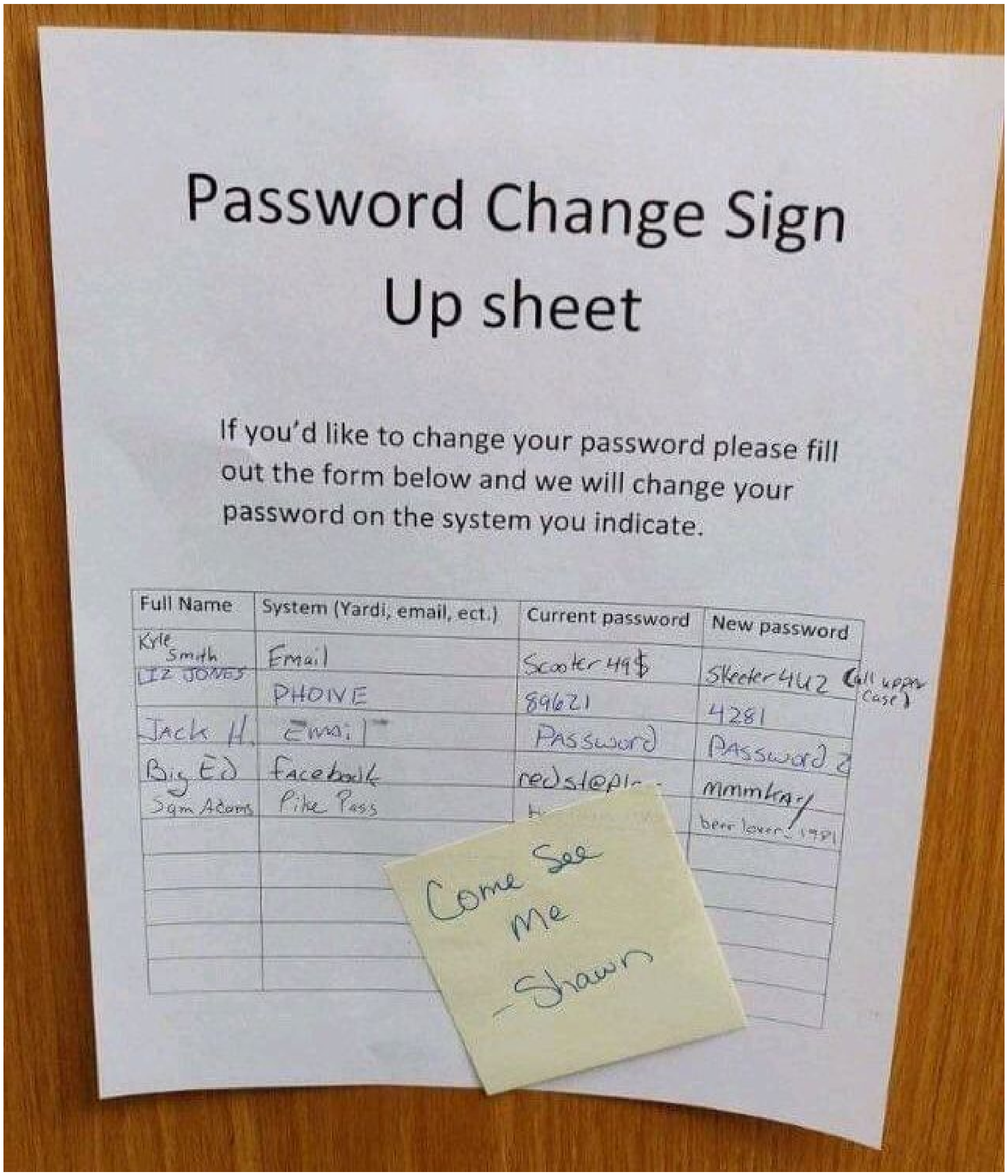
Security Now! #711 - 04-23-19
DNSpionage

This week on Security Now!

This week we discuss Google's use of their SensorVault tracking to assist law enforcement, time to update Drupal again... and speaking of "again": Facebook. We also look at Russia's newly approved legislation moving toward an Internet "off switch", a reminder that "USB Killers" are a real thing, the news of Marcus Hutchins' plea deal, an actively exploited Windows 0-day, a bunch of Microsoft Edge news, the Win7 end-of-life notices, something from the "I did say this was bound to happen" department, some miscellaneous news, and then we examine the latest detailed threat research from Cisco's Talos group about the leveraging of DNSpionage.



Password Change Sign Up sheet

If you'd like to change your password please fill out the form below and we will change your password on the system you indicate.

Full Name	System (Yardi, email, ect.)	Current password	New password
Kyle Smith	Email	Scooter49$	Skeeter442 (all upper case)
LIZ JONES	PHONE	89621	4281
Jack H.	Email	PASSword	PASSword2
Big ED	Facebook	redst@pl-	mmmkay
Sgm Adams	Pike Pass	h	beer lover! 1981

Come See Me
-Shawn

Security News

Google uses its "SensorVault" to help catch the bad guys
So we know that Google tracks us everywhere, even when we have Google's Location History feature disabled. Last August we talked about the fact that many of Google's apps, when running on either Android or iOS, continually monitor their users' location. Apps such as Maps or the weather update service on Android continuously continuously precise latitude and longitude. Back then we talked about how the movements of a Princeton professor were continuously tracked even while "Location History" was disabled. In response to the Associated Press investigation, Googled responded: "There are a number of different ways that Google may use location to improve people's experience, including Location History, Web, and App Activity, and through device-level Location Services. We provide clear descriptions of these tools, and robust controls so people can turn them on or off, and delete their histories at any time." And we'll recall that it's actually quite involved to do so. Google explains that it uses location tracking features to improve its users' experience, like "personalized maps, recommendations based on places you've visited, help finding your phone, real-time traffic updates about your commute, and more useful ads."

An interesting feature of this which has recently been receiving more attention recently is that Google may also share its users' location data with federal authorities who are conducting criminal investigations when asked to do so with a warrant. The system works the way we would have designed it if asked:

Law enforcement first needs obtain a "geofence" warrant.

The authorities then reach out to Google, armed with that warrant, for the purpose of learning about smartphones that were in the area of a crime at the time of the crime.

After receiving the warrant, Google queries their massive "SensorVault" database to gather 1st pass "all possible phones" location information and forwards that to investigators. For this 1st pass, each device is identified by an anonymous ID code, not the identity of the device.

Investigators review the data, look for patterns of the devices near the crime scene, and then request additional location data about specific devices that appear to be relevant. This allows them to see the particular device movement beyond the original area defined in the warrant.

As investigators narrow down their search to a few devices, which they have strong reason to believe may be useful for providing information crucial to the case as either suspects or witnesses, Google then reveals the real name, email address and other data associated with the devices.

The system is not perfect. It has resulted in false arrests. But so do human witnesses. And overall this is being used more and more often by law enforcement in the resolution of crimes.

Time to update Drupal to close a pair of Moderately Critical vulnerabilities.
https://www.drupal.org/security

To get some sense of perspective, let's look back over just the past 12 months...

2018-April-18 / Drupal core - MODERATELY CRITICAL - Cross Site Scripting
CKEditor, a third-party JavaScript library included in Drupal core, has fixed a cross-site scripting (XSS) vulnerability. The vulnerability stemmed from the fact that it was possible to execute XSS inside CKEditor when using the image2 plugin (which Drupal 8 core also uses).

2018-April-25 / Drupal core - HIGHLY CRITICAL - Remote Code Execution
A remote code execution vulnerability exists within multiple subsystems of Drupal 7.x and 8.x. This potentially allows attackers to exploit multiple attack vectors on a Drupal site, which could result in the site being compromised. This vulnerability is related to Drupal core - Highly critical - Remote Code Execution - SA-CORE-2018-002. Both SA-CORE-2018-002 and this vulnerability are being exploited in the wild.

2018-Aug-1 / Drupal Core - 3rd-party libraries -SA-CORE-2018-005

2018-October-17 / Drupal Core - Multiple Vulnerabilities - SA-CORE-2018-006

2019-January-16 / Drupal core - CRITICAL - Third Party Libraries - SA-CORE-2019-001
Drupal core uses the third-party PEAR Archive_Tar library. This library has released a security update which impacts some Drupal configurations. Refer to CVE-2018-1000888 for details.

2019-January-16 / Drupal core - CRITICAL - Arbitrary PHP code execution - SA-CORE-2019-002
A remote code execution vulnerability exists in PHP's built-in phar stream wrapper when performing file operations on an untrusted phar:// URI. Some Drupal code (core, contrib, and custom) may be performing file operations on insufficiently validated user input, thereby being exposed to this vulnerability. This vulnerability is mitigated by the fact that such code paths typically require access to an administrative permission or an atypical configuration.

2019-February-20 / Drupal core - HIGHLY CRITICAL - Remote Code Execution
Some field types do not properly sanitize data from non-form sources. This can lead to arbitrary PHP code execution in some cases.

2019-March-20 / Drupal core - MODERATELY CRITICAL - Cross Site Scripting
Under certain circumstances the File module/subsystem allows a malicious user to upload a file that can trigger a cross-site scripting (XSS) vulnerability.

2019-April-17 / Drupal core - MODERATELY CRITICAL - Cross Site Scripting
The jQuery project released version 3.4.0, and as part of that, disclosed a security vulnerability that affects all prior versions. As described in their release notes:
jQuery 3.4.0 includes a fix for some unintended behavior when using jQuery.extend(true, {}, ...). If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype. This fix is included in jQuery 3.4.0, but patch diffs exist to patch previous jQuery versions.

2019-April-17 / Drupal core - MODERATELY CRITICAL - Multiple Vulnerabilities
Validation messages were not escaped when using the form theme of the PHP templating engine which, when validation messages may contain user input, could result in an XSS.

So that's 10 significant vulnerabilities in 12 months. We can no longer count this as extreme by today's measures. After all, Microsoft just patched 72 vulnerabilities in one month and two 0-days in each of the past three months. But those vulnerabilities do cover quite a lot of code real estate for Microsoft.

What seems abundantly clear is that creating secure means for keeping code up to date is crucial for any widely deployed software system.

Facebook, again...
Two weeks ago I shared the astonishing (and really almost unbelievable) news that Facebook had been popping up interstitial notices requiring users to turn over their eMail account PASSWORDS as a means of verifying them.

Rather than eMailing a nonce in a link to their eMail account and asking the user to please click on it, FaceBook was actually asking for their password. The ONLY THING Facebook could do with such a password is to use it to authenticate to and sign into their account.

So, at this point, you would have to imagine that it could not possibly get any worse, right? Wrong!

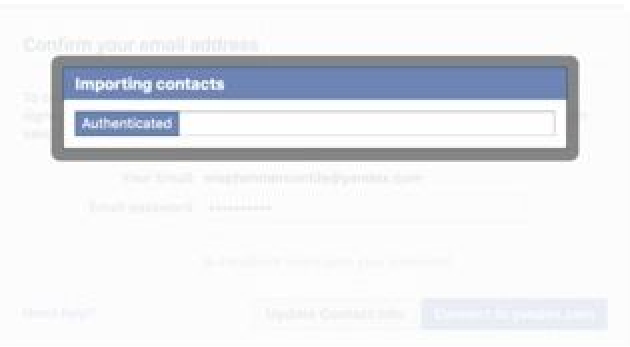
It turns out that FaceBook WAS in fact logging onto those eMail accounts... And not only that, they were then downloading and storing all of the user's contact information without their permission.

https://www.businessinsider.in/Facebook-says-it-unintentionally-uploaded-1-5-million-peoples-email-contacts-without-their-consent/articleshow/68930320.cms

For Business Insider, last Thursday, under the headline "Facebook says it 'unintentionally uploaded' 1.5 million people's email contacts without their consent" in exclusive reporting, Rob Price wrote: Since May 2016, the social-networking company has collected the contact lists of 1.5 million users new to the social network. The Silicon Valley company said the contact data was "unintentionally uploaded to Facebook," and it is now deleting them.

The revelation comes after pseudononymous security researcher e-sushi noticed that Facebook was asking some users to enter their email passwords when they signed up for new accounts to verify their identities, a move widely condemned by security experts. Business Insider then discovered that if you entered your email password, a message popped up saying it was "importing" your contacts without asking for permission first.

At the time, it wasn't clear what was happening — but Wednesday, Facebook disclosed to Business Insider that 1.5 million people's contacts were collected this way and fed into Facebook's systems, where they were used to improve Facebook's ad targeting, build Facebook's web of social connections, and recommend friends to add.



A Facebook spokesperson said before May 2016, it offered an option to verify a user's account using their email password and [then] voluntarily upload their contacts at the same time. However, they said, the company changed the feature, and the text informing users that their contacts would be uploaded was deleted — but the underlying functionality was not.

Facebook didn't access the content of users' emails, the spokesperson added. But users' contacts can still be highly sensitive data — revealing who people are communicating with and connect to.

While 1.5 million people's contact books were directly harvested by Facebook, the total number of people whose contact information was improperly obtained by Facebook may well be in the dozens or even hundreds of millions, as people sometimes have hundreds of contacts stored on their email accounts. The spokesperson could not provide a figure for the total number of contacts obtained this way.

Note also that the contact downloads are essentially the raw material for the referential database. Once that raw material has been downloaded and "absorbed" by facebook, it CAN be freely deleted without any loss. So Facebook is not saying that they are deleting all of the FRUITS of that ill gotten information, only the raw information itself.

A Facebook spokesperson said in a statement: "Last month we stopped offering email password verification as an option for people verifying their account when signing up for Facebook for the first time. When we looked into the steps people were going through to verify their accounts we found that in some cases people's email contacts were also unintentionally uploaded to Facebook when they created their account."

Facebook has said it didn't store the passwords. Okay. Not that it matters after they've sucked all of the accounts contact info. But in yet another Facebook privacy blunder which came to light last month, the company confirmed that it improperly stored hundreds of millions of user passwords in plain text rather than as hashes. At the time Facebook said that this plaintext password storage error affected hundreds of millions of Facebook Lite users, tens of millions of other Facebook users, and tens of thousands of Instagram users.

That Facebook disclosure was just updated last Thursday to say the number of affected Instagram accounts was much higher. Thursday's update said: "Since this post was published, we discovered additional logs of Instagram passwords being stored in a readable format. We now estimate that this issue impacted millions of Instagram users. We will be notifying these users as we did the others. Our investigation has determined that these stored passwords were not internally abused or improperly accessed." (How could they POSSIBLY make such an assertion after having <quote> "discovered additional logs of Instagram passwords being stored in a readable format"?? It's very CLEARLY a TOTAL and UTTER unorganized disaster over there.

And, last month, Mark Zuckerberg said he planned to rebrand the site he founded as a privacy service.

Yeah... Good luck with that. Perhaps set up an entirely new facility and rewrite the ENTIRE thing from scratch as a true privacy-centric service. What exists now is clearly beyond salvation.

Russia moves closer to adopting "Internet Master Cutoff Switch" legislation
Russia's lower chamber of parliament has backed a bill which privacy advocates fear could lead to the creation of a censorship system similar to China's Great Firewall. (Yeah, no kidding.)

The Associated Press reported last Tuesday that the State Duma, which is the lower house of the Federal Assembly of Russia, has advocated for the bill overwhelmingly.

The new regulations, if also accepted by the upper chamber -- which belongs to the Federal Assembly -- and signed into law by President Vladimir Putin, would require Internet Service Providers (ISPs) to route Russian Internet traffic locally through the country.

This would give Russian authorities the opportunity to use equipment and software to establish man-in-the-middle communication eavesdropping, as well as to block and censor global content that Russia does not want its citizens to be able to access.

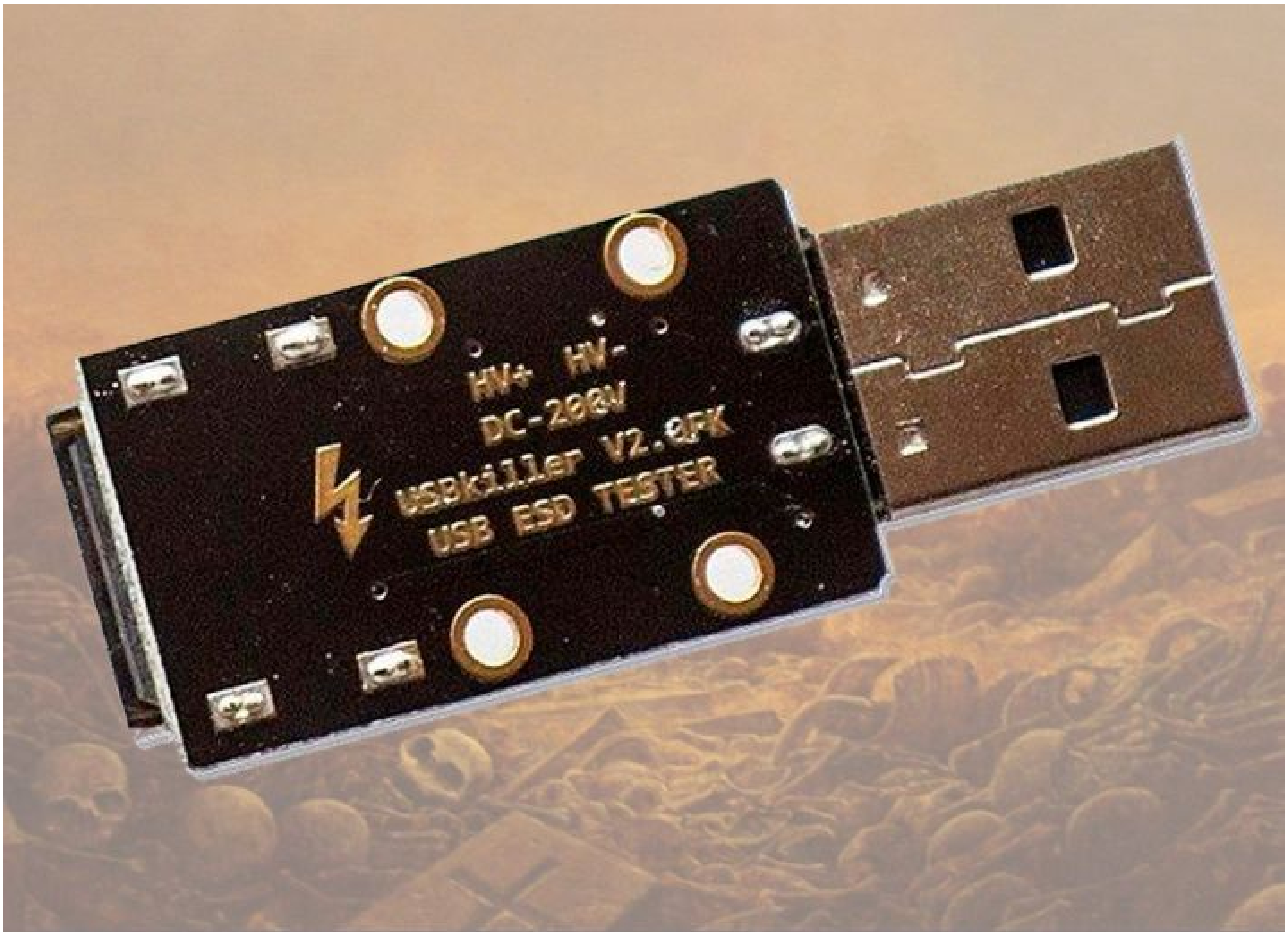
Although the Russian government has said that it will bear the cost and will reimburse ISPs, the country's ISPs would be required to provide equipment to exchange points approved by Russia's telecoms watchdog, Roskomnazor. And a localized Domain Name System (DNS) would be prepared to support the localization of content. Ruskynet.

Advocates of the bill claim this would be a protective measure only to protect the Internet in the country should a hostile entity cut off access -- as well as a means to insulate Russian traffic from potential cyberattacks by foreign entities by removing traffic rerouting outside of local systems.

However, others believe that a top-level control mechanism of this power would give Russian lawmakers the overarching authority to control the web in the country, as well as monitor its citizens' online habits.

And... Last time we talked about this our listeners reminded us that the full unfiltered Internet is in orbit above us all the time. So no one is strictly limited to what land lines or short-range WiFi can carry.

Just a reminder that the "USBKiller" is a real thing…



ZDNet carried the story of a 27 year old Indian national who graduated two years ago with an MBA from the College of St. Rose in New York. He has just changed his plea from "not guilty" to "guilty" ... And it occurred to me that the plea change might have something to do with the videos he made of himself killing the college's computers, which the prosecutors got their hand on.

The incident took place on February 14, according to court documents obtained by ZDNet. He recorded filmed himself while destroying some of the computers.

"I'm going to kill this guy," "it's dead," and "it's gone. Boom," he said on recordings obtained by the prosecution.

The suspect destroyed 59 computers, but also seven computer monitors and computer-enhanced podiums that had open USB slots.

ZDNet writes: He did it using USB Killer, a weaponized thumb drive that he purchased from a well-known online store that sells these types of devices. (Amazon and eBay both have them.)

As we know, USB Killers work by charging high voltage capacitors on a thumb drive from the 5v USB power. Once the capacitors have fully charged, a high-voltage transistor is turned on to instantly dump their high voltage burst into the PC's typically unprotected circuitry. On motherboards without adequate transient protection, and those whose main chipsets directly handle the system USB ports, this can reach into and terminally kill the entire computer.

In total, equipment damages amounted to $51,109, along with $7,362 in employee time for investigating and replacing destroyed hardware, which the suspect has agreed to pay as part of the plea deal, according to court documents.

Akuthota was arrested on February 22 and will be sentenced later this year, on August 12. He faces up to ten years in prison, a fine of up to $250,000, and a term of post-imprisonment supervised release of up to 3 years.

What can someone do??

Marcus Hutchins aka "malware tech" posted the following public statement to his blog Friday: https://www.malwaretech.com/public-statement

> Legal Case Update
> As you may be aware, I’ve pleaded guilty to two charges related to writing malware in the years prior to my career in security. I regret these actions and accept full responsibility for my mistakes. Having grown up, I’ve since been using the same skills that I misused several years ago for constructive purposes. I will continue to devote my time to keeping people safe from malware attacks.

On his Twitter page, Marcus said: “*To be clear: this statement wasn’t required by the plea deal, it was my decision to post it.*” — MalwareTech (@MalwareTechBlog) April 21, 2019

We've spoken of Marcus a few times in the past. He's a good guy who does have a prior life of black hat hacking. But he's been wearing a white hat for many years and he had hoped to put the mistakes of his past behind him. As we reported at the time, he was nabbed by Las Vegas police at the Mccarren airport in Las Vegas while preparing to head back to his home in the UK.

And Marcus Hutchins was hailed for squashing the WannaCry ransomware outbreak in May 2017. We'll recall that after examining the code he noted that it was performing a DNS query for the IP address of a non-registered domain. So he registered the domain... and to the world's surprise and great relief the WannaCry ransomware worm stopped its Internet-spanning propagation. At that instant the WannaCry ransomware was just starting to go exponential. It had infected more than 200,000 systems in 150 countries and caused billions of dollars in damages.

However, before he was doing good, he was involved with creating the Kronos banking malware. So, now 24 years old, Marcus has filed a plea agreement admitting guilt to two of 10 counts in the Eastern District of Wisconsin on Friday – one charge for distributing Kronos and the other charge for conspiracy.
Unfortunately, it appears that the US has decided to make an example out of Marcus. According

to court documents he now faces up to 10 years in prison and $500,000 in fines.

One of the Window's 0-days patched two weeks ago...
... is under heavy use in the wild to facilitate full system takeover.

We're learning more about one of the two 0-day flaws Microsoft patched two weeks ago... Since it's now in active use in advanced APT campaigns. As we noted last week, it was discovered by researchers at Kaspersky Lab on Saint Patrick's Day earlier this year when it was used against one of the customers under Kaspersky's protection. It's a use-after-free bug in the Windows kernel win32k.sys module. The flaw allows a local privilege escalation (LPE) and it's being used in advanced persistent threat (APT) campaigns targeting 64-bit versions of Windows (from Windows 7 through older builds of Windows 10).

The attackers are using the bug to establish persistent backdoors in targeted machines, gaining the ability to run arbitrary code in kernel mode. An attacker can then install programs; view, change or delete data and create new accounts with full user rights.

What's most likely in this instance is not that the fix was reverse engineered as is often done to discover previously unknown bugs once they've been fixed... But rather that those who were previously deploying this potent flaw in limited and highly targeted attacks now know that it's useful lifetime is extremely limited since it's been patched. So prior restraint is discarded and they are racing to exploit it into the rapidly dwindling base of still-vulnerable machines.

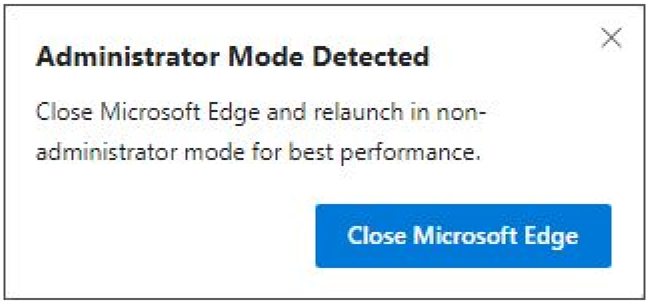
Edge: Warns when it's being run with Admin privilege.
https://www.bleepingcomputer.com/news/security/new-microsoft-edge-to-warn-users-when-in-administrator-mode/

I want to note that I'm very impressed by Bleeping Computer. In his coverage of this, Bleeping Computer's founder, Lawrence Abrams, walks the user through a chain of events that could caused Edge to inadvertently be run with Admin privileges.

The reason this that's not good is that when one process is spawned from another, that spawned process inherits the account, rights and privileges of its owner.

Lawrence takes through a scenario where to edit the hosts file it's necessary to run Notepad as an Admin. Then, while editing the Hosts file you encounter a suspicious entry. Highlighting it and right-clicking you select "Search with Bing" and Edge is launched with Admin rights. If you then, out of curiosity, decide to go to the suspicious domain with your browser, it will be carrying admin rights... Giving malware greater opportunities to exploit your machine. And since the rights persist, you might later download and run something from the browser... And it, too, would have Admin privilege.

So, while at first glance Microsoft's "Administrator Mode Detected" pop-up might seem odd, it's clear that it can be useful.



Administrator Mode Detected

Close Microsoft Edge and relaunch in non-administrator mode for best performance.

We've often noted that today our web browser IS the attack surface of our Internet-connected machines. So we really do need them to be protecting us at all times.

Edge: Microsoft Edge Application Guard.

Microsoft's new Chromium-based Edge browser is also in the process of gaining the ability to run within Microsoft's powerful Windows Defender Application Guard sandbox. It can actually run now, but several warning messages which are useful to inform the user when something has been blocked are not yet present.

I've often lamented the need for a really really strong isolated super-strong sandbox for our browsing. This integration of Chromium and Edge with Win10's latest security features is really looking like the answer to that need.

Edge: The chameleon

While we're on the topic of Edge, it's interesting to note that as the new Chromium Edge re-acquires some of the unique capabilities of its EdgeHTML-based predecessor, it will be dynamically changing its "User Agent" string to show different faces to different sites. For example, when visiting netflix, hbonow, hbogo, napster or sling Edge will display its "Edge" persona. But when visiting facebook, messenger or stan.com.au (an Australian media streamer) Edge will masquerade as "Chrome." First of all, this is not the way a standard-based Internet is supposed to work. This is a horrific kludge. So I hope this is some sort of transitional mess that will be going away eventually. It's like the way all web browsers used to always need to have "mozilla" in their user agent strings.

https://webaim.org/blog/user-agent-string-history/

The Win7 End-Of-Life notices have begun.

My best buddy sent me a text saying... Whaaaaaat???

I haven't seen it yet.

> After 10 years, support for Windows 7 is nearing the end.
> January 14, 2020 is the last day Microsoft will offer security updates and technical support for computers running Windows 7. We known can be difficult, that's why we're reaching out early to help you backup your files and prepare for what's next.

And from the "I did say this was bound to happen" department..
We just have the news that Mozilla's Firefox browser will be enabling "Hyperlink Auditing" (aka URL click tracking) by default in a forthcoming release of Firefox.

https://www.bleepingcomputer.com/news/software/mozilla-firefox-to-enable-hyperlink-ping-tracking-by-default/

Quoting from BleepingComputer's coverage of this news...

Mozilla feels it's a performance improvement

While some users feel this feature is a privacy risk, browsers developers feel that trackers are going to track, so you might as well offer a solution that provides better performance.

In a post by Apple, the WebKit developers explain that hyperlink auditing pings are a performance improvement because unlike other tracking methods, they do not block or delay the navigation to the requested site.

<Apple> "Just turning off the Ping attribute or the Beacon API doesn’t solve the privacy implications of link click analytics. Instead, it creates an incentive for websites to adopt tracking techniques that hurt the user experience. In effect, the choice between supporting Ping and not is not one of privacy, rather it is a choice between a good user experience and a bad one."

After reading Apple's post, I (writes Lawrence Abrams) contacted Mozilla to see if they agreed with the views expressed in the WebKit article.

Mozilla told BleepingComputer via email that they agreed with Apple's views on hyperlink auditing. Furthermore, they stated that the only reason it is not currently enabled by default in Firefox is because their implementation is not ready.

*<Mozilla> "We agree that enabling the hyperlink ping attribute that is commonly used for hyperlink auditing isn’t a question of privacy but a matter of improving the user experience by giving websites a better way to implement hyperlink auditing without the performance downsides of the other existing methods listed in the webkit.org blog post. In fact, we already support the sendBeacon API and the reason we don’t yet en*able the hyperlink ping attribute is that our implementation of this feature isn’t yet complete."

When we asked if they felt that users should at least be given the ability to disable the feature if they wish, Mozilla stated that they did not believe it would have any "meaningful improvement" to a user's privacy.

<Mozilla>"We don’t believe that offering an option to disable this feature alone will have any meaningful improvement in the user privacy, since website can (and often already do) detect the various supported mechanisms for hyperlink auditing in each browser and disabling the more user friendly mechanisms will cause them to fall back to the less user friendly ones, without actually disabling the hyperlink auditing functionality itself."

Brave states it will continue to block this feature

After Mozilla's response, we also contacted Brave Software to ask if they had any plans to enable hyperlink auditing in their browser.

<Brave>"Disabling hyperlink auditing is a crucial privacy feature, and Brave has always disabled this by default," Catherine Corre, Head of Communications at Brave Software, told BleepingComputer via email. "Brave users expect this protection from our browser."

I know this is a fraught topic. After what some of our listeners felt was my own capitulation on this issue I received angry and annoyed feedback through several channels. But it is supremely difficult to not be tracked on the Internet.

Miscellany

https://blog.grc.com

I had two blogs hosted at wordpress.com. I had a CNAME record in my DNS mapping "steve.grc.com" and "blog.grc.com" to wordpress.com. That worked years ago, but it's hostile to TLS since Wordpress doesn't have certificates for those domain names. So browser have been complaining. Wordpress is written in PHP, and I now have a mature PHP server facility setup at GRC. So I decided to setup my own Wordpress system there. Today, I'm glad that I did it for the experience. And I suppose that being me I would do it again. But simply setting up at Wordpress.com is SO MUCH EASIER.

Bolting the thing down is crucial. And after I'd done everything I knew to do I went looking around the Net for the advice of those with more Wordpress-specific experience (and some scars from arrows) than I had. I was pleased that no one had anything to say that I hadn't already arranged a superior solution for. For example, everyone mentions that a really strong password is important and some advice was to have a password lockout in place. Well, I, of course, have a 32-character total gibberish password. But my Wordpress login page cannot even be reached by anyone who is not at one of a very few known IP addresses. So I seemed to be pretty well protected. However, in my roaming I encountered a site that made me think of one of our podcast sponsors -- Wordpress:

"The Top 10 Security Mistakes That Self-Hosted WordPress Blogs Make"
https://antjanus.com/blog/tutorials/the-top-10-security-mistakes-that-self-hosted-wordpress-blogs-make/

According to Forbes, one out of every 6 websites on the Internet is powered by WordPress (nearly 60 million in all), with 100,000 more popping up each day. Wordpress.com currently hosts over 56 million blogs. As of this writing, WordPress stats did not include the number of self-hosted blogs, but rest assured there are many of us! I've been using WordPress since Gold days and it only gets better with each release.

In the past I have been the victim of two WordPress hacks. At the time of the first hack, I was on a managed VPS. All maintenance and administrative tasks (including software updates) was administered by the hosting provider. In my case, the software was rarely updated.

Running a self-hosted blog comes with myriad responsibilities. It is not like you can merely install it and be done with it. Your first priority should be to familiarize yourself with the platform, along with the pros and cons of self-hosting or hosting your blog at WordPress.com.

If you self-host you will need to be somewhat technically savvy – if not, hire someone who is. When you self-host you are responsible for technical maintenance (backend configuration; backups; blog security; logs; spam filtering; and updates).

Take the time to find a reputable and reliable hosting service – do your research first. You don't want to end up on a server that is easily compromised, is slow to update software, has bad tech support, or has too much down time.

The fact that hackers and cybercriminals favor targeting WordPress is for the same reason they favor exploiting Microsoft Windows – it's popular!

I have seen a lot of site admins downplay the importance of updating CMS software and hardening company blogs. This is especially prevalent with small businesses and startups that rely solely on a development teams to schedule site updates and releases.

I've also seen many home businesses slap together self-hosted blogs (because they noticed that cpanel had a Fantastico, Softaculous or an Installatron autoinstaller), and they think that all they have to do is populate their blog with posts, widgets and plugins. For the love of Matt Mullenweg, please check out wordpress.com…

Out with the old (cert), in with the new (EV cert)...

We're going through some upheaval at the moment over the expiration, after three years, of my long standing Authenticode code-signing certificate. As this was approaching I noticed that DigiCert was offering "EV Code Signing" -- which I had never really paid attention to even though Microsoft introduced it seven years ago in 2012. It turns out that to perform EV code signing a developer **must** have a physical encryption dongle, and the secret key **must** be buried deep inside it. This, of course, prevents it from being exfiltrated electronically over a network. So, of course, that's what I got from DigiCert, and I'm really tickled to be able to "EV sign" my work with it for the next three years.

However, the trouble with ANY =new= certificate, is that it will initially have not accumulated any reputation. Literally… I mean reputation. The certificate that I had for three years, had become well known to all A/V systems. They had seen everything signed with it and they all knew that, until they learned otherwise, any code signed with that certificate, even any previously unknown code, was okay… all other signals notwithstanding.

Three years ago, back in 2016, THAT certificate was new, and while I was testing the pre-release of Never10, all sorts of A/V warning were going off. All of these A/V systems are now massively heuristic and if they see anything that remotely looks suspicious they flag it. The last thing they want to be accused of, is missing something. After a week or so of beta testing Never10 before its release, things quieted down and all was well… because the new certificate was beginning to acquire the spotless reputation that it would continue to enjoy for the intervening three years.

But I have a new certificate, and even though it's "EV", it hasn't had a chance to earn a reputation. After all, anyone can obtain a certificate, even an EV certificate. It's what they **do** with it that counts. When I saw what was happening, I immediately re-signed GRC's most popular existing freeware with the new EV cert. DNSBench, InSpectre, Never10 and LeakTest are now signed with the new EV cert. (I didn't resign "SecurAble" which is currently our second most downloaded utility after DNSBench because it contains a secondary device driver and rebuilding it is rather involved… though now I'm thinking that I might co-sign it.) In aggregate, about 3500 copies of those four programs are downloaded every day. So that new certificate will start to be seen and should start to obtain a reputation.

What I'm curious about, now, is whether obtaining the new EV cert in advance and co-signing my code with both the old **and** the new certificate, might have allowed me to avoid and straddle this annoying gap-of-trust I'm now suffering through? Perhaps that's a means of establishing trust in an unknown certificate by having the old cert vouch for the new cert? I'll definitely give that a try three years from now!

And the cool thing is that once this new EV cert's reputation has been earned, there's every reason to believe that the fact that it's EV, and can only be signed with a hardware dongle, will give its reputation even more strength.

For additional information:
http://blog.richpollock.com/2014/06/code-signing-windows-executables-using-authenticode-2-extended-validation-certificates/

On the tube

- StarTrek: Discovery all available on CBS All Access.
- So is the Twilight Zone reboot.
- Game of Thrones / it's all about relationships.

SpinRite

And speaking of my newly relaunched blog…
On April 19th, 2019 at 5:13 pm, MADMAN in MN posted:

Dear Steve,

As what might be called an “historic” user of SpinRite, I have two questions for you:

1. Do you still make available a “retail version” of the product? Or is SpinRite a “download only” at this point?

2. Searching high and low for my several versions of SpinRite, I’ve yet to find the original book/software/serial numbers so… If I could provide you (privately) my name, addresses lived-at when purchased (and registered), for the version(s) I own, would you have existing records to verify my status as an owner?

Any response will be appreciated. As were a number of your utilities (‘Leak Test,’ ‘DCOMbobulator,’ “Never 10') but above them all I’d hardly be able to tell you how MANY times SpinRite saved flaky, error-ridden, discs — be they floppies or HDDs!

I replied to Madman in MN...

My next post here (https://blog.grc.com) will be about my roadmap for SpinRite, since people who don’t listen to the Security Now! podcast will not have heard about my plans. And having them documented would be useful in any event.

1. We (thank god) no longer have any physical shipment of SpinRite. Only the download. That makes the lives of my little three-employee company so much more sane.

2. We still maintain a database of every copy of SpinRite ever purchased going back 30+ years now. We have pre- and post- online databases, and the pre-online db is written in FoxPro (a dBase II clone). If you will write to Sue at our sales eMail which is always sales{current year}@grc.com she’ll be glad to look you up and verify your status.

Since SpinRite v6 has been in use for the past 15 years, we are giving serious consideration to terminating upgrades from earlier versions once v6.1 is formally released under the thinking that since v6.1 is going to be so much faster and more capable than v6.0, and we’re going to be giving it away to all v6.0 owners going back 15 years… that should be a sufficient commitment to our previous customers. And that anyone who’s still interested in SpinRite at all will have upgraded to v6 sometime in the past 15 years… so they’ll be covered.

And thank you VERY MUCH for your interest and support! ALL of those other things you mentioned that I have done, and have been able to give away, is because of people purchasing SpinRite.

/Steve.

Hijacking DNS

Cisco Talos:

DNS Hijacking Abuses Trust In Core Internet Service

https://blog.talosintelligence.com/2019/04/seaturtle.html

PREFACE

This blog post discusses the technical details of a state-sponsored attack manipulating DNS systems. While this incident is limited to targeting primarily national security organizations in the Middle East and North Africa, and we do not want to overstate the consequences of this specific campaign, we are concerned that the success of this operation will lead to actors more broadly attacking the global DNS system. DNS is a foundational technology supporting the Internet. Manipulating that system has the potential to undermine the trust users have on the internet. That trust and the stability of the DNS system as a whole drives the global economy. Responsible nations should avoid targeting this system, work together to establish an accepted global norm that this system and the organizations that control it are off-limits, and cooperate in pursuing those actors who act irresponsibly by targeting this system.

EXECUTIVE SUMMARY

Cisco Talos has discovered a new cyber threat campaign that we are calling "Sea Turtle," which is targeting public and private entities, including national security organizations, located primarily in the Middle East and North Africa. The ongoing operation likely began as early as January 2017 and has continued through the first quarter of 2019. Our investigation revealed that at least 40 different organizations across 13 different countries were compromised during this campaign. We assess with high confidence that this activity is being carried out by an advanced, state-sponsored actor that seeks to obtain persistent access to sensitive networks and systems.

The actors behind this campaign have focused on using DNS hijacking as a mechanism for achieving their ultimate objectives. DNS hijacking occurs when the actor can illicitly modify DNS name records to point users to actor-controlled servers. The Department of Homeland Security (DHS) issued an alert about this activity on Jan. 24 2019, warning that an attacker could redirect user traffic and obtain valid encryption certificates for an organization's domain names.

In the Sea Turtle campaign, Talos was able to identify two distinct groups of victims. The first group, we identify as primary victims, includes national security organizations, ministries of foreign affairs, and prominent energy organizations. The threat actor targeted third-party entities that provide services to these primary entities to obtain access. Targets that fall into the secondary victim category include numerous DNS registrars, telecommunication companies, and internet service providers. One of the most notable aspects of this campaign was how they were able to perform DNS hijacking of their primary victims by first targeting these third-party entities.

We assess with high confidence that these operations are distinctly different and independent from the operations performed by DNSpionage, which we reported on in November 2018. The Sea Turtle campaign almost certainly poses a more severe threat than DNSpionage given the actor's methodology in targeting various DNS registrars and registries. The level of access we presume necessary to engage in DNS hijacking successfully indicates an ongoing, high degree of threat to organizations in the targeted regions. Due to the effectiveness of this approach, we encourage all organizations, globally, to ensure they have taken steps to minimize the possibility of malicious actors duplicating this attack methodology.

The threat actors behind the Sea Turtle campaign show clear signs of being highly capable and brazen in their endeavors. The actors are responsible for the first publicly confirmed case against an organizations that manages a root server zone, highlighting the attacker's sophistication. Notably, the threat actors have continued their attacks despite public reports documenting various aspects of their activity, suggesting they are unusually brazen and may be difficult to deter going forward. In most cases, threat actors typically stop or slow down their activities once their campaigns are publicly revealed.

This post provides the technical findings you would typically see in a Talos blog. We will also offer some commentary on the threat actor's tradecraft, including possible explanations about the actor's attack methodology and thought process. Finally, we will share the IOCs that we have observed thus far, although we are confident there are more that we have not seen.

Assessed Sea Turtle DNS hijacking methodology

It is important to remember that the DNS hijacking is merely a means for the attackers to achieve their primary objective. Based on observed behaviors, we believe the actor ultimately intended to steal credentials to gain access to networks and systems of interest. To achieve their goals, the actors behind Sea Turtle:

- Established a means to control the DNS records of the target.
- Modified DNS records to point legitimate users of the target to actor-controlled servers.
- Captured legitimate user credentials when users interacted with these actor-controlled servers.

INITIAL ACCESS
The threat actors behind the Sea Turtle campaign gained initial access either by exploiting known vulnerabilities or by sending spear-phishing emails. Talos believes that the threat actors have exploited multiple known CVEs to either gain initial access or to move laterally within an affected organization. Based on our research, we know the actor utilizes the following known exploits:

- CVE-2009-1151: PHP code injection vulnerability affecting phpMyAdmin
- CVE-2014-6271: RCE affecting GNU bash system, specifically SMTP (Shellshock CVEs)
- CVE-2017-3881: RCE by unauthenticated user with elevated privileges Cisco switches
- CVE-2017-6736: Remote Code Exploit (RCE) for Cisco integrated Service Router 2811
- CVE-2017-12617: RCE affecting Apache web servers running Tomcat
- CVE-2018-0296: Directory traversal allowing unauthorized access to Cisco Adaptive Security Appliances (ASAs) and firewalls
- CVE-2018-7600: RCE for Website built with Drupal, aka "Drupalgeddon"

Credential harvesting: Man-in-the-middle servers

Once the threat actors accessed a domain's DNS records, the next step was to set up a man-in-the-middle (MitM) framework on an actor-controlled server.

The next step for the actor was to build MitM servers that impersonated legitimate services to capture user credentials. Once these credentials were captured, the user would then be passed to the legitimate service. to evade detection, the actors performed "certificate impersonation," a technique in which the attacker obtained a certificate authority-signed X.509 certificate from another provider for the same domain imitating the one already used by the targeted organization. For example, if a DigiCert certificate protected a website, the threat actors would obtain a certificate for the same domain but from another provider, such as Let's Encrypt or Comodo. This tactic would make detecting the MitM attack more difficult, as a user's web browser would still display the expected "SSL padlock" in the URL bar.

When the victim entered their password into the attacker's spoofed webpage, the actor would capture these credentials for future use. The only indication a victim received was a brief lag between when the user entered their information and when they obtained access to the service. This would also leave almost no evidence for network defenders to discover, as legitimate network credentials were used to access the accounts.

How is this tradecraft different?

The threat actors behind the Sea Turtle campaign have proven to be highly capable, as they have been able to perform operations for over two years and have been undeterred by public reports documenting various aspects of their activity. This cyber threat campaign represents the first known case of a domain name registry organization that was compromised for cyber espionage operations.

In order to distinguish this activity from the previous reporting on other attackers, such as those affiliated with DNSpionage, below is a list of traits that are unique to the threat actors behind the Sea Turtle campaign:

- These actors perform DNS hijacking through the use of actor-controlled name servers.
- These actors have been more aggressive in their pursuit targeting DNS registries and a number of registrars, including those that manage ccTLDs.
- These actors use Let's Encrypts, Comodo, Sectigo, and self-signed certificates in their MitM servers to gain the initial round of credentials.
- Once they have access to the network, they steal the organization's legitimate SSL certificate and use it on actor-controlled servers.

~30~