

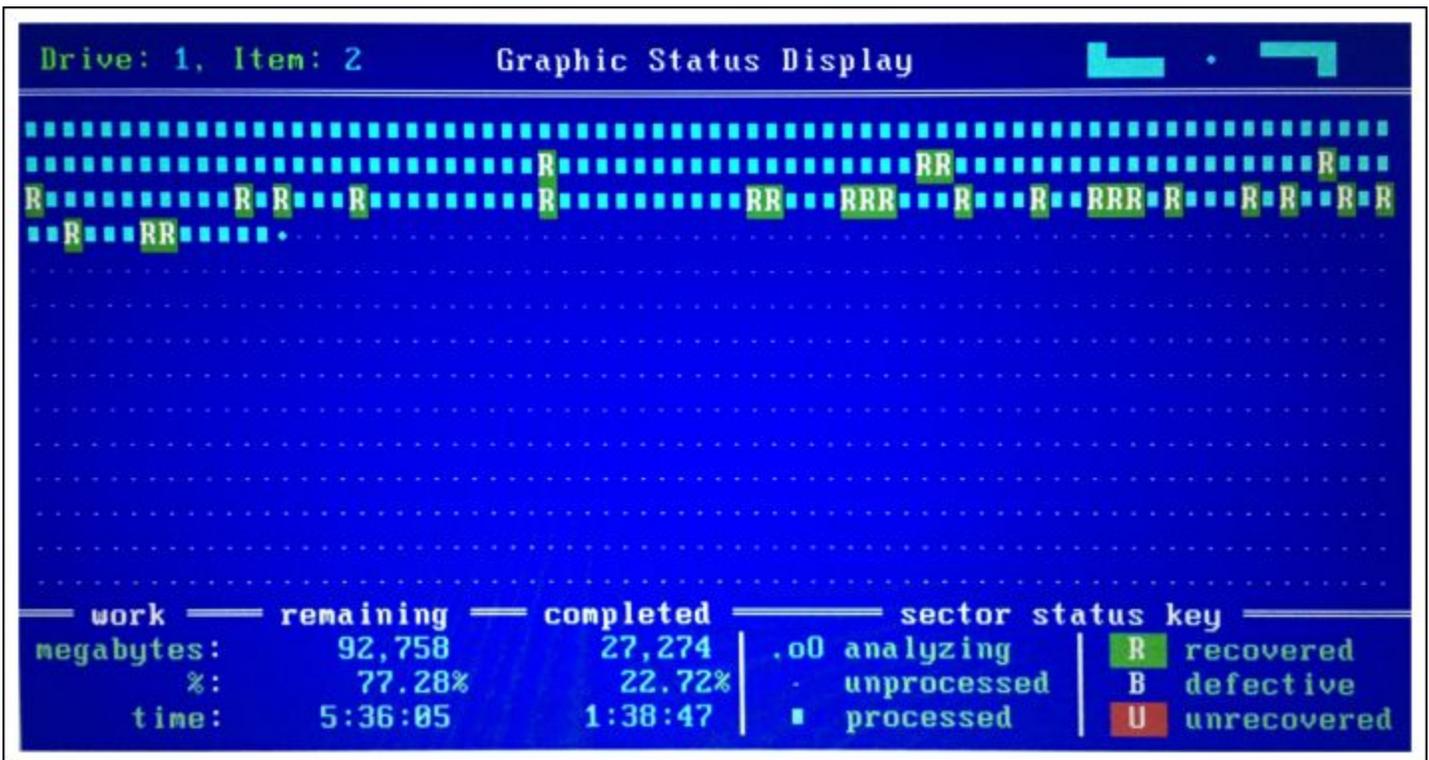
Security Now! #738 - 10-29-19

A Foregone Conclusion

This week on Security Now!

This week we look at another collision created by 3rd-party A/V, a powerful new Windows Defender feature that's easy to have missed, a public database breach by someone who should know better, what's worse than having all your files encrypted?, a VERY nice-looking fully-encrypted and free eMail service engineered in privacy-respecting Germany, stats coming back from Firefox's newly enhanced tracking privacy protection, a new and very bad remote code execution vulnerability affecting NGINX web servers and the planned introduction of RCS to replace SMS next year. We also have a piece of SQRL news and some miscellany. Then we look at the outcome of a recent appellate court decision which complicates the decision about whether using a password or a biometric is more "judgement proof."

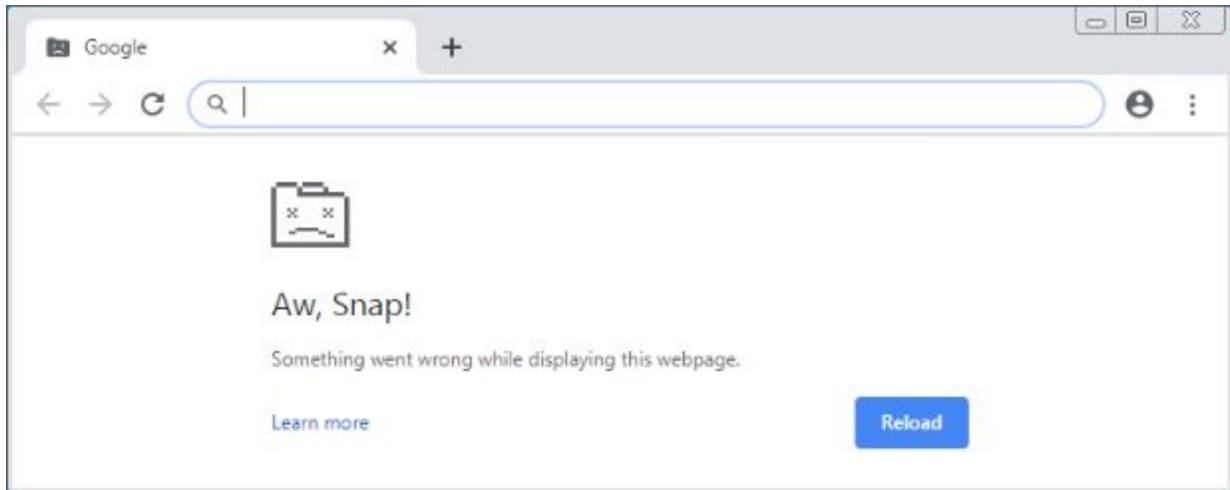
A Classic Example of SpinRite Success:



Security News

3rd-party A/V strikes again

Last Tuesday Chrome began updating itself to release 78... and in addition to its 37 welcome security fixes, its experimental support for increasingly controversial DoH (DNS over HTTPS) support, page tab mouse hover drop-downs, and some other goodies which will require manual enabling, a subset of users immediately reported receiving Chrome's equivalent of Windows' Blue Screen of Death. In the case of Chrome, it's the "Aw snap! Something went wrong while displaying this webpage." message...



Many people began reporting that they are experiencing the annoying Aw, Snap! error. Those who launched the browser after installing the update reported that they could no longer use Chrome at all.

The trouble was tracked down to a collision between Chrome and systems running an older version of the Symantec Endpoint Protection (SEP) security suite -- any version of SEP older than v14.2. As we know, Symantec Endpoint Protection is a popular enterprise security product, but it turns out that the outdated version has an incompatibility a newly added feature in the most recent release of Chrome. It turned out that the problem also affected the Chromium version of Microsoft's Edge.

<https://support.symantec.com/us/en/article.tech256047.html>

The collision was the result of Microsoft's new "Code Integrity" feature which checks the drivers and system files for signs of unauthorized tampering. This new feature was added in Windows 10 v1803 and it has since been rebranded WDAC, for "Windows Defender Application Control." With that rebranding can an extension of its "protection" to include all applications running on Windows, with the default being pretty much nothing. We've touched upon this in the past. It's a highly effective way to obtain a locked-down "whitelisted" system which will only run applications on the whitelist. But it's also somewhat of a pain in the butt since it turns out that Windows users wish to run all sorts of programs not signed by Microsoft.

One solution will be to update to the current release of Symantec's Endpoint Protection. All systems other than Server 2016 and Win10 RS1 can do that today. Those two Windows editions will need to wait until Tuesday, November 12th when Symantec will be fixing this.

In the mean time, or if upgrading SEP isn't convenient, a workaround is to run Chrome with "RenderingCodeIntegrity" disabled. There are two ways to do this:

1. Go to the Start menu and search for Google Chrome.
2. Right-click on the Chrome icon and click Properties.
3. Navigate to the Target field and paste the following command at the end:
-disable-features=RendererCodeIntegrity
4. Click the Apply button to save the changes.

This will append that option to Chrome's launch from that icon. I don't have the problem, so I wasn't able to determine whether launching Chrome from a system link when it's the default URL handler, would invoke this option, but I cannot see how it could.

So, for a more robust workaround, the registry can be edited:

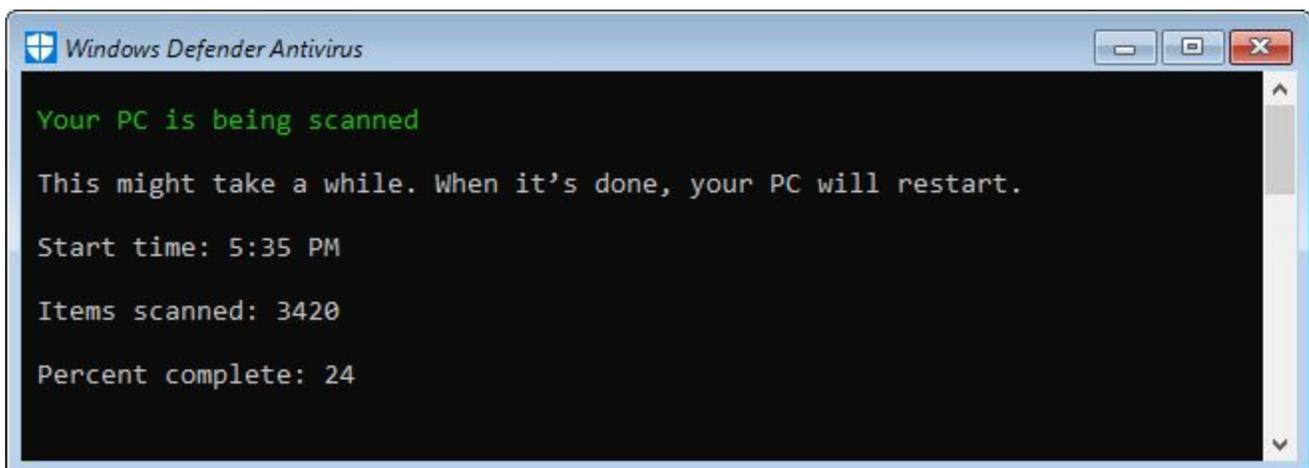
```
Key: HKLM\Software\Policies\Google\Chrome  
Name: RendererCodeIntegrityEnabled  
Type: DWORD (32-bit)  
Value: 0
```

Windows Defender Offline Scan

Speaking of A/V tools, I wanted to make sure that everyone was aware of Windows Defender's offline scanning mode. We've covered the insidious nature of rootkits...

<< recap about rootkits >>

The Defender in Win10 has the traditional Quick, Full & Custom scan modes. But it also has a new offline mode that can be used to sidestep the power of rootkits to find and remove even the most stubborn malware:



Clicking on "Scan Options" under Windows' "Virus and Threat Detection" will reveal a new additional option for an offline scan. When chosen, Windows will reboot into its Windows Recovery Environment, which is a small edition of Windows that lives in its own disk partition and is used for maintenance and recovery work.

Anyway... just an FYI of an interesting, useful and built-in tool that's available if anything should ever happen cause you to wonder whether something nasty might be hiding inside your machine, unseen and avoiding detection, as rootkits have proven themselves able to.

All Your Database Are Belong To Us

Even before I was skipping over the news of the ransomware attack du jour, I was skipping over news of large publicly-exposed databases. And, yes, I know, I've mentioned plenty of those in the past. But even so, you'd be amazed to know how many I have deliberately spared everyone from hearing about. So, today the bar is higher, but it's not infinitely high... And a major breach by a major player such as Adobe still seem worth of dishonorable mention.

Data hunter Bob Diachenko, some of whose previous discoveries we've covered in the past, discovered a wide open public and unsecured database belonging to Adobe Saturday before last, on October 19th. That database contained the email addresses and other Adobe-specific information of nearly 7.5 million customers of Adobe's Creative Cloud. It included:

1. Account creation date
2. Adobe products used
3. Subscription status
4. Whether the user is an Adobe employee
5. Member IDs
6. Country
7. Time since last login
8. Payment status

So that's the email addresses of approximately half of Creative Cloud's customer base. However, what was not exposed this time was any obviously compromising information such as their passwords or payment information. Nevertheless, it's EXACTLY this sort of "inside information" -- member IDs, products used, subscription status, etc. that can be leveraged to make phishing eMails FAR more believable and can therefore induce users to download and run malware, such as ransomware. Once upon a time there was less mischief that miscreants could get up to with this sort of data breach. Those times are long gone.

And the bigger question to ask here is, exactly how does this happen in an organization such as Adobe where security =MUST= be an out-in-front requirement? How does any database containing 7.5 million of Adobe's customers get placed -- by mistake -- onto the public Internet?

<https://theblog.adobe.com/security-update/>

In a "Security Update" dated last Friday, October 25th, Adobe posted:

At Adobe, we believe transparency with our customers is important. As such, we wanted to share a security update.

Late last week, Adobe became aware of a vulnerability related to work on one of our prototype environments. We promptly shut down the misconfigured environment, addressing the vulnerability.

The environment contained Creative Cloud customer information, including e-mail addresses, but did not include any passwords or financial information. This issue was not connected to, nor did it affect, the operation of any Adobe core products or services.

We are reviewing our development processes to help prevent a similar issue occurring in the future.

Should you have any questions, we encourage you to contact us at:

<https://helpx.adobe.com/contact.html>

Note that their "transparency" was not really optional since the news of this breach was widely published that same day.

To their credit, Adobe did take the exposed data offline immediately upon being notified of its exposure on October 19th. But they say that they are reviewing their development processes to help prevent similar issues in the future. One would have hoped that was done after their very similar October 2013 breach which impacted at least 38 million of their users, 3 million of whom had their encrypted credit cards and login credentials exposed.

"The Cloud" is extremely powerful and capable. But with great power comes great responsibility.

So I'll pose the rhetorical question...

Q: When would you wish that it *WAS* just ransomware?

And the answer is:

A: When the attackers who got into your network stole all of your data (rather than encrypting it) and are threatening to expose it to the world unless you pay 4 bitcoins.

That's what just happened to the city of Johannesburg, South Africa this past weekend, who spent the weekend struggling to recover from its second cyberattack this year as key affected (and infected) services and systems were taken offline.

The city first alerted users of the attack via Twitter on Thursday 24 October and since then the note left by the attackers, calling themselves the "Shadow Kill Hackers" reads as follows:

YOUR CITY HAS BEEN HACKED

Hello Joburg city! Here are Shadow Kill Hackers speaking. All of your servers and data have been hacked. We have dozens of backdoors inside your city.
We have control of everything in your city. We can shut off everything with a button. We also compromised all passwords and sensitive data, such as finance and personal population information.
Your city must pay us 4.0 Bitcoins (thats a very small amount of money) to the following address (19GUXmfkus3YCVNWcoHwgbJSqLusUNZakt) until October 28 17:00PM your time.
If you don't pay on time, we will upload the whole data available to anyone in the internet.
If you pay on time, we will destroy all the data we have, and we will send your IT a full report about how we hacked your systems and your security holes.
Contact us for more information at shadowkill@tutanota.com
Have a nice weekend. Shadow Kill Hackers Group.

I'll note that while the English used by this group suggests that they are not native English speakers, they do have good taste in secure encrypted eMail services. "Tutanova", a GDPR compliant, German-engineered, client-side encrypted eMail service with a useful free starter tier... does look very nice. They are extremely privacy-centric and ask for donation support.

Tutanova

So I took a hint from the "Shadow Kill Hackers" and checked-out Tutanova. I'm VERY impressed.

<https://tutanota.com/> <https://tutanota.com/community> <https://tutanota.com/pricing>

They say:

Join us in our fight for privacy! In the future Tutanota will be the privacy-respecting alternative for Google with a calendar, notes, cloud storage - everything encrypted by default! This is our dream of the future Internet and we are truly amazed by the feedback and the support we have received from you so far. We invite all of you to get in touch and to support our goal in bringing privacy and security to the world.

Secure email for everybody.

Tutanota is the world's most secure email service, easy to use and private by design. Sign up for free to take control of your mailbox.

Safe from Attackers

With end-to-end encryption and 2FA, your emails have never been more secure. The built-in encryption guarantees that your mailbox belongs to you: Nobody can decrypt or read your data.

We Love Open Source

Tutanota is open source so security experts can verify the code that protects your emails. Our Android app is Google-free making Tutanota the best open source email service.

On Your Favorite Device

With apps for iOS and Android, your secure emails are available anytime. Our fast web client & fully-featured apps make sure encrypted emails are a pleasant experience.

Free Secure Email without Ads

We take your email needs seriously by offering you a secure email service free from advertisements. We work around-the-clock to bring you an email service that lets you focus on the important things. From our clear and minimalist design to whitelabel customizations, from our free version for personal use to our fully-featured secure business email for companies, we are committed to making sure Tutanota is the only email service you will ever need.

We Encrypt Everything

We guarantee that your data in Tutanota is always encrypted - whether you use our secure webmail client, our apps for Android and iOS, or our desktop clients for Windows, Linux and Mac OS. Easily send a secure email with the knowledge that Tutanota encrypts subject, body and all attachments for you. Import all your contacts into Tutanota's encrypted address book, and rest assured that nobody else can access your contacts' personal information.

Anonymous Email: Privacy Is a Human Right

Everybody has the right to privacy and security. That's why the basic secure email account in Tutanota will always be free with all security features included. No personal information such as phone numbers is required to access your anonymous email account. We do not log IP addresses and strip IP addresses from emails sent and received. Your right to privacy is at the heart of Tutanota.

Proudly Engineered in Germany

Our entire team is based in Hanover, Germany. All your free encrypted emails are stored on our own servers in highly secured data centers in Germany. With its strict data protection laws and the GDPR, Germany has some of the best laws in the world to protect your secure emails. The German constitution stresses the importance of freedom of expression and of the human right to privacy.

It's hard to argue with those principles, which I know that many of our listeners here share. So I wanted to put Tutanova.com on everyone's radar. (And I guess we can thank the Shadow Kill Hackers for bringing Tutanova to our attention. :)

Firefox Privacy Protection

For just little old boring "I never surf anywhere exciting" me, Firefox has blocked 3,080 web trackers for me since September 4th, 2019.

But Mozilla just released figures to demonstrate Firefox's anti-tracking effectiveness: Their stats show that Firefox blocked 450 billion cross-site tracking requests since 2 July, shortly after enhanced tracking protection was first introduced. And since that time the rate of tracking protection blocking has risen to 10 billion blocks per day.

So... What do you think Jason? Time to perhaps give Firefox another look? You know that Leo has, right? He's completely switched back to Firefox.

I'll note, also that Firefox is continuing to move forward with their "Lockwise" password manager facility. Lockwise can now generate a secure password when signing up for a new account. This can also be used to replace a current weak one with a new and more secure one. Mozilla notes

that access to local browser-contained Lockwise database can be protected using Apple's FaceID or Android's TouchID biometric recognition systems.

And it appears that encryption used with Lockwise was done correctly:

- AES-256-GCM encryption, a tamper-resistant block cipher technology.
- onepw protocol to sign into Firefox accounts and obtain encryption keys.
- PBKDF2 and HKDF with SHA-256 to create the encryption key from your Firefox account's username and password.

My problem is that I'm just not browser monogamous. I'm a bit fickle when it comes to browsers. Some things cannot be beat under Firefox, like managing a large number of open tabs with the tabs sidebar. But Chrome does have its place, and I'm occasionally faced with IE or Edge. So having a polygamous password manager, such as Lastpass, allows me to keep everyone in one place, regardless of which browser I'm in the mood to use at any moment.

Bad new PHP/NGINX RCE being exploited in the wild

Saturday, a new and dangerous PHP flaw surfaced. The very short version is: If your site uses the very popular NGINX webserver, and PHP with the high-performance PHP-FPM feature enabled for optimum performance, you'll want to IMMEDIATELY (if not sooner) update to the latest versions of PHP, v7.3.11 and PHP v7.2.24. Why? ... Because your site is probably susceptible to a serious remote code execution vulnerability.

Okay, so what's going on?

PHP-FPM is the "FastCGI Process Manager." It's an alternative means for sites to invoke the PHP language interpreter. It brings increased efficiency through reduced per-instantiation overhead, which helps to keep busier PHP-based websites from becoming bogged down by PHP-invocation overhead.

The problem is, the NGINX version of this module contains a newly disclosed vulnerability that could allow unauthorized attackers to remotely hack the site's server.

More specifically, the vulnerability, tracked as CVE-2019-11043, affects websites with certain configurations of PHP-FPM that are common in the wild as demonstrated by a proof-of-concept (PoC) exploit for the flaw which has been released publicly.

The primary vulnerability is an "env_path_info" underflow memory corruption in the PHP-FPM module, and, when chained together with other issues, could allow attackers to remotely execute arbitrary code on vulnerable web servers.

The vulnerability was spotted by Andrew Danau, a security researcher at Wallarm while hunting for bugs in a Capture The Flag competition, which was then weaponized by two of his fellow researchers, Omar Ganiev and Emil Lerner, who developed a fully working remote code execution exploit.

Though the publicly released PoC exploit is designed to specifically target vulnerable servers running PHP 7+ versions, the PHP-FPM underflow bug also affects earlier PHP versions and could be weaponized in a different way.

In brief, a website is vulnerable, if:

- NGINX is configured to forward PHP pages requests to PHP-FPM processor,
- `fastcgi_split_path_info` directive is present in the configuration and includes a regular expression beginning with a '^' symbol and ending with a '\$' symbol,
- `PATH_INFO` variable is defined with `fastcgi_param` directive,
- There are no checks like `try_files $uri =404` or `if (-f $uri)` to determine whether a file exists or not.

This vulnerable NGINX and PHP-FPM configuration looks like the following example:

NGINX and PHP-FPM configuration looks like the following example:

```
location ~ [^/]\.php(/|$) {
    ...
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    fastcgi_param PATH_INFO      $fastcgi_path_info;
    fastcgi_pass    php:9000;
    ...
}
```

Here, the `fastcgi_split_path_info` directive is used to split the URL of PHP web pages into two parts. The regular expression, which defines the `fastcgi_split_path_info` directive, as shown, can be manipulated by using the newline character in a way that the split function eventually sets the path info empty.

Next, since there is an arithmetic pointer in FPM code that incorrectly assumes that `env_path_info` has a prefix equal to the path to the php script without actually verifying the existence of the file on the server, the issue can be exploited by an attacker to overwrite data in memory by requesting specially crafted URLs of the targeted websites.

While this particular configuration might seem unlikely to exist widely in the wild, this list of preconditions required for successful exploitation turns out to be a very commonly used pattern. It is widely used by web hosting providers and is available on the Internet as part of many PHP FPM tutorials.

For example, one affected web hosting provider is Nextcloud who released an [advisory](#) yesterday warning its users that "the default Nextcloud NGINX configuration is vulnerable to this attack." It recommends system administrators to take immediate actions.

A Patch for this vulnerability was released just last Friday, nearly a month after researchers reported it to the PHP developer team. Since the PoC exploit is already available, and the patch released just Friday, it's likely possible that hackers may have already started scanning the Internet in search for vulnerable websites. Don't be one of them!

<https://lab.wallarm.com/php-remote-code-execution-0-day-discovered-in-real-world-ctf-exercise/>

Goodbye SMS (maybe kinda) Hello RCS?

Next year, in 2020, all of the major US mobile phone carriers, including AT&T, Verizon, T-Mobile, and Sprint will finally be joining forces to launch an initiative to SMS with the long-awaited RCS mobile messaging standard.

We potentially (and hopefully) care because the nationwide switch over to RCS would serve to raise the incredibly low lowest common denominator for cellular content exchange well above where it currently sits with SMS.

The initiative is also working with its carrier ownership group and other companies to develop and deploy the new RCS standard in a new text messaging app for Android phones that is expected to be launched in 2020. However, because it got tired of waiting, earlier this year, Google independently released RCS messaging for Android smartphones in two countries, the United Kingdom and France. So that effort will presumably be integrated into the formal RCS if and when it actually does happen.

So, the goal of this joint venture, which is called the Cross Carrier Messaging Initiative (CCMI), is to finally get the four major cellular carriers off their butts to deliver GSMA's Rich Communications Service (RCS) industry standard to consumers and businesses, both in the United States and globally.

Although the RCS standard was developed more than a decade ago, it has never been adopted widely due to the quagmire of mobile carrier and phone maker politics; as an example, Apple has no interest in RCS because it's already offering more than that through iMessages... and iMessage is notoriously end-to-end encrypted, which RCS it not and cannot be.

A few carriers and services have struck out on their own to offer non-universal versions of the new messaging standard which limits the exchange of RCS-based messages to the subscribers of their own networks... so that hasn't been very useful.

So what is RCS??

Unlike our now-ancient SMS technology, RCS-based enhanced text message service supports high-resolution photo sharing, location sharing, group messaging, animated stickers, read receipts, and some other features which are all reminiscent of Apple's iMessage. So in that sense it does significantly raise the base-level lowest-common-denominator messaging functionality that comes installed on phones by default. And we should remember that there is large lower-tier "feature phone" market lying beneath the "smartphone" market. Those will likely be the greatest feature recipients of this change.

However, unlike iMessage, Signal, WhatsApp, Threema, etc... as I noted, RCS-based messages are not end-to-end encrypted. They do, however, contain message verification and brand certification mechanisms (whatever that is) to ensure that users are interacting with legitimate brands to protect them from fraudulent accounts, impersonators, or phishing.

Also in RCS's favor is that whereas SMS communicates over SMPP using the insecure SS7 protocols, all RCS traffic between the device and the network can be protected using SIP over TLS encryption. so we get endpoint authentication and communications privacy through encryption.

The news reporting of this notes that the CCMi project has not yet fully developed its RCS-based messaging standard (really? after more than a decade?), so it's not yet clear, at this moment, if the major U.S. carriers will implement protections for their users' privacy from government surveillance. But I'm sure that many of us would be willing to bet that won't happen at this late stage of the US government's campaign against absolute privacy within its borders.

However, the announcement does say that it will "enable an enhanced experience to privately send individual or group chats across carriers with high-quality pictures and videos."

"Enable an enhanced experience." Hmm. That sure sounds like corporate-speak written by a PR flack who was also unsure exactly what the system would do. It sure better "enable an enhanced experience" or what have we been waiting for?

While not inspiring much confidence, someone wrote something for Sprint's CEO Michel Combes to put his name on. It reads: "The CCMi will bring a consistent, engaging experience that makes it easy for consumers and businesses to interact in an environment they can trust. As we have seen in Asia, messaging is poised to become the next significant digital platform. CCMi will make it easy for consumers to navigate their lives from a smartphone." said Sprint CEO Michel Combes."

Uhhh... hello?? Can you hear me now?? We've all been doing that for the past decade, using smartphones to navigate our lives! Thanks for coming.

And, not to be left out, Verizon's Consumer Group CEO Ronan Dunne put his name on the following: "At Verizon, our customers depend on reliable text messaging to easily connect them to the people they care about most. Yet, we can deliver even more working together as an industry. CCMi will create the foundation for an innovative digital platform that not only connects consumers with friends and family but also offers a seamless experience for consumers to connect with businesses in a compelling and trusted environment."

Uhhh... okay, that's called "The Internet." Wow.

So... certainly, from a security standpoint, putting as much distance between us and SMS as possible is "A Good Thing." And even if we don't have end-to-end encryption, if there is on-the-wire encryption and endpoint authentication added to a universally-available low-level messaging platform to replace SMS for things like one-time-passcode loops, that would still be way better than the pathetic SMS system that we have today. It seems very clear that it won't be a replacement for any of our truly secure end-to-end smartphone-based messaging services.

But raising the lowest common denominator is never a bad thing.

Maybe it'll happen.

SQL News

<https://www.sqlfor.net/>

"SQLForNet" is middleware for implementing SQL in your ASP.net Core solution. It offers a fully protocol-complaint server-side solution with a high degree of customization. Our goal is to enable any and all ASP.net Core sites to enable SQL login for their visitors.

Miscellany

Tim Kington @TimKington

You made a serendipitous mistake during the podcast this week, and said "securious". I loved it! I think it should be the byline for the show: "Security Now - the podcast for the securious."

Dr.Flav's posting in grc's Security Now newsgroup:

The official checkra1n domain is now live with a valid cert (SSL Labs grade A)

<https://checkra1n.com> They have wisely registered other possible squatting options

<https://checkra.in>, <https://checkra1n.io>, <https://checkra1n.dev>, <https://checkra1n.net>

I gave the scam site an appropriate review in WOT though it will need more reviews before triggering any warnings in the reputation systems.

<https://www.mywot.com/en/scorecard/checkrain.com>

Forced Password Disclosure

The subtitle for this discussion should be "Supreme Court... Where art thou?"

I wanted to give a new Oregon state appellate court decision, which addresses the question of whether forcing the disclosure of a memorized device-unlock password, is, in fact, tantamount to compelling self-incriminating testimony against one's interest, which, thanks to the 5th amendment to the US Constitution, is a protected right.

The privilege against compelled self-incrimination is defined as: "The constitutional right of a person to refuse to answer questions or otherwise give testimony against himself."

We have many times discussed this issue, since, at this point in the development of the consumer product industry we broadly have two common ways of unlocking our devices: As we discussed last week, "Biometrics" (something you are) and the more traditional password, passcode or swipe sequence, all which can be lumped under a secret which is "something you know."

In the US we have this problem that where there is no ultra-clear constitutional black-letter law, nor any more recent ruling by the ultimate (thus supreme) court of the land, lesser courts and judges are left with the freedom and burden of attempting to interpret what has come before in order to create a foundation and context for new cases which are brought before them.

So this has all boiled down to: Are we safer against forced device unlock using a password or a biometric? The upshot, so far, has been that people who have memorized their passwords are safer because they are protected by the 5th amendment against being forced or coerced in any way to testify against themselves by divulging their device's unlock password.

The problem is... this is not something that our US Supreme Court has yet ruled on, so the lower courts are flip-flopping around. And another flip-flop happened last week...

An Oregon appeals court last Wednesday decided that a woman whose judgement was impaired by methamphetamine when she crashed into a tree seriously injuring one adult and five children passengers, **can** be forced to unlock her iPhone with something she has committed to memory.

The appellate court's decision states that, in their opinion, which is their ruling on the matter, it is **NOT** a violation of her Fifth Amendment rights against self-incrimination...

... because the fact that she knows her phone passcode is "A Foregone Conclusion."

So let's step back a bit: A 27-year-old woman who crashed her car into a tree in Salem, Oregon -- injuring herself, her friend and the five children in the back seat -- did not want to give police any help in building a criminal case against her.

Police wanted to search the contents of an iPhone they found in Catrice Pittman's purse, but she never confirmed whether it was hers and wasn't offering up a passcode. Her defense attorney argued that forcing her to do so would violate her rights against self-incrimination under the

Fifth Amendment of the U.S. Constitution and Article 1 Section 12 of the Oregon Constitution.

But a Marion County judge sided with police and prosecutors by ordering Pittman to enter her passcode. The ruling was appealed. Then, Wednesday, the Oregon Court of Appeals agreed with that ruling -- **in a first-of-its-kind opinion for an appeals court in Oregon.**

The ruling likely will make it easier for Oregon police to compel access to the contents of suspects' cellphones -- and all of the massive amount of personal information they contain, including photos, videos, past Internet searches, texts and phone contact lists of everyone that the phone's owner has contacted, and when.

Sarah Laidlaw, the deputy public defender who represented Pittman on appeal said: "The upshot of this case is when police have a warrant to get information off a cellphone and the government knows the phone is yours and you have the passcode, then the court **can** compel you to enter the passcode."

Ryan Scott, a criminal defense attorney in Portland who is not associated with Pittman's case, but closely follows appeals cases, said the ruling is an example of a continuing erosion of rights. Ryan said: "Our rights are a little less than they were yesterday. But for those of us following this area of law it's not a surprise." He said that federal case law has been leaning in this direction for the past few years.

Scott said the ruling won't affect many Oregon defendants whose phones are seized by police because police already have technology that allows them to crack into most of those phones. But sometimes, as apparently with Pittman's phone, police can't get in. Scott said that the latest iPhones, more often than other phones, have proven difficult. Scott volunteered that, from his experience, for people who want their information private, he would recommend getting an iPhone. And he added that "Apple is not paying me to say that."

Pittman had argued through her attorneys that punching in her passcode would amount to "testifying" against herself because doing so would effectively admit that the iPhone was hers and she knew the passcode. But the Court of Appeals ruled that because police already had good reason to believe the phone was hers given its location in her purse, the fact that she knew its passcode was already a "foregone conclusion." In other words, **she could be compelled to cooperate as an exception to her constitutional rights.**

In doing some additional digging around, I saw that this odd-seeming "foregone conclusion" standard has been coming up a lot recently in these compelled-unlocking cases. It allows prosecutors to bypass Fifth Amendment protections if the government can show that it knows that the defendant knows the passcode to unlock a device. If I ever needed another reason to be a coder rather than an attorney, this is it, since this makes my head hurt.

The reason we need the Supreme Court is that, as we have covered in the past, not all courts, nor even all appellate courts, have ruled in this direction.

One example is the decision that came out of the Florida Court of Appeal in November 2018 regarding a case that's similar to that of Pittman. It involved an intoxicated person who crashed his car, leading to the injury or death of passengers, then refused to unlock his iPhone for police.

In Florida, the court refused a request from police that they be allowed to compel an underage driver to provide the passcode for his iPhone because of the “contents of his mind” argument Fifth Amendment protection.

But the Florida court went beyond that, saying that whereas the government in the past has only had to show that the defendant knows their passcode, with the evolution of encryption, the government needed to show that it also knew that specific evidence needed to prosecute the case **was** on the device – not just that there was a reasonable certainty the device could be unlocked by the person targeted by the order.

If prosecutors already knew what was on the phone, and that it was the evidence needed to prosecute the case, they didn’t prove it, the Florida court said at the time. From the order to quash the passcode request:

Because the state did not show, with any particularity, knowledge of the evidence within the phone, the trial court could not find that the contents of the phone were already known to the state and thus within the “foregone conclusion” exception.

Regardless of the “foregone conclusion” standard, producing a passcode is testimonial and has the potential to harm the defendant, just like any other Fifth Amendment violation would, the Florida court said. It’s not as if the passcode itself does anything for the government. What it’s really after is what lies beyond that passcode: information it can use as evidence against the defendant who’s being compelled to produce it.

So, the Oregon decision doesn’t change the way all courts will henceforth interpret these gadget-unlock, Fifth Amendment cases -- because opinions wrapped in judgements are still just that: opinions. What it does is create yet another decision to which prosecutors and courts can refer when arguing and deciding future cases.

It appears that we need to be a little less glib about the biometrics vs passcode debate. Until the US Supreme court finally rules on this passcodes are probably the safer bet since rulings have been coming down on both sides of the question.

