Security Now! #788 - 10-13-20

Well Known URI's

This week on Security Now!

This week we catch up with Chrome 86's handful of security-related improvements, we touch on several recent Ransomware events, and on the consequences of not logging free WiFi users in France. We look at the results of an amazing bit of hacking of Apple, update on the enduring ZeroLogon threat, introduce the revenge of DNT with legislation-enhanced GPC, and another renewed attack on undecryptable E2EE now by seven countries. Then, following a bit of SpinRite and GRC forum news, we're going to add the concept of IANA-registered well-known URIs to our bag of tricks knowledgebase.

And, if you didn't already think that hard drive technology had totally jumped the shark...

Triple Stage Actuator

Western Digital's 18TB and 20TB HDDs integrate the industry's first TSA onto a 9-disk platform. The TSA makes use of three pivot points: the Voice Control Motor (VCM) Actuator, the Milliactuator and the Microactuator. Using three pivot points enables a higher bandwidth servo control resulting in a more precise positioning of the head on the track. With greater head position accuracy, tracks can be written closer together for higher TPI and greater areal density, resulting in higher capacity HDDs.

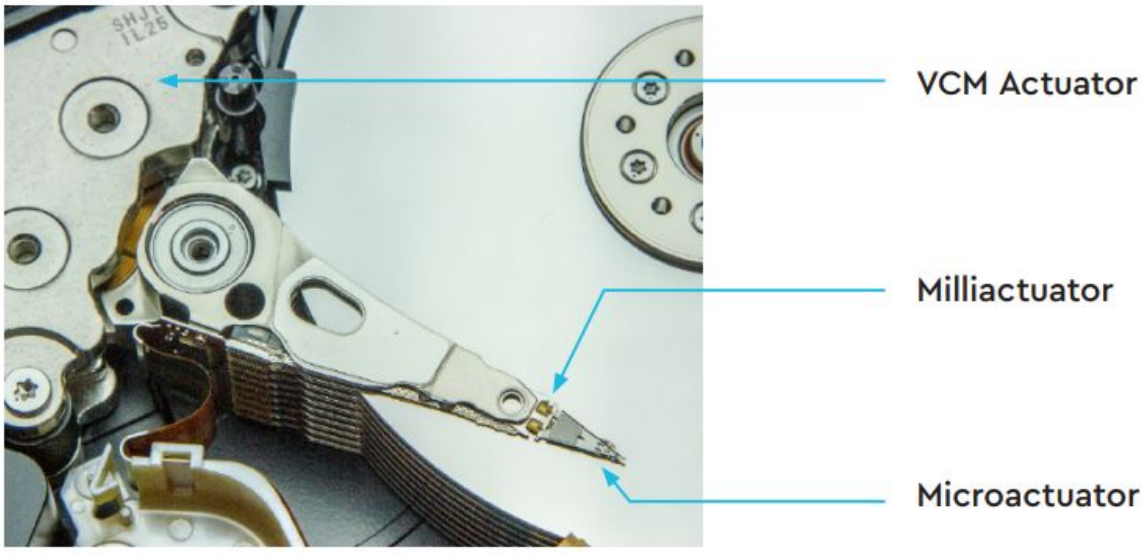

Figure 5: Triple Stage Actuator (TSA)

Browser News

Chrome gets 86'd!

The main theme for Chrome 84 was updates to its user interface. And the recently released Chrome 85 was primarily about performance and API enhancements. And last week's Chrome 86, which was dropped into the stable channel last Tuesday, focuses primarily upon enhanced security features and still more API enhancements, including "what could possibly go wrong?" access to the entire file system of its hosting OS. But more on that in a minute.

As usual, your Chrome may need, as mine did, a bit of encouragement to make the jump. I'd been using it daily for the past week while 86 was "out there." But it wasn't until I was assembling this podcast last night that I went to "Help / About" that it said "Oh, crap! ... And updated itself from 85 to 86."

Last December, with Chrome 79, our desktop versions of Chrome acquired the "Password Checkup" feature which would scan the user's saved and sync'd passwords for collisions with known leaked passwords from other site's data breaches. Google renamed that feature "Safety Check" last May, and now this useful feature has been added to the mobile editions of Chrome for Android and iOS.

With Chrome 86, we now also have built-in support for a particular type of Well Known URI. It was the much more broad and general case of this that has become this week's podcast topic. So we'll discuss that at greater length before we wrap up this week.

Chrome users on iOS are also getting the touch-to-fill feature which Android users have had since July. It's essentially biometric authentication for password filling on iOS. Chrome detects the site that the user is navigating and will then automatically prompt the user to autofill passwords if it has credentials for the site. Since Chrome will only trigger on an exact domain name match, this has the benefit that's common with password managers of helping to avoid phishing and visually close spoofing site names. And since iOS offers strong biometrics, Chrome 86 will support that available hardware, requiring a quick biometric re-authentication before it will auto-fill.

And, yeah, we've all had this with our industrial strength password managers for some time. So none of this is new for us. And our password managers are not only cross-platform, but also cross-browser-make, which Chrome, of course, is not. But there are many people who have still not made the leap to a 3rd-party password manager. And if we haven't got 'em by now, it seems unlikely that we will. So building much stronger security into the base browser they're always going to be using seems like a useful compromise.

Also, in May of this year with Chrome 83, our desktop Chrome's acquired the "Enhanced Safe Browsing" feature to provide more phishing and malware detection. With 86, the mobile platforms will both be acquiring that protection, too.
I was surprised to learn that earlier Chrome wasn't already warning its users when submitting insecure form data. That seems like a no-brainer which we, of course, have been talking about for more than a decade. But at least Chrome is now offering that protection.

And Chrome will continue with its warning and blocking when downloading insecure assets from secure pages. In 86, executable and archive files are blocked by default, while Chrome shows warnings for office-related document downloads.

We've noted that browser-based file transfer protocol (FTP) is not long for this world. Its death will occur in stages, beginning now.

With today's Chrome 86 - FTP will still be enabled by default for (most) stable channel users, but it will be disabled for the Canary and Beta pre-release channels. I said "most" stable channel users because FTP will also be experimentally disabled for one percent of stable users. If you find that it's disabled for you, and you need it — after you get through asking yourself why — and, really, do take a moment — 86 will allow you to re-enable it from the command line using the --enable-ftp command line flag.

When we get to Chrome 87, the disabled percentage will be increased to a coin toss with 50% of Chrome 87 users discovering to their dismay that their favorite FTP resources can no longer be downloaded.

And, finally, with Chrome 88, FTP will be completely sunsetted.

And now, as for that new "what could possibly go wrong?" new feature, get a load of this:

Chrome 86's new "Native File System API" is activated by default in Chrome 86. Google boasts that it will enable developers to build powerful web apps that interact with files on the user's local device. Naturally, you wouldn't want any random malvertising ad to have full and unfettered API access to your entire machine. So the new API is locked up hidden behind a permission prompt to prevent websites from accessing any local files without authorization. And after a user grants the browser access, this API allows a website to behave like a locally installed app, reading, saving and interacting with files and folders on the user's device. Google expects this new API to be used to power a broad range of interactive web apps such as IDEs, photo and video editors, text editors, and more...

To which, I rhetorically repeat "What could possibly go wrong?" You just know that someone is going to want to automate that pesky permission prompt. I mean, really, who wants to have to be constantly giving permission?

Ransomware

Carnival Corporation

The world's largest cruise line operator, Carnival Corporation, confirmed in their FCC filing last week that the personal information of customers, employees, and ship crews was stolen during an August ransomware attack. Carnival employs more than 150,000 people from roughly 150 countries and caters to — or at least used to — over 13 million guests each year.
Although Carnival hasn't disclosed anything about the attack, the cybersecurity intelligence firm Bad Packets discovered that Carnival had multiple potential points of initial entry and compromise which a ransomware attacker might use to get in. Specifically, multiple Citrix Application Delivery Controller devices and Palo Alto Networks firewalls. The Citrix ADC was

found to be vulnerable to CVE-2019-19781. The firmware was updated in January, but not by Carnival. And the Palo Alto Network firewalls had the CVE-2020-2021 problem, which was patched at the end of June 2020... But again, Carnival didn't get the memo. On the other hand, it's difficult to imagine an industry that was hit harder by COVID-19 than cruise lines. In any event, both of these vulnerabilities can be used by ransomware gangs as stepping stones to breach a corporate network allowing them to move laterally and collecting credentials needed to take over admin accounts and get into the Windows domain controller.

Ransomware + "Access Providers"

There's a chilling but, in hindsight, not unexpected new development in the Ransomware scene: The major ransomware network operators are beginning to purchase access to rich corporate networks from independent "Access Providers." In other words, a new layer of specialization is emerging as the ransomware cybercrime methodology continues to mature.

Accenture's Cyber Threat Intelligence (CTI) team has released new research on emerging cybersecurity trends. This includes the results of their investigation into the nature of relationships between ransomware operators and exploit sellers.

https://www.accenture.com/us-en/blogs/cyber-defense/destructive-relationship-between-network-access-sellers-and-ransomware-groups

In their piece titled "***Shady deals: The destructive relationship between network access sellers and ransomware groups***" they explain:

> Ransomware groups are taking advantage of opportunities to purchase network access on dark web forums to quickly compromise networks across a variety of industries and unleash their disabling malware. Network Access Sellers' expertise lies in the ability to gain corporate and government network access, which they then sell to other cyber-crime groups for a handsome profit. These cyber-crime groups can use purchased network access to slash the typical difficult requirement of gaining initial access, establishing persistence, and moving laterally across a network.
>
> Network Access Sellers typically develop an initial network vulnerability and infiltrate the victim network to gain complete corporate network access. Once that access is gained, the network access sellers sell it on dark web forums, usually for anywhere between US$300 and US$10,000, depending on the size and revenue of the victim.
>
> The majority of network access offerings are advertised on underground forums with some or all of the following information:
>
> - Generalized victim industry information (for example private corporation, medical institution, governmental agency, educational etc)
> - Country the victim operates in
> - Type of access for sale (for example "VPN", "Citrix", "RDP")
> - Number of machines on the network
> - Additional company information (for example number of employees, and revenue)

The amount of information provided can occasionally lead to the identification of the victim.

Accenture CTI assesses that the network access market has been driven by the increased diversity of ways that data can be monetized. Previously, cyber criminals wishing to make a profit on underground forums primarily targeted financial data due to its ease of monetization. However, the Nikolay threat-group (aka Fxmsp) popularized selling network accesses beginning in 2018 by proving there was a large demand for their service and that regular sales could be highly profitable. Although financial data remains central to underground economies, sensitive Personally Identifiable Information (PII) and company data, or the promise of access to this data, is profitable because this data can be further monetized through direct sale or by holding it ransom.

Since the start of 2020 and the emergence of the now-popular "ransomware with data theft and extortion" tactics, ransomware gangs have successfully utilized dark web platforms to outsource complicated aspects of a network compromise. A successful ransomware attack hinges on the development and maintenance of stable network access which comes with a higher risk of detection and requires time and effort. Access sellers fill this niche market for ransomware groups.

As of September 2020, we actively track more than 25 persistent Network Access Sellers as well as the occasional one-off seller, with more entering the scene on a weekly basis. Network Access Sellers operate on the same forums as actors associated with the ransomware gangs Maze, Lockbit, Avaddon, Exorcist, NetWalker, Sodinokibi, and others.

We assess with high confidence that this ecosystem will continue to thrive, so long as reputable, invite-only dark web forums provide the platform on which network access sellers and ransomware gangs can securely exchange goods and services.

So, yeah... what we're seeing is the emergence of specialization: The ransomware market has become so profitable that it will support horizontal integration. In other words, it's not necessary for the extortionists to be vertically integrated and doing everything themselves. They can afford to outsource the front end work of obtaining access to corporate networks and they can afford to pay a pretty penny for that access. That allows hackers with network penetration skills to focus **only** on getting in, without needing to have the skill set to monetize the access that they obtain. Because, now there's an eager and active bidding market for that access on the Dark Web.

Tyler Technologies

The largest software company in North America you've never heard of — Tyler Technologies — services the public sector, with over $1.2 billion in annual revenue and 5,500 employees. And Wednesday, September 23rd, Tyler was hit with a cyberattack by the RansomExx ransomware operators — the same team that was behind recent attacks on Konica Minolta and IPG Photonics.

Tyler immediately disconnected portions of their network to prevent the ransomware's spread and to limit their many clients' exposure. CIO Matt Bieri emailed clients, writing: "Early this morning, we became aware that an unauthorized intruder had disrupted access to some of our internal systems. Upon discovery and out of an abundance of caution, we shut down points of

access to external systems and immediately began investigating and remediating the problem."

They did succeed in containing and preventing its spread into their clients' networks. Tyler said that they were severely impacted and expected that it would take 30 days to recover operations fully. So, Tyler paid the ransom to recover their encrypted data, though they've not saying how much they paid. RansomExx is one of the groups known to exfiltrate data before encrypting it, and to then threaten to release the stolen data unless a victim pays the ransom. Since many school districts, court systems, and local and state governments in the US are Tyler Technologies customers, the risk of public disclosure of sensitive information is significant. So this concern may have been a significant factor in the decision to pay the ransom.

And...
Several public school districts have recently been hit by ransomware. Classes for tens of thousands of remote learning and hybrid students have been affected, and hundreds of schools have suffered closures. The ransomware

Security News

No connection logs? In France you go to jail!
The week before last, at least five bar and cafe managers in the French city of Grenoble were arrested and taken into custody for running open WiFi networks at their establishments and not keeping logs of previously-connected users.

The bar and cafe owners were arrested for violating a 14-year-old law in France that dictates that all Internet Service Providers must keep logs on all of their users for at least one year. Naturally, the bar and café owners never considered that such a law might apply to them, not being commercial Internet Service Providers. But, by the letter of the law, they are, indeed, providing Internet Services. The bar and cafe owners were unaware that such a law even existed, let alone that it applied to them, since they had never received any notifications from their union, which usually sends alerts of industry-wide legal requirements. However, in their coverage of the incident, the French media pointed out that the law's text didn't only apply to internet service providers (ISPs) in the broad meaning of the word — as in telecommunications providers — but also to any persons who provide internet access, may it be free of charge or via password-protected networks.

And although the bar and cafe owners were released after questioning, that might not be the end of it since, according to French law number 2006-64, they now risk up to one year in prison, a personal fine of up to €75,000, and a business fine of up to €375,000. Whew!

As countries around the world began introducing data logging laws for their local ISPs, connection logging is a feature that's now offered by most commercial routers. The trouble is, this makes far more sense for actual ISP than for open WiFi access points. In a world where iOS is now fully randomizing its device's MAC address there is literally nothing to log. Or at least nothing worth logging. Open WiFi in a cafe will obtain and could log the MAC address claimed to be owned by the device.
Ten years ago that might have been useful. But no longer. Once upon a time, law enforcement

agencies were able to use such logs to track down malicious behavior or details about suspects who were using public WiFi networks to commit crimes. As we know, 24 bits of the 48-bit MAC would identify the manufacturer of the device, and if the remaining 24 bits were manufacturer assigned and static, then they could, indeed, uniquely identify a specific laptop or phone.

So here's an instance where a feature designed to enhance its user's privacy is doing exactly that, perfectly... And rendering a user of a WiFi access point truly anonymous.

Hacking the Apple
What do you get when a team of 5 talented security researchers settle down during the global pandemic to poke at Apple's online services for three months during July, August and September?

What you get is some significantly more secure services as a result of their discovery and responsible disclosure of 55 vulnerabilities, 11 of which are critical.

And what they got, for 28 of the vulnerabilities that Apple has been able to process so far, is a total bug bounty payout of more than a quarter million dollars: $288,500 to be precise.

https://samcurry.net/hacking-apple/
Sam Curry, the lead researcher, blogged: “When we first started this project we had no idea we'd spend a little bit over three months working towards its completion. This was originally meant to be a side project that we'd work on every once in a while, but with all of the extra free time with the pandemic we each ended up putting a few hundred hours into it.""
In addition to the 11 critical flaws, 29 were high severity, 13 medium, and 2 low severity. They could have allowed an attacker to “fully compromise both customer and employee applications, launch a worm capable of automatically taking over a victim's iCloud account, retrieve source code for internal Apple projects, fully compromise an industrial control warehouse software used by Apple, and take over the sessions of Apple employees with the capability of accessing management tools and sensitive resources.”

All I can say is... I'm glad these guys are on our side!

The flaws meant a bad actor could easily hijack a user's iCloud account and steal all the photos, calendar information, videos, and documents, in addition to forwarding the same exploit to all of their contacts.

To Apple's credit — and likely to their shock and horror — once the flaws were responsibly disclosed, Apple patched the flaws within 1-2 business days, with a few of them fixed in 4 to 6 hours.

I especially loved Sam's description of how this all happened — and please take heed all of you other big companies, because it's Apple who is now so much richer for having dangled and paid those bounties. Here's how this all began:

Sam wrote:

While scrolling through Twitter sometime around July I noticed a blog post being shared where a researcher was awarded $100,000 from Apple for discovering an authentication bypass that allowed them to arbitrarily access any Apple customer account. This was surprising to me as I previously understood that Apple's bug bounty program only awarded security vulnerabilities affecting their physical products and did not payout for issues affecting their web assets.
[Note that Sam's byline reads: "Web Application Security Researcher"]

> Zero-day in Sign in with Apple - bounty $100k https://t.co/9lGeXcni3K
> — Bhavuk Jain (@bhavukjain1) May 30, 2020

"What if I say, your Email ID is all I need to takeover your account on your favorite website or an app. Sounds scary, right? This is what a bug in Sign-in-with-Apple allowed me to do.

In the month of April, I found a zero-day in Sign-in-with-Apple that affected third-party applications which were using it and didn't implement their own additional security measures. This bug could have resulted in a full account takeover of user accounts on that third party application irrespective of a victim having a valid Apple ID or not.

For this vulnerability, I was paid $100,000 by Apple under their Apple Security Bounty program.

[That caught Sam's attention. His blog posting continues...

After finishing the article, I did a quick Google search and found their program page where it detailed that Apple was willing to pay for vulnerabilities "with significant impact to users" **regardless** of whether or not the asset was explicitly listed [as] in scope.
This caught my attention as an interesting opportunity to investigate a new program which appeared to have a wide scope and fun functionality. At the time, I had never worked on the Apple bug bounty program, so I didn't really have any idea what to expect but decided "why not try my luck and see what I could find?"

In order to make the project more fun, I sent a few messages to hackers I'd worked with in the past and asked if they'd like to work together on the program. Even though there was no guarantee regarding payouts, nor an understanding of how the program worked, everyone said yes, and we began hacking on Apple.

Bad guys are highly motivated by their own self interest to find high-value flaws in other peoples' work. But it's not **only** the bad guys who are good hackers. It's the bad guys who are typically the most motivated. Everybody needs to put bread on the table — good guys and bad guys. And much as good guys might **want** to spend their valuable time fixing other peoples' stuff, they cannot typically afford to. This is why generous bug bounty programs work. It was the story of the $100,000 windfall that caught Sam's attention. And unless it had, we would not be recounting this story and all of those previously undiscovered problems would have remained undiscovered. Someone would have eventually found them. The question is who?

ZeroLogon

So, first we have Microsoft's update last week. What we know is that after the initial flurry of immediate action leveraging ZeroLogon, it has now been taken up by the mainstream serious threat actors. It has been incorporated into the campaigns of several well known ransomware extortionists. Being a fundamental flaw in a core security protocol and given the demonstrated lack of patching vigilance we keep seeing, it will likely go down in history as one of the more devastating Windows vulnerabilities. And there's more than a little competition for that role.

ZeroLogon, the FBI, DHS and our forthcoming election security:
And then we have the unfortunate timing of all this, since the FBI and the CISA, the cyber-security arm of DHS, last week said that they've detected hackers exploiting the ZeroLogon vulnerability against state and local governments where, in some cases, the attacks are being used to breach the networks used to support elections.

[As an aside, I recently noted that Florida's early vote counting machines are already up, running, and humming along since early voting this year has blown through all past records. So, I was gratified to hear the mention dropped into the reporting that none of their machines are on the Internet — just the clear awareness that that's important was some comfort. So thank goodness for that. I dearly hope that off-the-Net operation will be common practice. The machines are internally accruing vote totals and they will not be queried until after the polls have closed.]

But, as for the ZeroLogon concerns of the FBI & DHS, last week they wrote:

> This recent malicious activity has often, but not exclusively, been directed at federal and state, local, tribal, and territorial (for which SLTT is an abbreviation) government networks. Although it does not appear these targets are being selected because of their proximity to elections information, there may be some risk to elections information housed on government networks.
>
> CISA is aware of some instances where this activity resulted in unauthorized access to elections support systems; however, CISA has no evidence to date that integrity of elections data has been compromised. There are steps that election officials, their supporting SLTT IT staff, and vendors can take to help defend against this malicious cyber activity.

Yeah... stop using Windows for anything that's mission critical. Find the most security-hardened Unix or Linux around and use that.

What has been observed is that to gain their initial access, the attackers are exploiting vulnerabilities in firewalls, VPNs, and other products from companies including Juniper, Pulse Secure, Citrix, and Palo Alto Networks. All of the vulnerabilities—Zerologon included—have received patches, but as we know, many critical systems and networks have not updated. In this instance, this lack of action is placing governments and elections systems everywhere, at risk.

The revenge of DNT, as GPC, now enhanced with legislation
https://globalprivacycontrol.org/press-release/20201007.html
Last Wednesday's announcement reads:

> Announcing Global Privacy Control: Making it Easy for Consumers to Exercise Their Privacy Rights
>
> With the introduction of privacy regulations such as the California Consumer Privacy Act (CCPA) and the General Data Protection Regulation (GDPR), consumers have more rights to limit the sale and sharing of their personal data than ever before. CCPA in particular gives California residents a legal right to opt out of the sale of their data and requires businesses to respect user preferences through a signal from their web browser communicating the consumer's request to opt out.
>
> While this is great progress, it doesn't amount to much if it is hard for people to take advantage of their new rights. Today, there is no defined or accepted technical standard for how such a web browser signal would work. Without that, users don't have an easy way to express their preferences.
>
> Indeed, in his recent testimony before the US Senate, California Attorney General Xavier Becerra explained:
>
> One provision of our regulations intended to facilitate the submission of a request to opt-out of sale by requiring businesses to comply when a consumer has enabled a global privacy control at the device or browser level, which should be less time-consuming and burdensome. I urge the technology community to develop consumer-friendly controls to make exercise of the right to opt out of the sale of information meaningful and frictionless.
>
> This effort, initially spearheaded by Georgetown Law and Wesleyan University, now includes The New York Times, The Washington Post, Financial Times, Automattic (WordPress.com & Tumblr), Glitch, DuckDuckGo, Brave, Mozilla, Disconnect, Abine, Digital Content Next (DCN), Consumer Reports, and the EFF.
>
> In the initial experimental phase, individuals can download browsers and extensions from Abine, Brave, Disconnect, DuckDuckGo, and EFF in order to communicate their "do not sell or share" preference to participating publishers. Additionally, we are committed to developing GPC into an open standard that many other organizations will support and are in the process of identifying the best venue for this proposal.

We look forward to working with Attorney General Becerra to make GPC legally binding under CCPA. At the same time, we are exploring GPC's applicability and functionality with regard to other similar laws worldwide, such as the GDPR. We are excited about the prospect of empowering people with an easy-to-use tool to exercise their privacy rights.

https://globalprivacycontrol.org/

A user agent MUST NOT generate a Sec-GPC header field if the user's Global Privacy Control preference is not enabled.

A user agent MUST generate a Sec-GPC header field with a field-value that is exactly the numeric character "1" if the user's Global Privacy Control preference is set.

Example 1: Example GPC Request

```
GET /something/here HTTP/1.1
Host: example.com
Sec-GPC: 1
```

The Sec-GPC is deliberately defined without an extension mechanism. Experience with previous similar headers shows that people tend to rely on string equality instead of parsing the value when testing for their presence, especially when extensions do not yet exist. Such checks would of course fail in the presence of extension content, which would in turn render the mechanism moot. Should extensions prove necessary to this standard, they will need to be implemented through other headers, which may in time supersede this one.

The Sec-GPC signal sent MUST reflect the user's preference, not the choice of some vendor, institution, site, or network-imposed mechanism outside the user's control. The basic principle is that a Sec-GPC preference expression is only transmitted if it reflects a deliberate choice by the user. What constitutes a deliberate choice may differ between regional regulations. For example, regulations in one jurisdiction may consider the use of a privacy-focused browser to imply a Sec-GPC preference, such as under the CCPA Final Statement of Reasons - Appendix E #73 ("The consumer exercises their choice by affirmatively choosing the privacy control, including when utilizing privacy-by-design products or services"), while regulations in another jurisdiction may require explicit consent from the user to send a Sec-GPC preference.

It should be noted that users' preferences for privacy is well established, and user agents are expected to convey user preferences as accurately as they can. To the extent possible, user agents SHOULD strive to represent user preferences, including if necessary by prompting users to decide whether they would prefer to have their data sold or not.

A site MAY produce a resource at a **.well-known URL** in order for a site to represent the fact that it abides by GPC. A GPC Support Resource has the well-known identifier /.well-known/gpc relative to the origin server's URL.

[In addition to Chrome 86, this is another reason for this week's topic... which we'll be getting to shortly.]

An origin server that receives a valid GET request targeting its GPC support status resource MUST send either a successful response containing a machine-readable representation of the site-wide tracking status, as defined below, or a sequence of redirects that leads to such a representation (possibly provided by a server at another origin).

```
GET /.well-known/gpc HTTP/1.1
Host: example.org
User-Agent: whatever


Content-Type: application/json

{
 "gpc": true,
 "version": 1
}
```

During the initial experimental phase, the GPC signal is not intended to convey legally binding requests; it is instead intended as a way to test effective protocols for communicating and complying with user requests to stop the sale or sharing of their personal information.

After the experimental phase, receiving a GPC signal may have legal effects, depending on factors such as the location of the individual sending the signal, the scope of the applicable law, as well as any separate agreement between the recipient of the signal and the individual.

For example, the use of the GPC signal by an individual will be intended to communicate the individual's intention to invoke the following rights, as applicable:

- Under the CCPA, the GPC signal will be intended to communicate a Do Not Sell request from a global privacy control, as per [CCPA-REGULATIONS] §999.315 for that browser or device, or, if known, the consumer.

- Where the GPC signal conflicts with the existing privacy settings a consumer has with the business, the business shall respect the GPC signal but may notify the consumer of the conflict and give the consumer an opportunity to confirm the business-specific privacy setting or participation in the financial incentive program [CCPA-REGULATIONS] §999.315(c)(2).

- Under NRS 603A, (Nevada Revised Statutes) a GPC signal will be intended to communicate a “Do Not Sell My Personal Information” request [SB220].

- The GDPR requires that "Natural persons should have control of their own personal data" ([GDPR], Recital 7). The GPC signal is intended to convey a general request that data controllers limit the sale or sharing of the user's personal data to other data controllers ([GDPR] Articles 7 & 21). This request is expressed with every interaction that the user agent has with the server.

> Note that this request is not meant to withdraw a user's consent to local storage as per the ePrivacy Directive ("cookie consent") ([EPRIVACY-DIRECTIVE]) nor is it intended to object to direct marketing under legitimate interest ([GDPR]).
>
> Given the complexities of existing consent frameworks, publishers who accept the GPC signal should disclose how they treat the GPC signal in that jurisdiction and how they deal with conflicts between the signal and other specific privacy choices that the user has already made directly with the publisher, including instances where third party sharing may be permitted such as sharing to service providers/processors, sharing at law or at the direction of the individual.

So, GPC is not synonymous with the failed Do Not Track header, but it's clearly going to become a flag which will initially launch many a lawsuit. Once legislation is in place, aggregators of personal information, such as advertising networks and other information gathering agencies that have been hiding behind browser redirect chains, will become subject to new scrutiny. And once all of our user agent browsers are capable of sending out the GPC beacon, which seems clearly destined to occur, we'll have a pain free means of asserting our personal preferences.

Now, if we could only have another static beacon which asserts "Yes, yes, yes, I know you need to store a cookie here, it's fine... just don't ask me." Then we'd really be set. :)

The Anti-E2EE drumbeat beats yet again

Members of the Five Eyes intelligence-sharing alliance, in addition to government representatives from Japan and India, have collectively published a statement over the weekend calling on tech companies to come up with a solution for law enforcement to access end-to-end encrypted communications — in the alliance's updated effort to push tech companies to develop some means for providing warrant-friendly encryption. So now this is the US, the UK, Canada, Australia, and New Zealand with Japan and India.

Representatives from all seven governments argue that the way E2EE encryption is currently supported on today's major tech platforms prohibits law enforcement from investigating crime rings, but also the tech platforms themselves from enforcing their own terms of service.

Signatories argue that "particular implementations of encryption technology" are currently posing challenges to law enforcement investigations, as the tech platforms themselves can't access some communications and provide needed data to investigators.

And once again they're leveraging child abuse by saying: "This, in turn, allows a safe haven for criminal activity and endangers the safety of "highly vulnerable members of our societies like sexually exploited children." In the seven governments press release, they wrote:

> We call on technology companies to work with governments to take the following steps, focused on reasonable, technically feasible solutions,
>
> Embed the safety of the public in system designs, thereby enabling companies to act against

illegal content and activity effectively with no reduction to safety, and facilitating the investigation and prosecution of offences and safeguarding the vulnerable;

Enable law enforcement access to content in a readable and usable format where an authorisation is lawfully issued, is necessary and proportionate, and is subject to strong safeguards and oversight; and

Engage in consultation with governments and other stakeholders to facilitate legal access in a way that is substantive and genuinely influences design decisions.

The government officials said they're committed to working with tech companies to develop solutions to allow users to continue using secure, encrypted communications, but to also allow law enforcement and tech companies to crack down on criminal activity.

The seven governments called for a means of decryption not only for encrypted instant messaging applications, but also for "device encryption, custom encrypted applications, and encryption across integrated platforms." (So that would include VPNs.)

As we know, I've often argued that Apple could enable iMessage and Facetime wiretaps, as could Zoom — in those instances they are managing the keys on behalf of their users. But even I see no possibility for feasibly decrypting unmanaged applications such as Threema or VPNs. That really WOULD require the insertion of an encryption backdoor, and that would be an unmitigated disaster.

SpinRite

I'm continuing to work toward the release of our "ReadSpeed" benchmark. The reason this is taking so long is that since this technology will be moved into SpinRite, it seemed to me that in the interest of minimizing the total delivery time for SpinRite — which is what we all want — I should be developing this code for its final form in SpinRite. So, most of what I've been doing is directly working on SpinRite, even though the first form it will take will be a full, system wide, mass storage benchmark. Once we have that working perfectly, what I will have can be neatly dropped right into SpinRite, since I've written it for SpinRite's direct use.

And the GRC forums are continuing to develop very nicely. We have 3,370 registered users and generally around 70 or 80 unregistered people visiting. The so-called "Off-Topic Lounge" has been so popular that it's clear that a community of sorts is beginning to form. So, although it wasn't my original plan, I'll be dividing that one forum into a handful of on-topic sub forums. We'll have a Security Now forum, and forums for security hardware and software, operating systems, networking, health, Sci-Fi and more. It's still my intention for the forum's primarily purpose to be the support of GRC's commercial and freeware offerings, but we have a bunch of terrific members who want to hang out and chat with similar terrific people, so I'm more than happy to provide a place for that.

Sci-Fi

And speaking of Sci-Fi...
Ryk Brown has just dropped book 15 of the second series of 15 of his truly fabulous Frontier's Saga series — which I recommend without reservation. It's wonderful, pleasant, pure escapist science fiction featuring a cast of very well formed and diverse characters who you really get to know and appreciate. And it has a lot of humor. He's one hell of a story teller.

So this book wraps up the second series of 15, being the 30th book, overall. Ryk has sketched out a 5 series, 75-book story arc... and I, for one, hope that he never runs out of steam because this is one of the best, easiest to read, action-packed space operas I've ever encountered.

However, what Ryk HAS run out of is patience with Amazon's Kindle Unlimited service. He's just not making any money, despite the immense popularity of his novel series. So it doesn't look like the next 45 books will be available there. He's been talking about finding some other way to distribute his eBooks, and no one cares. The fans of this series have made it clear to him that we'll pay whatever reasonable price he wants to charge, and that we'll go anywhere to purchase them.

Well Known URI's

As I teased at the top, this week's topic "Well Known URIs" was triggered by new support that's just been added to Chrome 86 for the so-called "Well Known URI Change Password" standard. To start, let's look at the description of Chrome's support. Under the topic of "Easier to change compromised passwords"...

> Starting with 86, Chrome's "Safety Check" supports the ".well-known/change-password" standard. This is a W3C standard that allows websites to specify the URL where users can go to change their passwords. Chrome 86 adding support for this standard means that users can press a button in the Chrome password settings screen and go directly to that page to change the password right away, rather than needing to search through a website's complicated structure.

That is just too cool. This is another of those perfect compromise solutions that provides a startling amount of benefit at very little cost.

One of the Wild West aspects of the World Wide Web has been that anyone could design a website pretty much anyway they chose. By convention, the root of a domain would contain a default landing page, but even that could be whatever the webmaster desired.

If it were only **people** who were ever visiting, that would be fine. But it wasn't long before it occurred to someone that "bots" could be created to mimic people's actions as they browsed a site... starting from that site's root page and following links just like a person might. This quickly became known as "spidering" since a bot was crawling around on the web.

The trouble was, the first bots were not very smart. They would sometimes get lost, do dumb things, started looping, issue dumb and expensive queries, or whatever. So the first divergence from the pure "anything goes" ad hoc design of websites was the agreement about the universal "/robots.txt" file. By universal agreement a text file named "robots.txt" would be placed on the root of any site which wished to control the actions of any bots that might enter. The robots.txt file would specify which areas a bot might go, and which regions were off limits to a bot. It even provided for bot-specific behavior by matching rules to a bot's User-Agent header. A bot's adherence to the robots.txt file was entirely optional. But well-behaved bots would, often for their own sake, abide by the file's requests.

So, the website's root "/" followed by the "/robots.txt" file were the first instances of "Well Known URIs." But the concept is so useful as we move more toward automation, that the World Wide Web Consortium (W3C) has formalized, standardized and hugely expanded upon this feature.

The work on this began 10 years ago, back in 2010, with RFC 5785 which was titled "Defining Well-Known Uniform Resource Identifiers (URIs)." It's Abstract was short and sweet:

> This memo defines a path prefix for "well-known locations", "/.well-known/", in selected Uniform Resource Identifier (URI) schemes.

Then, last May 2019 that RFC was formally obsoleted by RFC 8615. That RFC defines the function of all of this, as follows:

> Introduction
>
> Some applications on the Web require the discovery of information about an origin [RFC6454] (sometimes called "site-wide metadata") before making a request. For example, the Robots Exclusion Protocol (http://www.robotstxt.org) specifies a way for automated processes to obtain permission to access resources; likewise, the Platform for Privacy Preferences [P3P] tells user agents how to discover privacy policy before interacting with an origin server.
>
> While there are several ways to access per-resource metadata (e.g., HTTP header fields, PROPFIND in Web Distributed Authoring and Versioning (WebDAV) [RFC4918]), the perceived overhead (either in terms of client-perceived latency and/or deployment difficulties) associated with them often precludes their use in these scenarios.
>
> At the same time, it has become more popular to use HTTP as a substrate for non-Web protocols. Sometimes, such protocols need a way to locate one or more resources on a given host.
>
> When this happens, one solution is to designate a "well-known location" for data or services related to the origin overall, so that it can be easily located. However, this approach has the drawback of risking collisions, both with other such designated "well-known locations" and with resources that the origin has created (or wishes to create). Furthermore, defining well-known locations usurps the origin's control over its own URI space [RFC7320].

To address these uses, this memo reserves a path prefix in HTTP, HTTPS, WebSocket (WS), and Secure WebSocket (WSS) URIs for these "well-known locations", "/.well-known/". Future specifications that need to define a resource for such metadata can register their use to avoid collisions and minimise impingement upon origins' URI space.

So, the first thing to know is that IANA, the Internet Assigned Numbers Authority has formalized the registration, use and mean of all suffixes of that "/.well-known/" root... and a bunch of them, including "change-password" exist:

https://www.iana.org/assignments/well-known-uris/well-known-uris.xhtml

In the case of the "change-password" URI, which is a perfect example of this class of URIs for us to study, the W3C specifies that it does **not** directly return a site's password changing page. Instead, it returns a temporary URI redirect to the site's actual password changing page:

https://w3c.github.io/webappsec-change-password-url/

Client-side password management software helps improve both the security and usability of websites which require authentication. It improves security by reducing cross-site password reuse, and enhances usability by providing autofill functionality.

Sites currently lack a way to programmatically advertise where a user can change their password. By proposing a well-known URL for changing passwords, this specification enables password managers to help users change their passwords on sites which support it.
Servers should redirect HTTP requests for an origin's change password url to the actual page on which users may change their password by returning a response with a redirect status of 302, 303, or 307, and a Location header. Clients must handle such redirects when requesting a change password url.

Note: The above paragraph restricts servers to using temporary redirect codes. See Issue 13.

[Redirects must be temporary since some browser clients and proxies will cache permanent redirects and will not fetch the original URI when they have a local URI match. That would prevent websites from being able to change where this well-known URI points on their site.]

If necessary, servers may respond with an HTML document containing an http-equiv pragma directive in the refresh state. *[I hope not, that's an old-school kludge.]* Clients should handle such redirects when requesting a change password url.

Servers **must not** locate the actual change password page at the change password url.
See: RFC8615 §1.1 "Appropriate Use of Well-Known URIs."

Clients must handle ok status responses when requesting a change password url.

So the adoption of this would be very cool. Even in the absence of a password manager, a user's web browser could query any site being visited for an affirmative reply to "/.well-known/change-password" and then light up a little "Change your password here..." button in the

browser's UI to indicate that the browser knows where to take its user for that purpose. Or, if not on the browser UI, it would be perfect to place that at the top level of the browser's menu.

I love the concept of well-known, well-defined, IANA-registered URIs. It's a perfect example of something optional and easily implemented both by webmasters and clients, which can return a huge benefit to users. it's entirely foreseeable that a few years from now it will be standard practice for sites to help visitors maintain their security in this way.