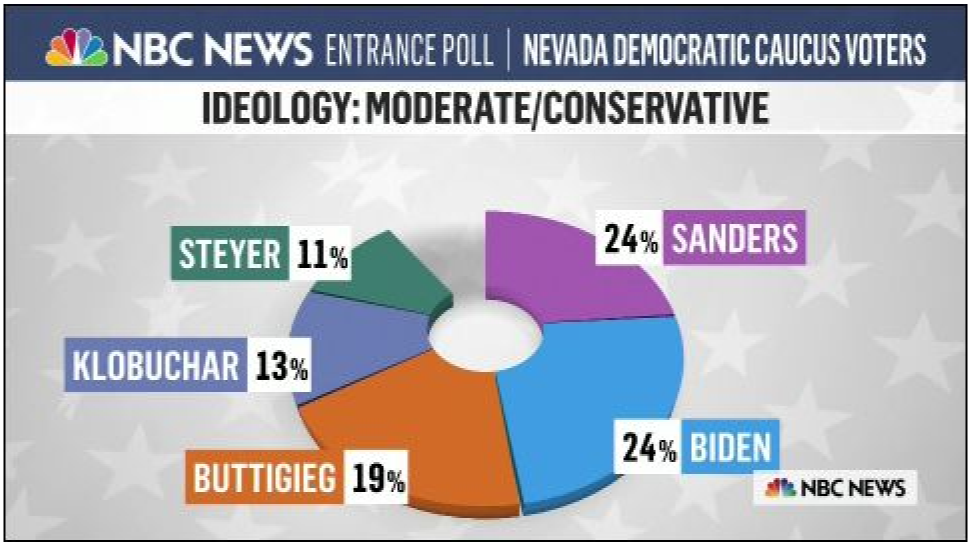
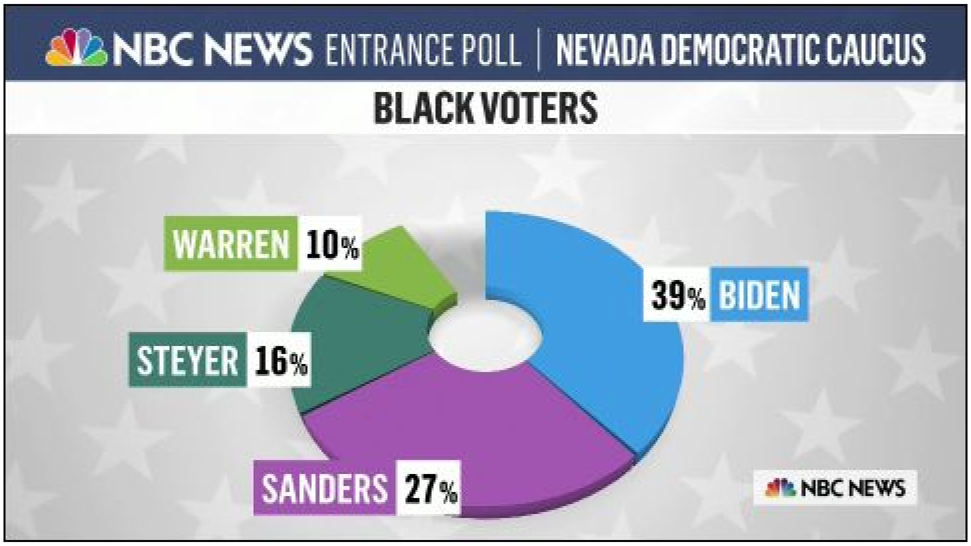
Security Now! #755 - 02-25-20
Apple’s Cert Surprise

This week on Security Now!

This week we reexamine the Windows 10 lost profiles problem, and also a consequence of the need to roll-back (or avoid in the first place) the Patch Tuesday disaster. We look at a new feature to arrive with the next Windows 10 feature release, unfortunately named the 2004 release, we also examine the details of a new attack on the 4G LTE and 5G cellular technology, the full default roll-out of Firefox's support for DoH, and also the availability of a powerful new sandboxing technology for Firefox. We also check in with Chrome's fix earlier today of a 0-day that was found being exploited in the wild. And finally, before turning our attention to the bomb that Apple dropped in the lap of the entire certificate industry last week, I'm going to update our listeners about the things I've learned after returning to the work on SpinRite's next iteration.

Our “Cranky Old Guy” Pictures of the Week





Security News

More Windows 10 lost profile pain

Okay, so remember how we said last week that even though after installing the February KB4532693 update roll-up your profile might have disappeared leaving you with an empty desktop and nothing in your Documents and other user-specific folders?

But that, the good news was, nothing was actually deleted and it could all be recovered? Well... Uhhh... it appears that may not always be true. Reports are accumulating of profiles occasionally actually being permanently deleted and responsible tech publications, such as Bleeping Computer are reporting that, yes, in fact, some users are apparently losing everything permanently. Bleeping Computer's article has the headline: “Windows 10 KB4532693 Update Bug Reportedly Deletes User Files.”

They explain everything we thought we knew, that the user's folder under the /users/ directory might have been renamed to .000 or .bak and not renamed back to their username after the update. And that uninstalling the update would bring it all back. But there is increasingly strong evidence presented by people, some of whom appear to know what they're doing, indicating that there's something even more wrong, such that the upgrade's renaming of the profile sometimes results, instead, in its permanent deletion.

Bleeping Computer goes into some detail about this, citing a page from the Windows Latest site:
https://www.windowslatest.com/2020/02/18/reports-windows-10-update-data-deletion-bug/

And some worrying tweets and postings to Microsoft’s Answer forum, such as...

https://answers.microsoft.com/en-us/windows/forum/all/cumulative-updates-february-11th-2020/548d4ded-39a1-4270-a866-627ea7c25de6?auth=1&page=3&msgid=cb23a2cf-0091-4c40-a20f-6fa0db07259e

> I was on the phone with Microsoft for 4 hours. Surface Pro 7 about 2 months old. There's no user data, no temporary account, no restore points. Uninstalling the update didn't work because my user data is gone. Microsoft is calling in the morning because they think my personal services are free. I know the only option is to flatten the Surface and start over, like Groundhog Day the movie. I have two desktops upgraded from Windows 7 that survived, if that helps.

https://answers.microsoft.com/en-us/windows/forum/all/cumulative-updates-february-11th-2020/548d4ded-39a1-4270-a866-627ea7c25de6?auth=1&page=10&msgid=c8c7800e-b6d5-495f-b22b-2f1c7e8169b0

> This update KB4532693 caused all the data on my laptop to be erased. Even after uninstalling the update, the laptop would not successfully boot. Then "Resetting" the laptop to factory settings while choosing to keep all personal data erased everything. EVERYTHING.

Bleeping Computer's advice is to first try restarting the system several times -- like perhaps as many as four to six times. Apparently that sometimes works.

As we know, these days it’s really the case that our backups need to have backups. And after a few close calls with old and spontaneously dying machines I’m now backed up every which way. And I’m seriously considering migrating my entire user profile to sync.com -- rather than just selected working directory hierarchies. That was all of my stuff will be sync’d and version-managed across multiple machines.

I firmly practice safe computing. So I hope that the danger from anything malicious successfully attacking me is minimal. But it’s disturbing to imagine that running a Windows Update could be the Trojan horse that I willingly allow to enter through my front gate.

A Micropatch for the jscript.dll problem

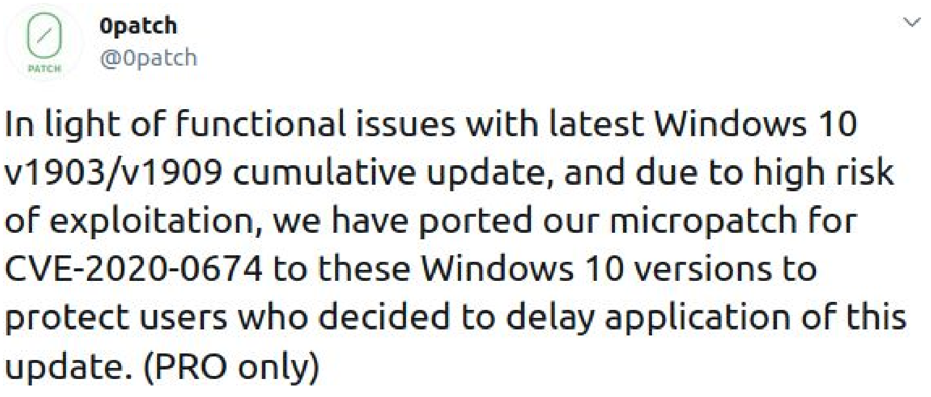
Remember the serious 0-day exploit that we learned about involving IE and the jscript.dll? We know that this vulnerability has been exploited in the wild in limited targeted attacks. The bad guys can leverage it to silently execute arbitrary commands on an unpatched system when the user visits a specially crafted website. And there are many other ways to get that jscript.dll to do an attacker's bidding.

The severity of the issue prompted Microsoft to suggest a short-term fix until KB4532693 became available. The flaw is being tracked as CVE-2020-0674 and the temporary fix was to remove all access to that DLL by deleting all security permissions from the file. Then nothing, good or bad, could cause its invocation. But many people soon discovered that their Windows Media Player would no longer work, and other things broke, such as HP and other USB printers.

The solution was Patch Tuesday's roll-up of fixes. But... well... we know how well that has worked out. February's update cure has proven to be worse than the problems it was trying to resolve. So many users -- especially corporations -- might wisely be choosing to just skip the February update and hope that March turns out better. But this leaves these cautious users with a problem...

Unless they restrict access to the jscript.dll they are vulnerable in the meantime to known targeted attacks which will almost certainly increase in prevalence now that the exploit is known and the window of their exploitation is closing. But, as we know, wholesale restriction to the DLL is overly broad and breaks too many other things.

So… enter the MicroPatch. Due to its continued demand, the guys at 0patch.com have extended the reach and scope of their patch for the 2020-0674 vulnerability in the jscript.dll.



0patch
@0patch

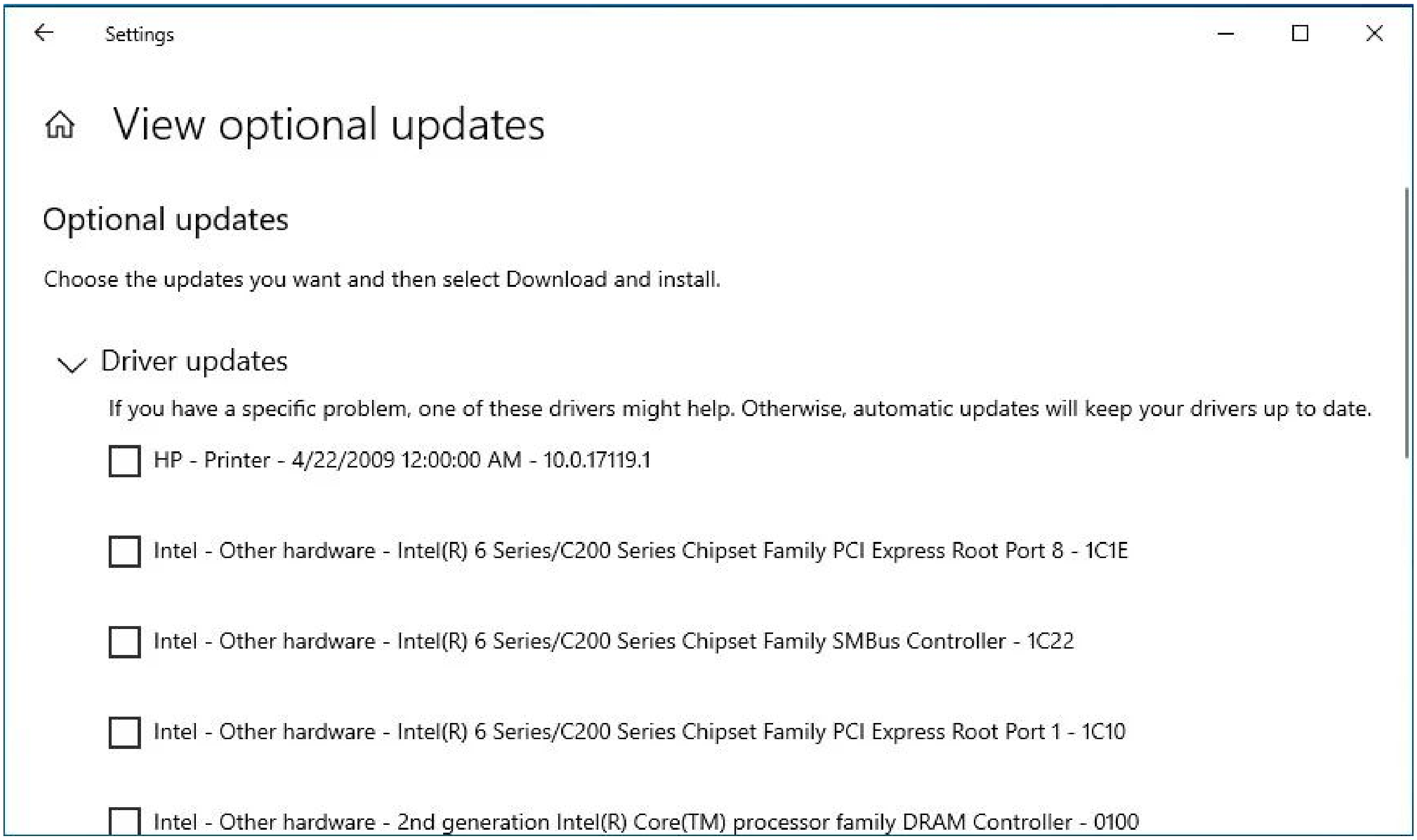
In light of functional issues with latest Windows 10 v1903/v1909 cumulative update, and due to high risk of exploitation, we have ported our micropatch for CVE-2020-0674 to these Windows 10 versions to protect users who decided to delay application of this update. (PRO only)

Their original interim solution was available for Windows 7, Windows 10 v1709/v1803/v1809, Windows Server 2008 R2, and Windows Server 2019... But they didn't have it working and available for v1903 or v1909. That's what's changed. It is offered to users of the free version of the service, which is allowed for non-commercial use only, as well as to paying customers (Pro - $25/agent/year - and Enterprise license holders).

https://0patch.com/poc/CVE-2020-0674/0patch_test.html

There's even a cool test page that can be used to verify the proof-of-concept and the fix. The test requires IE11 on Windows 7, Server 2008 R2 or Windows 10 v1903/v1909.

Coming in the next Feature Release (Win10 2004): Optional device driver updates
Starting now, device driver providers will be able to mark their device driver updates as "Automatic" or "Manual." Any device driver updates marked "Automatic" will be included in the Windows automatic update package. But any device driver updates marked "Manual" will first appear in the new "Optional Updates" section:



View optional updates

Optional updates

Choose the updates you want and then select Download and install.

Driver updates

If you have a specific problem, one of these drivers might help. Otherwise, automatic updates will keep your drivers up to date.

□ HP - Printer - 4/22/2009 12:00:00 AM - 10.0.17119.1

□ Intel - Other hardware - Intel(R) 6 Series/C200 Series Chipset Family PCI Express Root Port 8 - 1C1E

□ Intel - Other hardware - Intel(R) 6 Series/C200 Series Chipset Family SMBus Controller - 1C22

□ Intel - Other hardware - Intel(R) 6 Series/C200 Series Chipset Family PCI Express Root Port 1 - 1C10

□ Intel - Other hardware - 2nd generation Intel(R) Core(TM) processor family DRAM Controller - 0100

These changes will not be back-ported to earlier Win10 builds. So they will only be seen by those running the forthcoming (and annoyingly named) 2004 release of Windows 10.

Microsoft explains that this change will allow hardware developers to roll out new drivers and test them for reliability against a smaller group of Windows users before pushing them out to a wider audience. Microsoft believes these changes will help their customers to "get the highest quality, and most reliable drivers faster and with less friction." ... but what I see is a backsliding to a more complex user experience. What are typical users going to think when they are confronted with “Optional Updates”?? I have long felt that this was a very poor user experience design. Updates should not be optional -- they are either needed or they’re not needed. Which is it? I have never understood the logic of putting that decision in the hands of the user.

And, since Microsoft’s Kevin Tremblay clearly stated that: “The changes to Ux for Windows 10 20H1 detailed here will not be backported to previous releases. For older versions of Windows, “Manual Update” drivers will be obtained via Device Manager.” So… who’s going to go to the Device Manager for manual driver update? How would you know there were any? Do you step through each device and attempt to update its driver? This makes it seem as though they really are not necessary… in which case, presenting the option to users in the automatically forced update system seems wrong.

A new attack on 4G LTE and 5G

A team of researchers who have been poking at modern cell phone security and integrity are presenting the worrisome results of their latest research today, the 25th of February, 2020 during NDSS, the Network Distributed System Security Symposium (which, I guess really should be NDSSS, but perhaps they thought that was one 'S' too many) in San Diego, California.

There's a website for the work: https://imp4gt-attacks.net/ And we have the pre-release of their presentation paper: https://imp4gt-attacks.net/media/imp4gt_camera_ready.pdf

A full understanding for what they have, and have done, requires a thorough understanding of the inner workings of cell-system networking. And in this instance even the Abstract from their paper, which is normally useful, assumes too much background. But, their paper's introduction does give us a good sense for the importance of this work:

> Long Term Evolution (LTE) is the latest widely deployed mobile communication standard and is used by hundreds of millions of people worldwide. The protocol offers high-speed Internet access and packet-based telephony services and has become an integral component of our daily communication.We fundamentally rely on the security of LTE for a variety of applications. The security goals of LTE include, amongst others, mutual authentication, traffic confidentiality, and location privacy; any attack vector undermining these security aims has far-reaching implications to the use of LTE as a communication medium.
>
> In the context of mobile communication, mutual authenti-cation is an important security aim since it ensures that both communication parties (i. e., the user equipment and the net-work) mutually verify their identities. As the wireless medium is accessible for everyone in the vicinity and identifiers can be easily forged, mutual authentication is essential for build-ing trust between communication parties. Telecommunication providers rely on user authentication for accounting, authorization, and the association of data sessions to a legal person. The latter case is of particular importance in prosecution, in which a possible offender is accused of committing a crime via a mobile Internet connection. Additionally, users rely on network authentication for the confidentiality of their communication. One important example for missing network authentication is the second mobile network generation GSM (Global Systemfor Mobile Communications): by faking the identity of a legitimate network, an attacker can impersonate the network in GSM and eavesdrop on the communication of the victim.

> In contrast to earlier network generations, LTE establishes mutual authentication on layer three of the network stack using a provably secure Authentication and Key Agreement (AKA) protocol. Based on this protocol, subsequent encryption ensures the confidentiality of user and control data. Permanent integrity protection, however, is only applied to the **control** data. A recent study has revealed that missing integrity protection of the **user** plane on layer two allows to manipulate user data in a deterministic way. More specifically, a layer-two attacker in a Man-in-the-Middle (MitM) position between the phone and the network can introduce undetectable bit flips due to malleable encryption and redirect traffic to another destination. While this attack demonstrates the potential consequences of traffic manipulation, it is solely limited to redirecting traffic to another destination.
>
> In **this** work, we introduce a novel cross-layer attack concept that complements the known layer-two vulnerability (i. e., missing integrity protection on the user plane) with exploiting the default IP stack behavior of operating systems on layer three. More precisely, we make use of the reflection mechanism of certain IP packets, which allows us to not only redirect user-plane traffic, but also to create an encryption and decryption oracle that enables an adversary to perform a full impersonation of the phone or network on the user plane. We call this concept IMP4GT (IMPersonation in 4G neTworks, pronounced “impact”). IMP4GT completely breaks the mutual authentication property for the user plane on layer three, as an attacker can send and receive arbitrary IP packets despite any encryption.
>
> This attack has far-reaching consequences for providers and users. Providers can no longer assume that an IP connection originates from the user. Billing mechanisms can be triggered by an adversary, causing the exhaustion of data limits, and any access control or the providers’ firewall can be bypassed. A possible impersonation also has consequences for legal prosecution, as an attacker can establish arbitrary IP connections associated with the victim’s identity.

That encryption/decryption oracle is the key to this. They establish a man-in-the-middle interception using a software defined radio (SDR). They are then able to probe the encryption by flipping bits, which results in a failure and retransmission. They actually inject ICMP Unreachable and ICMP Ping packets into the stream in order to get either endpoint to reply.

To give our listeners a better and more convincing sense of this, yhey explain the operation of their Encryption and Decryption Oracles as follows:

Encryption Oracle.
The goal of an encryption oracle is to learn the keystream of a connection, which later allows [us] to encrypt and inject arbitrary packets. For encrypting a target plaintext, the oracle injects a known plaintext into the system. The system encrypts the packet by xor-ing the known-plaintext with a valid keystream for transmission, which is returned to the oracle. Now, the oracle can extract the valid keystream by xor-ing the known-plaintext on the encrypted packet. Any arbitrary payload can now be encrypted by xor-ing the target plaintext and the keystream.

(B) Decryption Oracle.
The goal of a decryption oracle isto decrypt and access the payload of an encrypted packet. To

achieve the decryption of a packet, the oracle manipulates the to-be-decrypted ciphertext and sends it to the system. The system decrypts the packet and subsequently sends it back to the oracle. In this way, we can receive the plaintext of encrypted packets.

=====
They go into far greater detail in their paper. But they have conclusively demonstrated a fundamental weakness in both 4G LTE and the forthcoming 5G, since neither of these systems provides the needed message integrity protection at the user layer. It must have been assumed -- by non-cryptographers -- that the encryption running at the user layer would sufficiently protect the user's communications. But we know that XOR-based stream ciphers, while highly attractive due to their economy and ease of implementation, are also highly susceptible to interception attacks that can trivially reveal the keystream if the plaintext can be known.

They clearly state that the only way for this to be fixed is for all of our existing cell-system infrastructure hardware to be upgraded at the smartphone and cell tower level. And we know that's never going to happen. They are hoping that there might still be time to head-off 5G, which repeats these mistakes, but they acknowledge that's unlikely.

Services such as iMessage and Signal, which provide their own application-level encryption are secure against this. HTTPS is less certain, since we rely upon some aspects of the integrity of the underlying network, such as DNS and that we are actually connecting to the machine we think we are, that this work has demonstrated the power to subvert.

And, in any event, due to the need for physical MITM proximity, this would only be applicable to targeted attacks. But it does, and it should, further shake the complacency we have with the security of our smartphones.

Starting today: DoH by default on Firefox

Mozilla's rollout of DND over HTTPS begins today and it will continue gradually over the next few weeks to confirm that no showstopper issues are discovered as DoH is enabled for Firefox's users in the United States.

The way this will work is that, starting today, all new Firefox installs in the US will have DoH enabled by default and it will be silently enabled for all Firefox US users during the coming weeks. The only users who will NOT receive this update are those who have specifically disabled DoH in Firefox's settings panel.

As we know, the move to encrypting DNS by tunneling it over HTTPS has not been welcomed by everyone. The most concerted push-back came from the UK, where an ISP association went as far as to nominate Mozilla for the title of 2019 Internet Villain due to its work on the DoH protocol. (A nomination they subsequently rescinded.)

But the ISPs warned that rolling out DoH would cripple the UK's national-wide firewall system that ISPs and law enforcement are using to limit access to child abuse websites and copyright infringement domains. After their lobbying efforts were joined by law enforcement and the British government Mozilla capitulated last July and announced they would not be enabling DoH for UK users... at least for now.

And whether or not ISP's and governments like it, DoH appears to be where the industry is headed. It is now supported by all major browsers... even if it's not always easy to find. It's in there. And even Microsoft has announced plans to add support for DoH to native Windows in the future.

As for Mozilla, today, Firefox's default DoH provider remains Cloudflare, a decision I wholeheartedly support. Not only has Cloudflare formally asserted that they will not log, nor monitor nor intercept this use of DNS over HTTPS in any way... but it's Cloudflare making that assertion and there are very few companies that I would trust more to actually honor that pledge.

But, worries over the adoption of a monoculture always carry some validity, so Mozilla also allows the selection of NextDNS -- also a good group -- for those who choose to be contrarian.

And in other important Firefox news:
A new next-generation WebAssembly sandbox is coming first to Linux and Mac and then to Windows.

It is the result of a bunch of hard work by a team from UC San Diego, UT Austin, Stanford University and Mozilla. It brings the next step in protecting users from both malicious and inadvertently exploitable libraries that their web pages might load.

https://github.com/shravanrn/LibrarySandboxing/blob/master/paper.pdf
The detailed technical paper describing it is titled: “Retrofitting Fine Grain Isolation in the Firefox Renderer”

> All major browsers today employ coarse grain privilege separation to limit the impact of vulnerabilities. To wit, they run renderers—the portion of the browser that handles untrusted user content from HTML parsing, to JavaScript execution, to image decoding and rendering—in separate sandboxed processes. This stops web attackers that manage to compromise the renderer from abusing local OS resources to, say, install malware.
>
> Unfortunately, this is no longer enough: nearly everything we care about today is done through a website. By compromising the renderer, an attacker gets total control of the current site and, often, any other sites the browser has credentials for. With services like Dropbox and Google Drive, privilege separation is insufficient even to protect local files that sync with the cloud.
>
> Browser vendors spend a huge amount of engineering effort trying to find renderer vulnerabilities in their own code. Unfortunately, many remain—frequently in the dozens of third-party libraries used by the renderer to decode audio, images, fonts, and other content. For example, an out-of-bounds write in libvorbis was used to exploit Firefox at Pwn2Own 2018. Both Chrome and Firefox were vulnerable to an integer-overflow bug in the libvpx video decoding library. Both also rely on the Skia graphics library, which had four remote code execution bugs until recently.

> To appreciate the impact of these vulnerabilities and the difficulty of mitigating them, consider a typical web user, Alice, that uses Gmail to read email in her browser. Suppose an intruder, Trudy, sends Alice an email that contains a link to her malicious site, hosted on sites.google.com. If Alice clicks on the link, her browser will navigate her to Trudy's site, which can embed an .ogg audio track or .webm video to exploit vulnerabilities in libvorbis and libvpx and compromise the renderer of Alice's browser. Trudy now has total control of Alice's Gmail account. Trudy can read and send emails as Alice, for example, to respond to password reset requests from other sites Alice belongs to. In most cases, Trudy can also attack cross site, i.e., she can access any other site that Alice is logged into (e.g., Alice's amazon.com account).
>
> Recent version of Chrome (and upcoming versions of Firefox) support Site Isolation, which isolates different sites from each other (e.g., *.google.com from *.amazon.com) to prevent such cross-site attacks. Unfortunately, Trudy might still be able to access {drive,pay,cloud}.google.com, which manage Alice's files, online payments, and cloud infrastructure—since the renderer that loads the malicious .ogg and .webm content might still be running in the same process as those origins.
>
> For many sites, Trudy might not even need to upload malicious content to the (trusted) victim origin (sites.google.com in our example). Most web applications load content, including images, fonts, and video, from different origins. Of the Alexa top 500 websites, for example, over 93% of the sites load at least one such cross-origin resource. And the libraries handling such content are not isolated from the embedding origin, even with Site Isolation.
>
> To mitigate these vulnerabilities, we need to harden the renderer itself. To this end, we extend the Firefox renderer to isolate third party libraries in fine grain sandboxes. Using this, we can prevent a compromised library from gaining control of the current origin or any other origin in the browser.

As we've often observed, our web browsers have become the largest attack surface that we extend out onto the Internet. By the very nature of Web Browsing, we're actively soliciting sites we know little about, or trusted sites that may have, themselves, been compromised, to send pages of code to our browser, and to instruct our browser to also fetch massive 3rd-party libraries and advertisements from all over the Internet. And we bring them all in to be processed, rendered, executed and displayed.

I'm so happy that we have these guys, and guys like them, watching our backs. Lord knows, we need it!

This new RLBox technology will first be deployed in Firefox 74 for Linux, which is set to be released in early March. Then, in April, RLBox will ship for Firefox 75 for Mac. And its deployment in Firefox for Windows will eventually catch up.

Chrome was just updated to close a 0-day:
Chrome users on Windows, Mac and Linux will want to be sure they're running v80.0.3987.122. This apparently doesn't affect Chrome OS nor iOS or Android.

The update was released this morning to address three security bugs, including a zero-day vulnerability that, as its designation implies, is being actively exploited in the wild. We don't yet know anything about the attacks, and only that the vulnerability has a CVE tracker which generically describes it as a “type confusion in V8” where V8 is Chrome's JavaScript interpreter/ compiler. We do know that the use of this bug in attacks was discovered exactly one week ago, on February 18, by a member of Google's Threat Analysis Group.

This one is the 3rd Chrome 0-day to be discovered in the wild in the past 12 months. Google patched the first Chrome 0-day last March and the second in November. And this is to be expected since Chrome, as the #1 browser on the Internet, is now also the #1 attack target.

SpinRite

- Re-reading all of the posts from 2013.
    - I started with a little more than 6,000 and so far I’ve plowed through about 1200, so I have about 4800 remaining to read.

- ACHI vs Legacy mode: A different choice after seven years

- UEFI vs BIOS -- Macs are going to pose a problem in the near term.
- rEFIt / rEFInd

- FreeDOS - Modified/Customized the Kernel

Apple’s Cert Surprise

Last week, before there had been any news coverage of the surprising event, I received the following letter from Dean Tayler, Senior Account Manager at DigiCert:

> Dear Steven Gibson,
>
> Earlier today, Apple announced that Safari will only trust certificates with a validity of 398 days or less (one year plus a renewal grace period). This policy goes into effect September 1, 2020.
>
> Certificates issued before that date are not affected and do not need to be replaced or modified—you can continue to issue 2-year certificates until August 31, 2020, and use them until their expiration. This announcement was made by Apple on February 19th at CA/Browser Forum, an industry standards group meeting.
>
> While it’s generally accepted that short-lived certificates will increase the security of the SSL ecosystem, we have been working with the browsers to time this change in a way that reduces the impact to our customers. While Apple’s decision was unilateral, we already have tools in place to make short-lived certificate management easier and we are working on additional solutions ahead of this change to offer you with greater certificate lifecycle automation options.
>
> I know this impacts your certificate management practices; that’s why I wanted to let you know about the coming change and tell you that we are responding. At DigiCert®, we always put our best foot forward when it comes to offering solutions for certificate management. We already developed robust certificate lifecycle automation tools, but with Apple’s plans now in play, it’s more important than ever to move to automated lifecycle management. With that, I wanted to share some information about our continued promise to offer the best in certificate automation.
>
> Before the Apple changes occur, we’ll add the ability for you to purchase multi-year certificate subscriptions to smooth planning and reduce the yearly work of buying and installing certificates. These subscriptions will let you reissue, renew, or replace a certificate as frequently as you need to without incurring additional fees. Our intention with offering subscriptions is to save you time and money.
>
> Hand-in-hand with multi-year certificate subscriptions is the ability to automate the entire certificate lifecycle. As the industry moves to shorter certificate lifetimes, automation is the key to keep your business running smoothly. To save time, avoid annual manual updates, and to avoid site downtime, DigiCert CertCentral offers you several ways to automate your SSL certificate needs: robust APIs, ACME integration, and our certificate automation tool. Automation allows you to spend more time doing what you want to do and less time managing certificates.
>
> If you have any questions about our plans, what features are on our roadmap, or if you have any questions about Apple’s certificate policy change, please don’t hesitate to reach out to me.

Okay. So, let's start by reviewing a bit of certificate history that's too easily forgotten.

Back when this podcast was just beginning, when Honey Monkeys were crawling around and, come to think of it, Leo's favorite password was “Monkey”, it was possible to obtain an SSL certificate from a certificate authority which would remain valid for eight to ten years from the time of purchase.

Then, in 2011, the Certification Authority Browser Forum (the CAB Forum) which included all of the major browser makers, decided that was too long due to the enduring flakiness of certificate revocation and the possibility that a rogue certificate might escape and be honored for as many as 10 years. So... The CAB forum all agreed to cut certificate maximum life in half, to 5 years.

And four years after that, in 2015 the time limit was dropped to three years with a limit of two years for EV certs.

And now, last week, in a unilateral move, Apple decided to flex its own muscles by announcing in a declaration during the CAB Forum meeting held in Bratislava, Slovakia that, effective as of this coming September 1st, 2020, none of its Safari browsers would consider any certificate issued after August 31st, 2020 to be valid if it had a lifetime greater than 398 days or effectively 13 months. The extra month is to serve as a grace period for pre-expiration certificate reissuance.

There's been a lot of opinion circulating as fact in the days since this. So I went looking for an authoritative reference and found a statement written by Dean Coclin who is DigiCert's Senior Director of Business Development, titled: Digicert's position on 1-year certificates:

> At the CA/Browser (CA/B) Forum in Bratislava, Slovakia, this week, Apple announced that beginning Sept. 1, newly issued publicly trusted TLS certificates are valid for no longer than 398 days. This followed a long history of the CA/B Forum community working to reduce certificate lifetimes and improve security, while balancing the needs of business owners in transitioning to shorter validity certificates.
>
> In August 2019, CA/B Forum Ballot SC22 was introduced by Google to reduce TLS certificate validity periods to one year. CAs reviewed this proposal with their customers and produced thousands of comments from users, which mostly showed opposition, due to the additional work required by IT teams to handle shorter validity periods. The ballot failed in the Forum, which meant certificate maximum lifetimes remained at two years.
>
> At one time, certificates were offered with a maximum validity of three years. A few years ago, they were reduced to two years. Fast forward to this week’s Apple announcement, which ultimately does what ballot SC22 failed to do: reduce certificate lifetimes to one year.
>
> Why did Apple unilaterally decide to enforce a shorter certificate lifetime? Their spokesperson said it was to “protect users.” We know from prior CA/B Forum discussions that longer certificate lifetimes proved to be challenging in replacing certificates, in the case of a major security incident. Apple clearly wants to avoid an ecosystem that cannot quickly respond to major certificate-related threats. Short-lived certificates improve security because they reduce the window of exposure if a TLS certificate is compromised. They also help remediate normal

> operational churn within organizations by ensuring yearly updates to identity such as company names, addresses and active domains. As with any improvement, shortening of lifetimes should be balanced against the hardship required of certificate users to implement these changes.
>
> It should be noted that Apple has not released a Knowledge Base article on this subject, so things could change pending their official announcement. Assuming, however, that the change is implemented, what does this mean for certificate users? For your website to be trusted by Safari, you will no longer be able to issue publicly trusted TLS certificates with validities longer than 398 days after Aug. 30, 2020. Any certificates issued before Sept. 1, 2020 will still be valid, regardless of the validity period (up to 825 days). Certificates that are not publicly trusted can still be recognized, up to a maximum validity of 825 days.
>
> DigiCert agrees that shorter lifetimes help enhance the security of the ecosystem and has the tools necessary to help our customers automate the certificate lifecycle process. We support short-lived certificates, with lifetimes as short as a few hours for customers with advanced automation capabilities. Additionally, our CertCentral platform includes the ability to schedule and automate replacement of EV, OV and DV certificates. Using CertCentral admins may take advantage of continuous discovery, renewal notices, thorough API integration and documentation, as well as support for orchestration layers. CertCentral also allows for multi-year purchases to smooth planning and 24/7 global support enabling the best experience in the industry.
>
> As certificate validity periods continue to decrease, automation will be a must for organizations' ability to manage shorter lifetimes. DigiCert is prepared with the industry's most advanced and reliable tools to help our customers take the necessary steps toward greater use of automation.

So, as we can see, Apple's move was considered unilateral because it wasn't the result of a consensus reached among vendors. Though I'd be surprised if there wasn't some behind-the-scenes coordination with other browser vendors. I doubt that everyone was equally surprised by this. The browser makers are and remain adamant that reducing validity is good for security because it reduces the time period in which compromised or bogus certificates can be exploited. And all of our long time listeners will well recall all the noise I've made over the lack of effective certificate revocation in the past. If you can't reliably revoke certificates the next best thing is for them to self-revoke through their own expiration.

So Apple, whose Safari browser currently enjoys an approximate 17% browser market share, decided to simply force the issue by leveraging the power, strength and size of its own closed platform over which it needs consensus from no one. Although there will be grumbling, no one in charge of any website wants to forsake all users of Mac and iOS devices. So that pretty much ends the discussion. Even if certificate authorities were to offer longer-life certificates after this coming August, no one would buy them for a web server.

And, since we are seeing older crypto components, such as SHA-1 signatures, slowly crumbling, forcing a shorter certificate life helps to facilitate a more rapid evolution of certificate crypto.

As we know, certificate renewal has always been a mess. I'm sure we've all bumped into the occasional website whose certificate recently expired, presumably without the webmaster being aware. And this hasn't only happened to small backwater sites. Major sites found themselves without a valid certificate too.

When certificates expired only every 3 years, it was easier to get caught off guard. Presumably, an annual certificate renewal will make certificate maintenance much more of "a thing" which might help to keep it front of mind.

I mentioned previously that even though my own operation is very small, the recent removal from our browsers of any display of EV certification props, the shorter lifetime of EV certs (2 years versus 3 for Organization Validation certs) and the inability of EV certs to support wildcard *.domains, caused me to recently ask DigiCert for a single non-EV cert that I could use everywhere. I'm glad I did.

But many much larger organizations are managing hundreds and sometimes thousands of individual certificates. So, forcing this sort of change could easily double or triple their work as maximum expirations drop from three years to just one.

Presumably the cost of shorter-lived certificates will be cut proportionately, but since some real work is required at the certificate authority's end to validate and verify an organization, if that now needs to be done for every re-issuance, we're requiring a lot more work for less payment.

Dean Tayler's note said: “Before the Apple changes occur, we’ll add the ability for you to purchase multi-year certificate subscriptions to smooth planning and reduce the yearly work of buying and installing certificates. These subscriptions will let you reissue, renew, or replace a certificate as frequently as you need to without incurring additional fees. Our intention with offering subscriptions is to save you time and money.”

So, this probably also means that organization validation will be performed on the same schedule that it was before, but now on “subscription” and “subscription renewal” rather than on “certificate issuance.” DigiCert has clearly been moving in this direction for some time since I was able to issue my own non-EV combo certificate virtually instantly with full automation and without any interaction since they already periodically recertify me and GRC's validity.