET assemblies encoded as Base64 sent to it via chat comments. Mandiant observed FREEFIRE being deployed by a threat actor through the following certutil command.*
>
> *Mandiant has also observed the deployment of various remote monitoring and management tools following the successful exploitation of CVE-2023-4966. Currently, Mandiant has observed the deployment of Atera, AnyDesk, and SplashTop to establish and maintain a foothold following exploitation of CVE-2023-4966.*
>
> *Mandiant is investigating intrusions across multiple verticals, including legal and professional services, technology, and government organizations. Given the widespread adoption of Citrix in enterprises globally, we suspect the number of impacted organizations is far greater and in several sectors.*
>
> *Mandiant is currently tracking four distinct uncategorized groups involved in exploiting this vulnerability. We have observed some lower degrees of confidence overlaps in post-exploitation stages among these UNC groups, like using the same recon commands and utilities available on Windows.*

> *Two threat clusters used Mimikatz for dumping process memory. Notably, there were no overlaps in infrastructure between these clusters of activity. The exploits were sourced from different VPN provider IP addresses and previously compromised third-party devices.*

So, even though the attack is low complexity and is easy to launch, all indications are that well versed and competent threat actors are behind these. They are using tried and true post-exploitation tactics to obtain a high degree of leverage in and persistence on their victim's networks.

So we are now inhabiting a world where the moment a patch to fix a remotely exploitable flaw is announced, powerful malignant forces jump on the patch, determine what was changed, design an exploit for any not-yet-patched devices, then race to take advantage of the vulnerability, using it against anyone who didn't instantly patch their devices the moment the trouble and its fix were announced.

Asymmetric warfare is notoriously difficult to fight. And this currently-broken security model has asymmetric aspects. Consider that a small group of miscreants only need to watch for security updates from the major appliance vendors. Yet on the other side, every single person who is responsible for the operation of every deployed instance of every one of those devices spread anywhere in the world must be just as vigilant as the small team of bad guys. And on top of that, everyone everywhere must be ready to apply the fix at any time, night, day, weekend or holiday.

The only way I can see this evolving is for the high-end enterprise appliance world to make the same move that the small office / residential router world has made of allowing these devices to be remotely autonomously updated without the need for the device's IT personnel to be involved. This feature should be enabled by default with IT personnel having the option to disable it if they understand and accept the risks that accompany doing so.

UDP packets are small, connectionless and inexpensive to send. So every device that has not been disabled could send a packet periodically to the device's manufacturer to check for any updates. One tiny packet every ten minutes would be more than sufficient. In the event of a critical update, an affirmative UDP reply would contain the URL of the update to download and apply and the certificate of the remote web server could be pinned to prevent any forgery. The appliance would bring up an HTTPS connection to download the updated module, install it and reboot itself.

I'm aware of the many sound arguments against this sort of autonomous upgrading. Its first appearance in Windows caused quite a stir. My Unix servers send eMail to inform me of their packages that are in need of attention. This is information they're obtaining without any assistance from me, though they stop short of performing those changes autonomously. So while autonomous patching of enterprise class appliances may pose some risk, more than 20,000 users of this one device just had their networks deeply compromised because, for whatever reason, they did not install the patch that the bad guys were reverse engineering before that reverse engineering was turned into an active exploit. If they had, 20,000 individual network compromise disasters would have been averted. It seems to me that given the world we now live in, it's time to move autonomous patch updating from the consumer desktop and router – where

it's now been proven to be providing much more benefit than harm – to the enterprise's border equipment which are subject to swift attack, as we've all just seen.
And just for the sake of discussion, there are many possible compromise measures. For example, the periodic UDP packet sent by the device back to its manufacturer could contain the device's current build version and the latest current eMail address and cellphone number for the organization's IT cyber security team. That information could be configurable in the device's admin setup as an "in case of critical vulnerability send eMail and text to:" data. That way, every one of the manufacturer's devices is pinging homebase with the information needed to alert its administrators the instant any new and sufficiently urgent problem is discovered. It's difficult to believe that we're not already doing this as an industry. And while we're here talking about this, since it's foreseeable that the first thing a compromised device might do is shutdown that early warning update service, the device's manufacturer should have these periodic "pings" continually updating a database and also preventing malicious changes by retaining a history of previous contact information. In that way, the moment a serious problem was discovered every admin could be made aware that they'll need to prepare for an update. So I suppose my point is, we're really being lame about things today and there's really no excuse for it.

CVSSv4

We've just seen – and we continually see – the burden being placed upon front line IT personnel to keep their networks safe. If someone were to say that it's basically an impossible task, it would be difficult to mount a convincing counter argument. As always with security, the good guys must prevent intrusion everywhere, all the time, all at once (wasn't that the title of a recent movie?) But the bad guys only need to find one mistake anywhere one time. It's not fair.

We've heard that IT security guys are stressed and it's not surprising. I've mentioned this before but I'll say it again: The job might not be for everyone. But if it sounds like it's a fit with your personality, the good news is, the world is desperately looking for you. I saw a statistic recently indicating that there's about a 50% shortfall in IT security staffing. Something like 4 million empty job openings needing to be filled. One of the many things I've learned from our listeners is that they credit their listening to this podcast with giving them the inspiration to learn more about this subject which subsequently allowed them to move into the cybersecurity job market.

Okay, I just found the study. A quote from its summary is: *"The global cybersecurity workforce is estimated to have reached more than 5.5 million professionals and even though that number is 9% higher than last year, 4 million experts are still needed worldwide to fill open positions across the industry."* What I found is the 2023 "Cyber Workforce Study" in which a record 14,865 cybersecurity professionals were polled and share their unique perspectives on the state of the workforce. I'm sorely tempted to share more of it since it's chock full of interesting data and statistics. But I have a link to its 84-page PDF in today's show notes and since the URL is quite long, it's this week's (carefully numbered) GRC shortcut, so it's: https://grc.sc/947

https://media.isc2.org/-/media/Project/ISC2/Main/Media/documents/research/ISC2_Cybersecurity_Workforce_Study_2023.pdf

Okay, so where am I going with this? The job is chaotic, the challenge is highly asymmetric and unfair. So the overworked cybersecurity professional needs all the help they can get. In a world of software vulnerabilities how does one know how to start the day? One of the blessings this industry has created in an attempt to bring some form of order from this chaos is the Common Vulnerability Scoring System, which we are constantly referring to as the CVSS. The initial v1 of the CVSS was instituted in February of 2005, the same year we began talking about these interesting issues every week. A little over two years later, in the summer of 2007 it moved to v2 where it sat until v3 was introduced in 2015. That version lasted until 2019, when it was tweaked a bit to give us version 3.1, which we've been using until now.

The reason the CVSS system has needed periodic maintenance and updating is that we're not living in a static world. The drama we're seeing playing out this instant with the Citrix Bleed vulnerability did not exist to nearly this extent before cryptocurrency, because it was far less clear how attackers in Russia and North Korea could monetize their cyber intrusions. And vulnerable devices like Citrix's NetScaler hadn't yet been created. Since today's cyber landscape has changed, so must the metrics we use to characterize today's threats. To that end, work has been completed on the next generation of Common Vulnerability Scoring System and we are now at CVSS version 4. So what's changed? There are four primary highlights:

1. CVSS scoring metrics have been added and redefined to improve the granularity and clarity of CVSS scores. With the previous standard it was common to have different types of vulnerabilities clumped around the same score even though it didn't accurately reflect each one's severity. More scoring metrics in CVSSv4 means a better spread across the whole scale.

2. In keeping with today's world, there are now ICS, OT, and IoT-specific scoring metrics. This includes scoring metrics such as "Safety," "Automatable," and "Recovery," to let critical infrastructure operators know if a security flaw just looks bad on paper or if it's actually exploitable and dangerous to their networks.

3. We also have new scoring metrics such as "Value Density," "Vulnerability Response Effort," and "Provider Urgency." Those have been added to help responders evaluate and prioritize vulnerabilities. The last two, "Vulnerability Response Effort" and "Provider Urgency" are intended to allow vendors to tell customers that a vulnerability needs to be patched ASAP. This is a capability that was **not** present in the current CVSS.

4. And finally, CVSS version 3's "Temporal" metrics group has been replaced with a new group called "Threat Metrics." Although this replacement group is intended to reflect the same exploitability and proof-of-concept availability as its predecessor, its application is significantly clearer under version 4.

So, while we'll still be seeing and quoting the same single 1 to 10 CVSS composite score, that score will more accurately track the urgency presented by its vulnerability and the detailed breakdown of that single score will provide cyber security professionals much needed additional details and may help them to decide how to start their day.

Ace Hardware cyberattack
I have a soft spot in my heart for Ace Hardware. We have several of Ace's 5700 retail stores in our area, and they have a truly amazing array of random little hardware bits, nuts, bolts, washers and all sorts of weird one-off brackets. Many projects were saved when I needed a particular size bolt or washer. There are still some things that are difficult to do online. When you need to match a bolt to a nut, there's no substitute for being there. Anyway, I bring this up because nearly 200 of their servers and 1,000 other systems were hit by a cyberattack the day before Halloween, on Monday, October 30th. The attack impacted Ace's ability to pick up new customer orders. Other impacted systems include warehouse management systems, reward points program, tech support call center, and the company's mobile assistant. Despite the attack, the company's 5,700 stores have remained open, although with reduced activity. I haven't needed any bolts recently, but it's good to know that I can still get them there!

Google abandons "Web DRM"
Remember that bizarre plan Google floated a few months ago, which would give websites absolute control over the extensions and other features that could be used by anyone visiting a website that wished to impose such restrictions? The good news is that the plan is dead. And I'm impressed that they didn't make a larger push for something that really didn't seem to be in the end user's best interest. We talked about it a bit and the only upside I could see was that a banking site, for example, might wish to enforce much tighter security when using some of its more secure services. As we know, some people are not very judicious in their choices of browser extensions. But the downside was that, for example, sites might restrict the use of ad blockers by disallowing their use. Anyway, I hope they didn't kill this because they've figured out some better way to do something similar. And if they did, I would hope that Mozilla would choose not to follow with Firefox.

Cisco IOS XE router attack
Recall the big Cisco IOS XE router vulnerability. That was the one which preceded today's Citrix Bleed mess. It was the result of a web authentication bypass which, as I ranted at the time, was entirely foreseeable and unnecessary because no web UI administration should **ever** be placed onto the public Internet. Cisco should know better. It should not even be an option since there are far more secure ways to do the same thing. And as I was putting this together for the podcast I came up with a new slogan I really like: ***"Easy of use, is no excuse."***

Anyway, this popped back up because Cisco's own Talos group just published a full technical analysis of what they are calling "BadCandy." It's the implant that's being deployed on those compromised Cisco routers thanks to those two 0-day vulnerabilities. In their report, Talos notes that the BadCandy malware has evolved into its third version, demonstrating that the threat actors behind this are still actively modifying their attacks to maintain access to the compromised boxes. They also noted that the latest v3 modifications appear to have worked, since the Shadowserver Foundation who had been monitoring the attack's progression has stopped detecting any infected systems. Presumably, they had long ago captured the IP addresses of the infected and vulnerable systems but no one coming along now would see anything amiss.

Bitwarden Gets Passkeys

Before this next bit of news I'm supposed to remind our listeners that Bitwarden is a sponsor of the TWiT network – as if we weren't all very aware of, pleased with, and even grateful for that fact. So what's the news? With release 2023.10.0 of the Bitwarden browser extension, it now fully supports FIDO2-style Passkeys. Bitwarden's mobile clients have not yet caught up, but this is acknowledged and it's on their development roadmap. Meanwhile, the browser extension appears to be ready for prime time. I have a link in the show notes to the full 2023.10.0 release notes:
https://bitwarden.com/help/releasenotes/#:~:text=Save%20passkeys%20to%20your%20vault%3A   And another link to the specific page discussing Bitwarden's browser extension support for Passkeys: https://bitwarden.com/help/storing-passkeys/

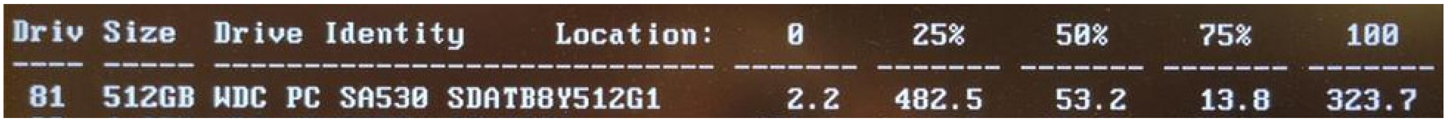
I haven't tried it myself, but from a quick scan of the page it appears to be all there. At the bottom of the "Storing Passkeys" page they have a short Q&A where one of the questions asked is: *"Are stored passkeys included in Bitwarden imports and exports?"* To which they reply: *"Passkeys imports and exports **will be included** in a future release."* So, not there yet, but they clearly recognize the need. And as we noted when we recently talked about this, apparently the slow-moving FIDO group are involving themselves in the creation of the import/export format. Although that's making us wait, we definitely want all Passkeys clients to support a single common cross-platform standard. So I'm quite happy to wait for that.

SpinRite

Recall the Tweet from a listener named Victor from a few weeks ago. His Twitter DM was dated October 18th. In it, he wrote:

> *I powered on a couple of years old desktop that had been unpowered for about a year. It took ages before the desktop was loaded, no errors anywhere but I decided to try your ReadSpeed. And look at those SSD speeds! Is it time to invest in SpinRite now?*
> *If SpinRite fixes this, I will try to encourage my employer to get a site license.*
> *Thank you mr. Gibson. Victor, Long-time SN listener, keep up the good work. To 999 and beyond ;) https://twitter.com/messages/media/1714750052470526215*

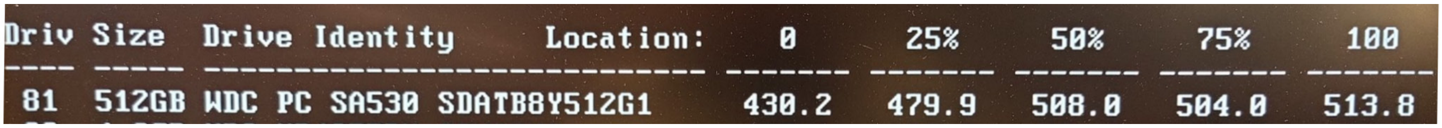
And he attached a screenshot from ReadSpeed which I described at the time. It showed and explained exactly what he was describing:



```
Driv Size  Drive Identity      Location:    0      25%     50%     75%     100
---- ----- ----------------------------- ------- ------- ------- ------- -------
 81  512GB WDC PC SA530 SDATB8Y512G1         2.2   482.5    53.2    13.8   323.7
```

Yesterday, when catching up with my Twitter feed for the podcast I found his follow-up. He did purchase a copy of SpinRite 6.0 then used his 6.0 license to immediately grab the SpinRite 6.1 release candidate. Here's what he wrote:

> *Now Mr. Gibson, I have some results for you, and possibly the listeners. I ran SpinRite 6.1 on Level 4, it took 28 hours, reported 6199 command timeouts, found & repaired 183 defective sectors. For comparison's sake, here is a new screenshot of ReadSpeed. Now the PC behaves like one could expect from an 8 core Intel i9700 with 64gigs of RAM.*
> *https://twitter.com/messages/media/1721493346768630232*



```
Driv Size  Drive Identity      Location:    0      25%     50%     75%     100
---- ----- ----------------------------- ------- ------- ------- ------- -------
 81  512GB WDC PC SA530 SDATB8Y512G1       430.2   479.9   508.0   504.0   513.8
```

Under SpinRite 6.1, a read scan of half a terabyte of directly-connected SSD would have taken less than an hour. But rescuing and resuscitating that very sick SSD required re-writing its recovered data. So that would slow things down. But not by nearly that much – probably only an hour or two. But that SSD was in seriously bad shape and it sounds like it made SpinRite work a lot to pull it back from the brink.

So, SpinRite 6.1 is obviously highly effective for today's solid state storage, in addition to what it has always been able to do for electromagnetic spinning drives. It turns out that electro**STATIC** storage is prone to long term charge loss through several different mechanisms and this only promises to become worse as engineers continue to succumb to pressure from their managers to squeeze ever more data into ever-smaller and fewer storage cells.

The good news is, SpinRite v6.1 can resolve those problems today. It's not as optimal as version 7 will be, but it works now and I'm not stopping once 6.1 is published 6.1 is the first big step forward but much more is on the way once I'm able to get away from DOS.

Closing the Loop

Sam Miorelli / @SamMiorelli

> *Hey @SGgrc, listening to #SN946 on IPv6. Spectrum – the main ISP for Central Florida for Tampa/Orlando/Space Coast – IPv6 is still notoriously problematic. For example, Pixel 3 and later phones go into WiFi disconnect loops when you let IPv6 hit cheap and good routers. For example, I have @NetgateUSA SG-2100. This is well documented on Reddit. Outbound DNS queries also frequently have long periods of time when IPv6 is enabled that they simply black hole on Spectrum (then randomly fixed for a while, then bad again - for months!) ISP shrugs. I fear the day we're forced to switch to IPv6 given how terrible the backend tech is maintained for home ISPs.*

Sam was following up on my somewhat pessimistic appraisal last week of what appears to be the true current state of IPv6. It'll be there when we really need it. But until "really" is in all caps, bold, italics and underlined, all of the prevailing evidence points to everyone doing everything they can to hold onto IPv4 until there's really no other choice.

Bob Grant / @swguru Provided some additional terrific feedback about IPv6

> *Hi Steve, As an IPv6 proponent for a number of years, I listened with interest to your answer to reader mail last week where you said nearly everything appears to be IPv6 ready. I thought I'd share my experience. It is certainly the case that most everyone's routers and many networks are dual stack with both IPv4 and IPv6. In the case of my ISP I am able to request an IPv6 /48 prefix which is 256 /64 networks so I can have as many as 256 separate networks, each having a full /64 IPv6 subnet. My firewall only allows established connections back in so there is no additional security issue over my IPv4 NAT. I recently upgraded my network and WiFI Access Points so it's trivial to segment multiple SSIDs to separate VLANs going back to my OPNsense router.*
>
> [So I'd say that we have, indeed, established that Bob is an IPv6 proponent]
>
> *As an experiment, I decided to set up an IPv6-only network where only IPv6 IPs and DNS would be used. I was quite disappointed to discover how few websites worked with IPv6 only.*
>
> *Huge sites like http://nfl.com, https://twitter.com, and many well-known universities (whom I won't embarrass by naming), and others like http://bitwarden.com all fail to load with an IPv6 only connection.* [And I'll just note that all of my own servers at GRC are among those, too, which are still IPv4-only.] *Many other sites work when using http://www.site.com but fail to redirect when using http://site.com. My three credit unions' landing pages load under IPv6 but the financial backends hosting the login process and displaying account balances fail because they are IPv4 only. Even Microsoft fails after the landing page because the http://login.live.com is not IPv6 enabled. I noticed many site's web pages load only because they use a content delivery network (like CloudFlare or Akamai) that supports both IPv4 and IPv6 at their border, thus proxying for their client's IPv4-only web servers.*
>
> *Kudos to Google, Amazon, Netflix, Facebook, Stanford, MIT, Harvard and the federal government for fully implementing IPv6. I hope some of this is useful. Bob*

VERY useful and interesting, Bob! Thanks for the status of IPv6 today's reality check!

CLT Cyber Security / @cltcyber

Hey Steve! As security pros we know what to do, but I'm having trouble explaining why this is the right way to my company. We have DigiCert TLS certificates for our websites, we need to keep those private keys secure, but if our server was ever compromised by a random attacker who obtained our private keys what could they really do? just trying to better articulate this to non security pros. Thanks, and to the 9's and beyond!

Okay. The danger presented by the compromise of a server's TLS private key certificate is one of impersonation, since it's only the server's possession of that private key certificate that allows it to assert its identity. So if someone else is able to assert that their server is your server, this paves the way to an impersonation attack. Depending upon who you are, that might be either a big deal or not so much. If this was a site that really mattered to its visitors in some way then the consequences could be significant.

However, even though the theft of the certificate may pave the way to such an impersonation attack, there's still a lot of pavement to traverse to pull off a working impersonation attack. The biggest roadblock to implementing the attack is that the web browsers or other connecting clients who would be spoofed by this need to believe that they are actually connecting to the authentic server. In other words, they need to lookup the IP address of the domain of the authentic server and then send TCP traffic back and forth to the IP that was looked up. In practice that means that either the DNS lookup needs to somehow be poisoned to return the attacker's server IP or the victim's IP traffic needs to be intercepted and redirected to the attacker's IP. One way or another, the domain name the client believes it's connecting to must match one of the domains that certificate authenticates.

So either DNS subversion or dynamic traffic interception. If that sounds like a high bar to reach, it can be, but it's entirely dependent upon the specifics of who, or what population, is targeted for spoofing. At the beginning of today's podcast I was talking about layered security. Here's another example of layering. Losing control of a website's certificate is not immediately the end of the world, since other layers are still in place to provide some protection. But having exclusive use of a website's certificate is not a layer you want to give up. This is why the Internet world has gone nuts several times during this podcast. Dan Kaminski's famous discovery of DNS spoofability would also not have been the end of the world, though it would have been bad for HTTP without TLS. Even so, the integrity of DNS wasn't a layer of security that anyone wanted to lose. And similarly the HeartBleed flaw, that potentially allowed some web server certificates to escape, got everyone's attention because, again, it would strip a layer of our multi-layered protection.

And I think that perhaps this also helps to put the never-exploited-as-far-as-anyone-knows Spectre and Meltdown vulnerabilities into a useful light. It might at first appear that the industry was **way** over-concerned about a purely theoretical vulnerability for which no known attack has ever succeeded. But robust inter-process isolation, which Spectre & Meltdown both threatened, is another layer. And in today's heterogeneous cloud computing landscape it's a particularly critical layer.

So, again, I cannot think of an instance where having too many layers of security is a bad thing. Having said that, when over-the-top security gets in the way, that's not good either.

Craig from Scotland / @airchie_uk

Was just listening to SN #941 and the part about public key crypto and factoring primes. It got me wondering, how likely is it that there may be collisions in the primes chosen by two different people? Or would it be feasible to create a rainbow table of factored primes allowing the discovery of the private key using a quick lookup of a public key?

That's a brilliant observation, and in the past it was discovered to be happening with somewhat horrifying regularity. There were problems with the quality of some of the early random number generators which tended to choose and then test for primality the same large prime. So, whoops!! Completely unrelated servers would coincidentally be sharing the same public/private key pairs. Not due to any collusion – well, except for the collusion of them both using the same poor sources of entropy. What was found to be happening was that servers were booted and were immediately being asked to produce a certificate. So the server hadn't yet had time to collect entropy from the environment and it could happen that two completely separate servers would both wind up picking the same keys.

And then, into this we add the birthday paradox. It teaches how quickly the number of collisions between pairs of unrelated items increases as the number of possible interactions increases. There's not a huge danger from sharing the same keys. But it's certainly not zero. First, you would need to compare your server's public key with everyone else's. If you did find a collision, since you know your server's private you also know the colliding party's private key. That's not good. But we already observed that just having that only removes one of the multiple layers of protection needed to exploit any advantage.

The takeaway here is that we don't want to be inadvertently sharing **our** private key with anyone else. So the best way to assure that is to be certain that the process which is picking keys is using the highest quality possible random source for its key guesses.

Chad Cosby / @itschadcosby

Hi Steve, I'm curious if you would share how and how often you run Spinrite on the drives in your Synology NAS. I too use a Synology and it feels like an absurd oversight that I trust my most valued data to an occasional glance at the "Drive Health" meter within DSM.

I suppose the question is how much redundancy are you using. I've never bothered to run SpinRite on any of my RAIDed drives. I have 4-drive RAID arrays everywhere. Every one of GRC's servers is running 4-drive RAID as are both of my Synology NAS's and my one still standing Drobo, though I think it has five drives. In every instance I'm running RAID-6. So that allows me to lose any two drives at the same time without any data loss. And once not too long ago I was nervously flying with no reserve on one server until I could get two replacement drives for it. Had drives ready for it, but I learned that it would not allow me to mix SSDs and spinning drives in the same array. So far, I've never lost any data and I've actually had more trouble with my SSDs than with the spinners. Some spinners just run forever and others seem to tire quickly. They last long enough that I wouldn't really call it infant mortality. It's more like teen angst. Anyway, I'm replacing my SSDs with spinning drives and with them being ridiculously huge and inexpensive I will always be running with RAID-6. And in that case I welcome any drive that gets tired and no longer wants to play to just say so.

John Carling / @JRCsystems

> *Hey Steve, Listening to 946, and hearing about the Requests to extend Windows 10 EOL. The group that did all the testing, are they aware of the recently revealed command line argument to Windows 11 setup?*
>
> ***Setup /product server***
>
> *This will install windows 11 on a windows 10 box that previously failed TPM 2.0 requirements. I've done it on 2 laptops and 1 desktop myself and they work just fine.*

Nice! Thanks John! And, as I was saying last week, this entire thing is arbitrary. It's Microsoft grossly abusing its monopoly position in the industry. I pay Microsoft a chunk of money every year for the privilege of being an MSDN developer with access to their range of workstations and servers for software testing. So I'll be able to move without trouble. But I'd sure rather remain on Win10. I suspect we're going to be seeing some fireworks as this EOL date approaches.

Michael Foley, I stand with 🇺🇦. / @baawfatml

> *Just watched the latest episode, now I have to check all my uses of snprintf for the last 30 years! "GRC" is also the acronym for the French name of the RCMP: The Royal Canadian Mounted Police (RCMP; French: Gendarmerie royale du Canada;* ***GRC****), from Wikipedia.*

So now we know!

Article 45

Okay. So there's a storm brewing in the EU. It's been quietly brewing for some time and it appears that we have another case of politicians mistakenly believing that they're able to simply dictate the terms and conditions under which tech companies will serve their populace regardless of the implications to that populace's security and privacy.

We all just saw something similar come to a head with the attempt to force backdoors into all encryption services. How'd that turn out? Every messaging provider simply said: "No thank you, we'll just leave, and you and your citizens can figure out what to do without us." The result was the addition of a nebulously worded "if it's technically feasible to do without weakening security" which was every strong encryption provider's get out of jail free card.

Now we're moving onto a similar challenge where, believe it or not, the EU might very well find itself and its citizens without any web browsers – or at least needing to return to the good old days of HTTP. The controversy revolves around a made-up thing known as QWACs which stands for Qualified Website Authentication Certificates. These QWACy things are a specific EU form of website certificate defined back in 2014 with the EU's eIDAS regulation. But I'm getting ahead of myself. So let's step back just a bit...

"eIDAS" stands for electronic IDentification, Authentication and trust Services. eIDAS is an EU regulation, first passed nine years ago in 2014. Its stated purpose is governing "electronic identification and trust services for electronic transactions." After it passed in 2014 its various provisions gradually took effect between 2016-2018. That regulation, which never actually did much and was largely ignored (and which, by the way, we did talk about at the time) has been under review and in an updating process for the past several years. It appeared to be going off the rails last year and the tech industry did what it could back then to say: "Hey guys, this is not looking like something we're going to be willing to do for you." But apparently the politicians just figured that they could enact any laws they wanted to and those techie geeks would have no choice other than to comply. Uh huh.

So about a year and a half ago, back in March of 2022, a who's who of global Internet security governance – two pages of co-signer's names and affiliations – wrote an open letter addressed to "Dear Honourable Member of the European Parliament, Dear Member of TELE Working Party," which begins:

> *We the undersigned are cybersecurity researchers, advocates, and practitioners. We write to you, in our individual capacities, to raise grave concerns regarding certain provisions of the legislative proposal for a European Digital Identity framework (the 'eIDAS revision'), and their impact on security on the web.*
>
> *While we understand that the intent of these provisions is to improve authentication on the web, they would in practice have the opposite effect of dramatically weakening web security. At a time when two-thirds of Europeans are concerned about being a victim of online identity theft and over one-third believe they are not able to sufficiently protect themselves against cybercrime, weakening the website security ecosystem is an untenable risk.*
>
> *We therefore urge you to amend the revised Article 45.2 to ensure that browsers can continue to undertake crucial security work to protect individuals from cybercrime on the web.*

To say that this letter appears to have fallen on deaf ears would be an understatement. The near-final text for eIDAS 2.0 has now been agreed upon by the EU's negotiators, and it appears to be even worse than the earlier draft. So now there's a new letter which, as of two days ago, has been signed by 466 scientists and researchers from 36 countries, as well as numerous NGOs. And Google also just added their name to this document. In this day and age, what this document describes is somewhat astonishing and I need to share the first few paragraphs so that you get a feeling for what hangs in the balance:

> *Dear Members of the European Parliament,*
> *Dear Member States of the Council of the European Union,*
>
> *We the undersigned are cybersecurity experts, researchers, and civil society organizations from across the globe.*
>
> *We have read the near-final text of the eIDAS digital identity reform which has been agreed on a technical level in the trilogue between representatives from the European Parliament, Council and Commission. We appreciate your efforts to improve the digital security of European citizens; it is of utmost importance that the digital interactions of citizens with government institutions and industry can be secure while protecting citizens' privacy. Indeed, having common technical standards and enabling secure cross-border electronic identity solutions is a solid step in this direction. However, we are extremely concerned that, as proposed in its current form, this legislation will* ***not*** *result in adequate technological safeguards for citizens and businesses, as intended. In fact, it will very likely result in less security for all.*
>
> *Last year, many of us wrote to you to highlight some of the dangers in the European Commission's proposed eIDAS regulation. After reading the near-final text, we are deeply concerned by the proposed text for* ***Article 45****.*
>
> *The current proposal radically expands the ability of governments to surveil both their own citizens and residents across the EU by providing them with the technical means to intercept encrypted web traffic, as well as undermining the existing oversight mechanisms relied on by European citizens.*
>
> *Concretely, the regulation enables each EU member state (and recognised third party countries) to designate cryptographic keys for which trust is mandatory; this trust can only be withdrawn with the government's permission (see Article 45a(4)).*
>
> *This means any EU member state or third party country, acting alone, is capable of intercepting the web traffic of any EU citizen and there is no effective recourse. We ask that you urgently reconsider this text and make clear that Article 45* ***will not interfere*** *with trust decisions around the cryptographic keys and certificates used to secure web traffic.*
>
> *Article 45 also bans security checks on EU web certificates unless expressly permitted by regulation when establishing encrypted web traffic connections (see Article 45(2a)). Instead of specifying a set of minimum security measures which must be enforced as a baseline, it effectively specifies an upper bound on the security measures which cannot be improved upon without the permission of ETSI.*

Skipping ahead a few pages, here's some detail that's actually difficult to believe, but it's true:

> *“The current text of Article 45* ***mandates*** *that browsers* ***must*** *accept any root certificates provided by any Member State (and any third party country approved by the EU). This will have severe consequences for the privacy of European citizens, the security of European commerce, and the Internet as a whole.*
>
> *Root certificates, controlled by so-called certificate authorities, provide the authentication mechanisms for websites by assuring the user that the cryptographic keys used to authenticate the website content belonging to that website. The owner of a root certificate can intercept users’ web traffic by replacing the website’s cryptographic keys with substitutes he controls. Such a substitution can occur even if the website has chosen to use a different certificate authority with a different root certificate.*
>
> *Any root certificate trusted by the browser can be used to compromise any website. There are multiple documented cases of abuse, because the security of some certificate authorities has been compromised. To avoid this, there exists legislation that regulates certificate authorities, complemented by public processes and continuous vigilance by the security community to reveal suspicious activities.*
>
> *The proposed eIDAS revision gives Member States the right to insert root certificates at will, with the aim to improve the digital security of European citizens by giving them new ways to obtain authentic information of who operates a website. In practice, this does exactly the opposite.*
>
> *Consider the situation in which one of the Member States (or any of the third party states recognized now or in the future) were to add a new authority to the EU Trusted List. The certificate would have to be immediately added to all browsers and distributed to all of their users across the EU as a trusted certificate. By using the substitution techniques explained above, the government-controlled authority would then be able to intercept the web traffic of not only their own citizens, but all EU citizens, including banking information, legally privileged information, medical records and family photos. This would be true even when visiting non-EU websites, as such an authority could issue certificates for any website that all browsers would have to accept. Additionally, although much of eIDAS2.0 regulation carefully gives citizens the capability to opt out from usage of new services and functionality, this is not the case for Article 45. Every citizen would have to trust those certificates, and thus every citizen would see their online safety threatened.*
>
> *Even if this misbehavior was discovered, under the current proposal it would not be possible to remove this certificate without the ultimate approval of the country having introduced the certificate authority. Neither eIDAS’s article 45 nor any provisions in adjacent EU legislation such as the NIS2 Directive provide any independent checks and balances on these decisions. Further, European citizens do not have an effective way to appeal these decisions. This situation would be unacceptably damaging to online trust and safety in Europe and across the world. We believe this legislative text must be urgently reworked to avoid these serious consequences by clarifying that eIDAS does not impose obligations to trust cryptographic keys used for encrypted web traffic.*

This letter goes for seven pages before we get to the 14 pages of signatures by everyone in the world in a place of authority who knows anything about the way our Internet security and privacy ecosystem is put together.

Mozilla authored their own letter which was dated last Thursday, November 2nd. It was co-signed by the the Bytecode Alliance, Cloudflare, DNS0.EU, Fastly, Internet Security Research Group (ISRG), the Linux Foundation, Mozilla, Mullvad, OpenSSF, Sigstore. It begins:

> *Dear Members of the European Parliament,*
> *Dear Member States of the Council of the European Union,*
>
> *We represent companies that build and secure the Internet. Our organizations are either based in Europe or offer products and services in Europe.*
>
> *We write to express our concern with the proposed eIDAS legislation. We appreciate efforts to use rulemaking to strengthen the security of the Internet and the leadership role that Europe has taken in fostering cross-border interoperability. However, leadership comes with a greater responsibility to consider the broader implications of changes.*

... and it goes on from there to express the same concerns and issues as the earlier open letter.

So now the question is, what happens next? A full year and a half ago the legislators were warned about this and were given a heads-up – with a full, detailed, careful and respectful explanation. They were very clearly told: Do not proceed down this path. They clearly blew it off, ignored it completely, and since then the wording of Article 45 has only grown more intolerable.

We've observed that in high-level high-stakes politics it's necessary to give the player who's holding the weaker hand a face saving way to back down. This happened with the encryption debate where the loser in that struggle created their own way to save face. But that didn't happen until those holding the stronger hand – the encryption service providers – were finally **forced** to deliver the ultimatum: *"If you outlaw our use of unbreakable encryption, you will leave us with no option other than to withdraw our then-illegal services from your territories."*

Is this going to come to that? Is this going to get to ultimatums? At this point, it appears so. And this will be another important juncture in the evolution of our Internet. Governments are going to learn, again, that they are smaller than the technology which they and their citizenry have grown to depend upon. It is theirs to use, but not to control.

The original open letter from March of 2022:
https://www.eff.org/files/2022/03/02/eidas_cybersecurity_community_open_letter_1_1.pdf

Today's 21-page (14 of signatures) updated letter to the EU:
https://nce.mpi-sp.org/index.php/s/cG88cptFdaDNyRr

Mozilla's own open letter to the EU:
https://blog.mozilla.org/netpolicy/files/2023/11/eIDAS-Industry-Letter.pdf

The proposed and agreed upon eIDAS 2.0 legislation:
https://www.europarl.europa.eu/RegData/docs_autres_institutions/commission_europeenne/com/2021/0281/COM_COM(2021)0281_EN.pdf