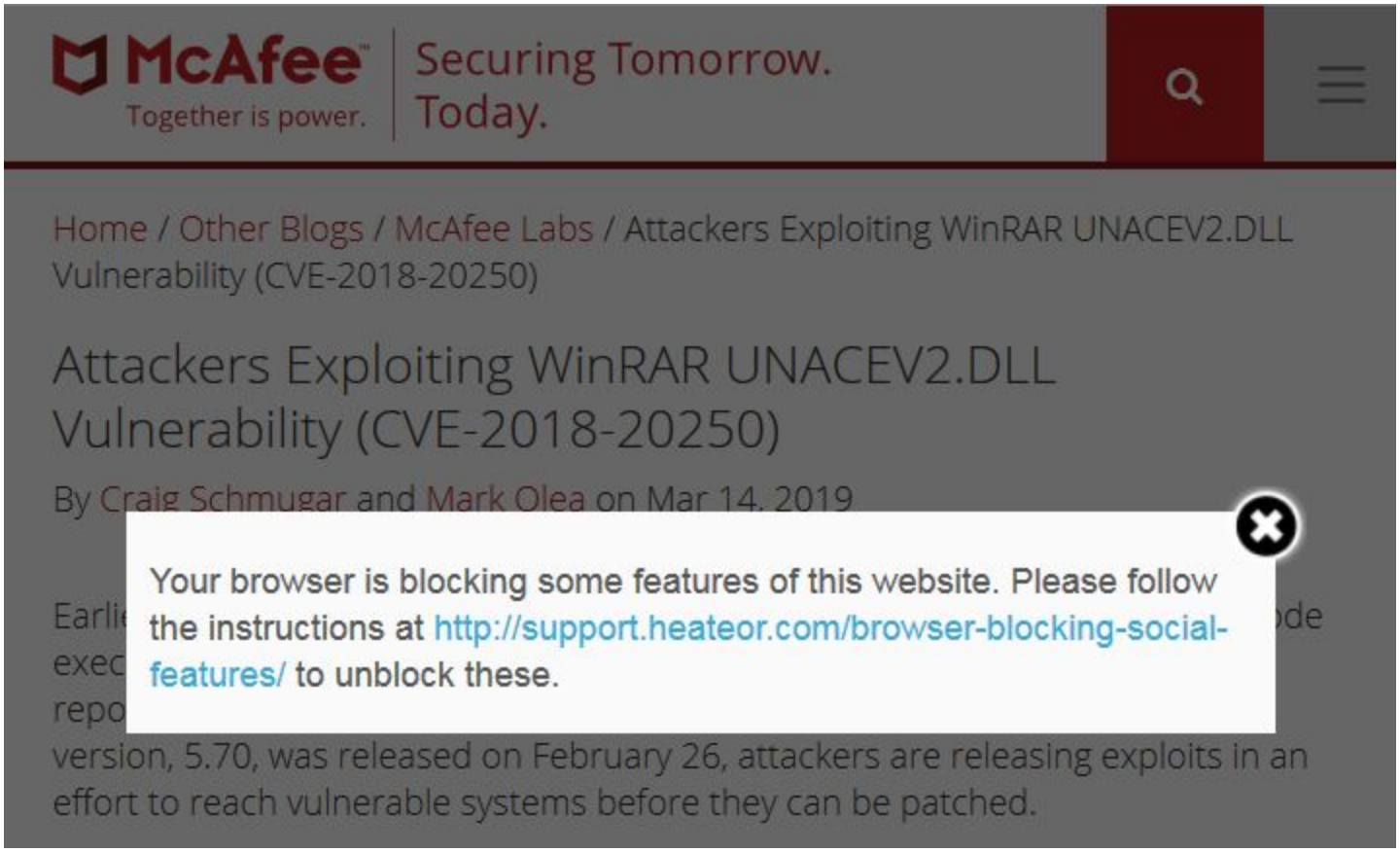
Security Now! #706 - 03-19-19

Open Source eVoting

This week on Security Now!

This week we look back at last week's March Patch Madness, we have an answer about the Win7 SHA256 Windows Update Update, big news regarding the many attacks leveraging the recently discovered WinRAR vulnerability, what happens when Apple, Google and GoDaddy all drop a bit, an update on a big recent jump in Mirai Botnet capability, some worrisome news about compromised Counter-Strike gaming servers, some welcome privacy enhancements coming in the next Android "Q", a pair of very odd web browser extensions for Chrome and Firefox from Microsoft, a bit of follow-up on last week's SPOILER topic, some closing the loop feedback from our terrific listeners, and an early look at a VERY exciting and encouraging project to create an entirely open eVoting system.

I seem to recall that McAfee was a security company once?



Security News

March (Patch) Madness
Last Tuesday was March's patch Tuesday.

Last week we talked about the Google Chrome exploit which was leveraging a pair of 0-day vulnerabilities, one in Chrome and another in Windows. Microsoft had only been informed of the problem the week before, and I wondered whether this would give them time to get it fixed for March. They did and it is.

In total, Microsoft addressed 64 CVE-listed security vulnerabilities in Windows OSes and other products, 17 rated critical, 45 important, 1 moderate and 1 low in severity. They spanned the various operating systems, IE, Edge, Office, SharePoint, the ChakraCore, Skype for Business, and Visual Studio NuGet. Four of the "Important" security vulnerabilities had been disclosed publicly but were not known to be exploited in the wild.

In addition to the one 0-day involved in the actively exploited in the wild Chrome attacks, Microsoft also patches another 0-day that was also under active attack. Both were elevation of privilege flaws residing in the Win32k.SYS driver.

Whereas that first 0-day was only seen to be affecting Win7 systems, the second 0-day elevation of privilege vulnerability, which was also being exploited in the wild, affects all Windows and server additions after Windows 7. So it seems likely that they were closely related and were separate versions of the same problem.

It's also worth noting that last Thursday, March 14th, the Chinese 360 Qihoo 360 Core Security group decided to publish a working proof-of-concept for the Win7 vulnerability. Their release justified this by saying: "Considering that some users are still using Windows 7 [yeah, about half] and this vulnerability combined with Chrome RCE (CVE-2019-5786) has been used for real APT attacks, so this 0day is very likely to be exploited to perform large-scale attacks and pose a real threat [except that it was fixed in Chrome before its announcement]. Therefore, 360 Core Security Technique Center constructed the POC and reproduced the vulnerability triggering process so that security vendors can reference to increase the corresponding protection measures."

The second flaw was detected and responsibly reported to Microsoft by security researchers Vasily Berdnikov and Boris Larin of Kaspersky Labs, who, in a blog post coincident with the patch's release, revealed that the flaw has being actively exploited in targeted attacks by several threat actors including, FruityArmor and SandCat.

ZDNet had an interesting note about the FruityArmor / SandCat connection. They wrote:
The November zero-day (CVE-2018-8589) was also abused by SandCat, a new group on the APT scene about which Kaspersky has few details -- such as its use of the March (CVE-2019-0797) and November (CVE-2018-8589) zero-days, the CHAINSHOT exploit, and the FinFisher/FinSpy hacking framework. What all this tells experts is that there's at least some type of connection between these two APTs -- FruityArmor and SandCat. They are either managed by the same intelligence service, or they're buying Windows zero-days from the same exploit vendor.

Kaspersky wrote: "CVE-2019-0797 is a race condition that is present in the win32k driver due to a lack of proper synchronization between undocumented syscalls NtDCompositionDiscardFrame and NtDCompositionDestroyConnection."

And as for the overall March Madness patching... As usual nearly all of the critical-rated vulnerabilities lead to remote code execution attacks and primarily impact various versions of Windows 10 and Server editions. Most of the flaws reside in Chakra Scripting Engine, VBScript Engine, DHCP Client, and IE.

While some of the "only important" vulnerabilities can also lead to remote code execution attacks, others allow elevation of privilege, information disclosure, and denial of service attacks.

SHA-2 Windows Update Signing Update
I also confirmed that last Tuesday DID update my Win7 machine with the ability to verify the SHA-2 signatures of future Windows updates -- until February 2020 when they are slated to stop. We talked about this before, how the Windows Update signature checking system in Windows 7 machines had not yet been taught to verify Windows Updates signed with SHA-256, and that they were therefore being co-signed with SHA-1 and SHA-256. Since leaving SHA-1 around was less secure than removing it, Microsoft realized that they would need to give Win7 systems the ability to verify SHA-256 signatures a few months before (just to be sure) removing the SHA-1 co-signing from their updates.

Anyway... when we talked about this before it wasn't 100% clear that this update would be automatically included in the monthly updates. So I've been keeping my eye out for that.

So the only takeaway here is that Win7 machines will need to obtain this update by June and before July, which is when all Windows updates will then only be signed with SHA-256. However, I would suspect that even after June, the preceding months of updates, which include KB4474419 and have all still be co-signed could be downloaded and installed, after which all subsequent updates would be available. So I doubt there's any drop-dead effect either way.

https://support.microsoft.com/en-us/help/4474419/sha-2-code-signing-support-update-for-windows-7-and-server-2008-r2

WinRAR users attacked!
Remember that WinRAR vulnerability that I was going on about two weeks ago?
Well, in the first 7 days since the vulnerability was disclosed, McAfee identified more than 100 unique exploits... and rapidly increasing. So in short order this lept from being a theoretical concern to an active threat. To recap just a bit...

A month ago, Check Point Research discovered a long-standing 19-year old "Path Traversal" vulnerability in the decompression module for ACE archives which was, up until their notification of the WinRAR maintainer, present in every version of WinRAR downloaded for the past 19 years. Rarlab sets that number at more than 500 million users of this program.

Rarlab immediately dropped support and with their version, 5.70, released February 26. And when updating to v5.70 over any previous release of WinRAR the two ACE-related files are proactively deleted so there's no chance that they can be used.

But WinRAR doesn't have any form of auto-update mechanism -- and I don't really fault an archiving tool for not auto-updating... except we wish it had one today. WinRAR's lack of any form of vulnerability warning or update system means that hundreds of millions of systems will remain vulnerable and willing to install executable payloads hidden in .ACE archives -- even if they do not carry that file extension -- forever.

One recent example piggybacks on a bootlegged copy of Ariana Grande's hit album "Thank U, Next" with a file name of "*Ariana_Grande-thank_u,_next(2019)_[320].rar*"

When a vulnerable version of WinRAR is used to extract the contents of this archive, a malicious payload is created in the Startup folder behind the scenes. User Account Control (UAC) does not apply, so no alert is displayed to the user. The next time the system restarts, the malware is run.

Many other security groups are seeing the same thing. Windows Defender and 3rd-party antimalware vendors will soon be, if they are not already, detecting ACE format archives and likely looking inside to see whether they contain an exploitation of this path traversal bug.

Note that as of last Friday, which is well downstream of all this, The Hacker News reports that only the infected bootlegged copy of an Ariana Grande's hit album is currently being detected as malware by only 11 security products, whereas 53 antivirus products fail to alert their users of anything being amiss.

Our takeaway is to update to WinRAR v5.70 or later. Anyone using WinRAR likely knows they're using WinRAR, so they will have hopefully, shortly after we talked about it, updated their systems. Unlike many of the problems we cover here, this one won't fix itself.

And while we're on the subject: WinRAR meets Ransomware.
Researchers at Qihoo 360 Threat Intelligence Center spotted, in the wild, an archive called "vk_4221345.rar" that delivers a new Ransomware malware payload they named JNEC.a.

When the RAR archive's contents are extracted with a vulnerable version of WinRAR, the ransomware encrypts the user's data on their computer, appending the .Jnec extension to the file's original extension. The extortion price for the decryption key is 0.05 bitcoins. With Bitcoin now persistently hovering around $4,000, that's about $200 USD.

GoDaddy, Apple, and Google misissue more than 2 million certificates!
This falls under the heading: "One thing leads to another."

Remember our recent discussion about the sketchy wannabe UAE-based Certificate Authority, which decided to name themselves "DarkMatter" and was now appealing to Mozilla to have their CA signing public key added into Mozilla's root certificate store?

Well, during a discussion on the mozilla.dev.security.policy group about Darkmatter's application to become a fully fledged cert-issuing CA, people poking into DarkMatter's existing counter-signed certs happened to discover that the company's supposedly 64-bit serial numbers in its certificates were actually one bit short. But then, engineers at other organizations who read the thread, realised that their OWN certificates were similarly affected. Whoops. One thing leads to another.

So what's behind this broadly made mistake?

As a consequence of everyone using the non-RFC-compliant default setting in a commonly used open source certificate serial number generator (known as EJBCA), GoDaddy, Apple and Google are now facing the revocation and reissuance of more than 2 million certificates. GoDaddy alone estimated that they had issued 1.8 million certificates.

In my opinion this is a tempest in a teapot.

RFC 5280: https://tools.ietf.org/html/rfc5280

Title: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"

CertificateSerialNumber ::= INTEGER

Appendix B. ASN.1 Notes

CAs MUST force the serialNumber to be a non-negative integer, that is, the sign bit in the DER encoding of the INTEGER value MUST be zero. This can be done by adding a leading (leftmost) `00'H octet if necessary. This removes a potential ambiguity in mapping between a string of octets and an integer value.

As noted in Section 4.1.2.2, serial numbers can be expected to contain long integers. Certificate users MUST be able to handle serialNumber values up to 20 octets in length. Conforming CAs MUST NOT use serialNumber values longer than 20 octets.

So what's this EJBCA? https://www.ejbca.org/ by "PrimeKey"

EJBCA - The Open Source CA
EJBCA® is a PKI Certificate Authority software, built using Java (JEE) technology. Robust, flexible, high performance, scalable, platform independent, and component based, EJBCA can be used stand-alone or integrated with other applications.

Extremely scalable and flexible, EJBCA is suitable to build a complete PKI infrastructure for any large enterprise or organization. If you only want to issue a few single certificates for testing, there are probably other options that will get you started quicker, but if you want a serious Certificate Authority to manage your Public Key Infrastructure, we recommend EJBCA.

WikiPedia:
EJBCA, is a free software public key infrastructure (PKI) certificate authority software package maintained and sponsored by the Swedish for-profit company PrimeKey Solutions AB, which holds the copyright to most of the codebase. The project's source code is available under terms of the Lesser GNU General Public License.

The system is implemented in Java EE and designed to be platform independent and fully clusterable,[1] to permit a greater degree of scalability than is typical of similar software packages. Multiple instances of EJBCA are run simultaneously, sharing a database containing the current certificate authorities (CAs). This permits each instance of the software to access any CA. The software also supports the use of a hardware security module (HSM), which provides additional security. Larger-scale installations would use multiple instances of EJBCA running on a cluster, a fully distributed database on a separate cluster and a third cluster with HSMs keeping the different CA keys.

EJBCA supports many common PKI architectures[2] such as all in a single server, distributed RAs and external validation authority.

.....

Unfortunately, until recently this very nice-looking and capable Open Source CA system was configured to generate 8-octet serial numbers. 8 octets is 8 bytes, which is 64 bits. Serial number fields are defined as signed integers. This must be some nutty committee decision, since a negative serial number makes no sense. But that's what it is.

Let’s take a moment to review binary number format…

In the industry standard 2's-complement math, the number ZERO is represented as all binary bits being zero. If you increment the binary value the first bit turns to a 1. Incrementing it again and the first bit goes back to zero and the second bit goes to 1. Now suppose that we return to a value of zero with all binary bits being zero... And we DECREMENT that zero value? What we get is ALL 1's. In other words, in standard 2's-complement binary, when interpreting a binary value as "signed" (meaning capable of representing both positive and negative numbers), all 1's is the value of -1. If we decrement again to -2, the least significant binary bit goes to a zero.

Notice that if we start at the value of ZERO and increment, the lowest bits start changing. And if we start at ZERO and decrement, everything goes to 1's and then if we keep going the lowest bits start changing.

If you think about this for a minute you'll see that the farthest left binary bit, the most significant bit, can be viewed as the SIGN of the value. If the binary value is positive the "sign bit" -- the leftmost bit -- will be zero. And if the binary value is negative the "sign bit" will be set to '1'.

So... Suppose that we wanted to generate an 8-byte random serial number, using signed 2's complement binary representation, that was also GUARANTEED to be positive (because negative serial numbers make no sense). We would use our system's source of entropy to generate 8 random bytes to form a 64-bit value. Then we would simply turn off the highest bit, the sign bit, to create a positive value.

So this is great, right?

But how much true entropy does the resulting serial number contain? We started with 64 bits. Then we forced one of them to always be off, killing its entropy. So now we have 63 bits.

And everything would be fine except that Section 7.1 of the CA Baseline Requirements states that certificate serial numbers must contain a minimum of 64 bits of entropy. Whoops!

The reason I said that I felt this was a tempest in a teapot is that we're talking about being 1-bit shy of the spec. Yes, a spec is a spec. And requirements are requirements. But all of those mis-issued certs were going to expire themselves after two or three years. And one bit less entropy in certificate serial numbers just isn't a huge cause for concern.

The concern with serial numbers is that they are sometimes used to "pin" certificates, though using the certificate's much longer thumbprint hash makes more sense. So we're concerned with the chance that another randomly-generated certificate's serial number might have the same serial number as the one we have pinned.

With the classic "Birthday Paradox" we would be asking "given a pool of certificates of a certain size, what's the likelihood of ANY two certificates having the same serial number." What's surprising there is how quickly that probability falls as the pool size increases. This happens since every certificate serial number is being compared with every other.

But that's not the worry here. In this case we're just worried that another certificate's serial number might have the same serial number as the one we care about. So the math is simple:

The CAB spec wants that two-cert serial number collision probability to be 1 in 2^64. That's one chance in 18,446,744,073,709,551,616. Naming large numbers we have million, billion, trillion, quadrillion, and finally quintillion. So with the full 64 bits the chance of collision with another randomly-generated certificate is 1 in 18.446 quintillion.

When we lose a bit, by forcing the first bit to be '0', we, of course, cut the total universe of possible serial numbers in half, thus doubling the collision probability from that 1 in 18.446 quintillion to only 1 in 9.223 quintillion. I just cannot get very worked up about that. Given that these certs will all be expiring themselves in two or three years, this problem would be resolving itself. And we know how well certificate revocation works!

So, when I looked at the EJBCA home page I smiled when I noted that the top of the News Feed said...

EJBCA can be configured to generate certificate serial numbers (positive integers) from 4 to 20 octets.

Posted: 2019-03-13

CONFIGURABLE SN ENTROPY, DEFAULT VALUE RAISED TO 20 OCTETS
CA/B Forum requires the use of 64 bit entropy when generating serial numbers (see CABF Ballot 164). Due to only positive values being valid serial numbers, 8 octets will only result in 63 bit entropy as the most-significant-bit will always be 0, hence we recommend larger sizes than 8 octets. Previously this was set using the property ca.serialnumberoctetsize in cesecore.properties, which has now been dropped and the value is instead set directly in the CA.

[In other words, it wasn't readily and explicitly configurable before. It is now.]

Possible values may range between 4 and 20 octets, and the default for all new CAs is 20 while upgraded CA's will retain whatever value was set in ca.serialnumberoctetsize, or 8 if none was set.

The Mirai Botnet is alive and well, and more scary and capable than ever:
https://unit42.paloaltonetworks.com/new-mirai-variant-targets-enterprise-wireless-presentation-display-systems/

The virulent Internet of Things malware, Mirai, which broke DDoS attack size in 2016 when it was used to attack Brian Krebs, the French Web host OVH and most famously the DNS provider Dyn, has been updated to target a new crop of devices, including two which are often found inside enterprise networks, where, as we know, bandwidth is often plentiful.

Mirai knows how to infect webcams, routers, DVRs, and many other Internet-connected devices, which typically ship with default credentials and run never-updated and thus woefully outdated versions of Linux.

Yesterday morning, Palo Alto Networks' Unit 42 posted this news of a new Mirai titled: "New Mirai Variant Targets Enterprise Wireless Presentation & Display Systems". I've edited a bit for length and clarity in this context:

The Mirai variant that Unit 42 discovered is notable for targeting different embedded devices like routers, network storage devices, NVRs (network video recorders), and IP cameras and using numerous exploits against them.

Specifically, Unit 42 found this new variant targeting WePresent WiPG-1000 Wireless Presentation systems, and in LG Supersign TVs. Both these devices are intended for use by businesses. This development indicates to us a potential shift to using Mirai to target enterprises. Attack code exploiting a WePresent command-injection vulnerability was published in 2017, while a remote code execution exploit for LG Supersign TVs has been available since last September. After being packaged into this new Mirai variant, the exploits become much more likely than previously to actively be used to compromise their vulnerable devices.

And this is not the first time Mirai has been aimed at enterprise networks. Last September, Palo

Alto Networks reported that Mirai was found targeting the same Apache Struts vulnerability hackers exploited to breach Equifax.

In addition to this newer targeting, this new variant of Mirai incorporates 11 new exploits in its multi-exploit kit, and new credentials to use in brute forcing device sign on.

These new features provide the botnet with a large attack surface. In particular, targeting enterprise links also grants it access to larger bandwidth, ultimately resulting in greater firepower available to the botnet for DDoS attacks.

These developments underscore the importance for enterprises to be aware of the IoT devices on their network, change default passwords, ensure that devices are fully up-to-date on patches. And in the case of devices that cannot be patched, to remove those devices from the network as a last resort.

In addition to the 11 newly added exploits, this latest sample retains 16 which were previously seen.

Mirai continues to encrypt its password stuffing credential table with the HEX key: 0xbeafdead. And when that key is used on this new variant, Palo Alto Networks found four new username:password combination:

- admin:huigu309
- root:huigu309
- CRAFTSPERSON:ALC#FGU
- root:videoflow

And, as we know, in addition to scanning for and spread itself to other vulnerable devices, the new version can be commanded to send out HTTP Flood DDoS attacks. The Mirai worm uses the domain epicrustserver[.]cf at port 23823 is for its Command & Control communications.

Counter-Strike Strikes its players

The malicious network has been taken down now, but there's an important lesson to be learned in its aftermath:

Malicious Counter-Strike 1.6 servers used 0-days to infect users with malware. The Russian antivirus firm, Dr.Web found that 39% (1,951) of all Counter-Strike 1.6 servers were malicious and were trying to infect their users with malware.

https://st.drweb.com/static/new-www/news/2019/march/belonard_trojan_en.pdf

Dr.Web wrote in their report:

Introduction

The game Counter-Strike 1.6 was released by Valve Corporation back in 2000. Despite its rather considerable age, it still has a large fan base. The number of players using official CS 1.6 clients

reaches an average of 20,000 people online, while the overall number of game servers registered on Steam exceeds 5,000. Selling, renting, and promoting game servers is now deemed an actual business, and these services can be purchased on various websites. For example, raising a server's rank for a week costs about 200 rubles ($3.11 USD), which is not much, but a large number of buyers make this strategy a rather successful business model.

Many owners of popular game servers also raise money from players by selling various privileges such as protection against bans, access to weapons, etc. Some server owners advertise themselves independently, while others purchase server promotion services from contractors. Having paid for a service, customers often remain oblivious as to how exactly their servers are advertised. As it turned out, the developer nicknamed, "Belonard", resorted to illegal means of promotion. His server infected the devices of players with a Trojan and used their accounts to promote other game servers.

The owner of the malicious server uses the vulnerabilities of the game client and a newly written Trojan as a technical foundation for their business. The Trojan is to infect players' devices and download malware to secure the Trojan in the system and distribute it to devices of other players. For that, they exploit Remote Code Execution (RCE) vulnerabilities, two of which have been found in the official game client and four in the pirated one.

Once set up in the system, the Belonard Trojan replaces the list of available game servers in the game client and creates proxies on the infected computer to spread the Trojan. As a rule, proxy servers show a lower ping, so other players will see them at the top of the list. By selecting one of them, a player gets redirected to a malicious server where their computer becomes infected with Trojan.Belonard.

Using this pattern, the developer of the Trojan managed to create a botnet that makes up a considerable part of the CS 1.6 game servers. According to our analysts, out of some 5,000 servers available from the official Steam client, 1,951 were created by the Belonard Trojan. This is 39% of all game servers. A network of this scale allowed the Trojan's developer to promote other servers for money, adding them to lists of available servers in infected game clients.

We previously reported a similar incident with CS 1.6, where a Trojan could infect a player's device via a malicious server. However, a user then had to approve the download of malicious files, while this time, a Trojan attacks devices unnoticed by the users. Doctor Web have informed Valve about these and other vulnerabilities of the game, but as of now, there is no data on when the vulnerabilities will be fixed.

Trojan.Belonard consists of 11 components and operates under different scenarios, depending on the game client. If the official client is used, the Trojan infects the device using an RCE vulnerability, exploited by the malicious server, and then establishes in the system. A clean pirated client is infected the same way. If a user downloads an infected client from the website of the owner of the malicious server, the Trojan's persistence in the system is ensured after the first launch of the game.

There's really nothing that a user might do to protect themselves from this threat. Perhaps stick to well known and trusted game servers and avoid the siren's call of a proxy that promises a lower ping time. I suppose the best takeaway is to maintain an awareness that this kind of thing IS going on and be a bit more cautious and suspicious than you might otherwise be.

Here comes Android "Q" with a bunch of new privacy features.
(What tasty treat begins with the letter 'Q' ?)

Android Q will finally be delivering MAC address randomization, new location data permission popup, no more clipboard sniffing.

The beta of "Q" was first released last week and it promises a bunch of welcome privacy improvements:

Access to clipboard data - Android apps can no longer access the Android operating system's clipboard data unless they are in focus (running in the foreground on screen). The exception to this is the current default input method editor -- such as the default keyboard.

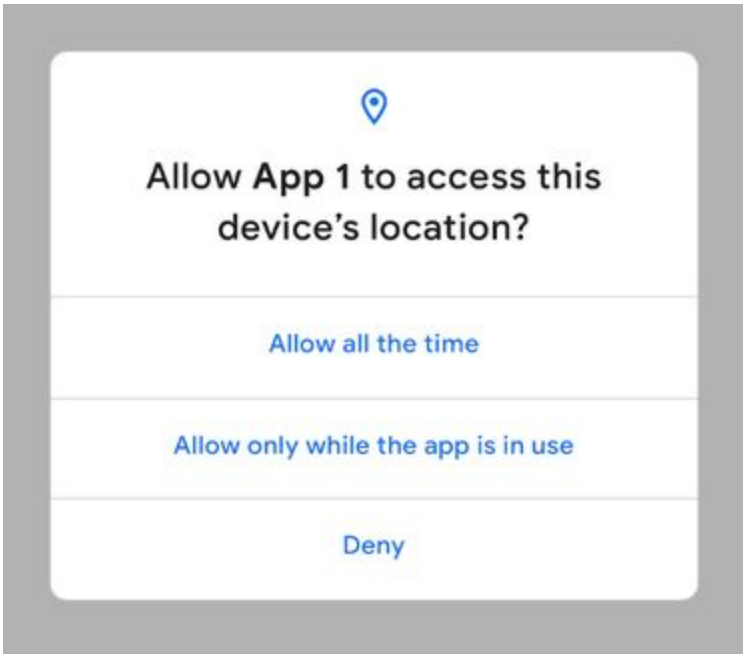
MAC address randomization on by default - Google introduced MAC address randomization in Android 6.0, but devices broadcast a random MAC address only when the smartphone would initiate a background Wi-Fi or Bluetooth scan. Android Q devices will now transmit a randomized MAC address by default, at all times, and for all communications. (This is even better than iOS, which, last time I looked, still reverts to its device's factory-set fixed MAC address.) And despite security researchers have shown that they can still track devices with randomized MAC addresses, support for this feature will reduce the efficiency of some data harvesting and user tracking operations.

Removing easy access to network data - Android Q will also remove the /proc/net function that gives out information about the device's network state. App developers have other alternatives, but those are safeguarded by permissions, which means that the very low hanging fruit used by some data harvesters has been removed.

And, similarly, easy access to device details is being curtailed. Starting with Android Q, Google will require app developers to request a special permission before they can access what Google calls "non-resettable device identifiers" such as the device's IMEI and serial number.

Non-ranked contact data - Google has also decided that Android Q will stop tracking contacts based on the frequency of interaction. Any app that received the permission to access the user's contacts will only get non-ranked contacts going forward.

And, finally, more control over location data - Android Q receives a new permissions pop-up prompt for accessing location data. Beginning with "Q", users will be able to give apps access to location data all the time or only when the app is in focus (in the foreground):



Headline: "Microsoft releases Application Guard Extensions for Chrome & Firefox"
https://www.zdnet.com/article/microsoft-releases-application-guard-extension-for-chrome-and-firefox/

When I first encountered the headline I thought "Wow! How cool! This really IS a new Microsoft!"

Across the tech press, the coverage of this begins by saying something like:

"Microsoft has released browser extensions, one for Chrome and another for Firefox, which port the Windows Defender Application Guard technology from Edge to Chrome and Firefox.

The extensions only work for Chrome and Firefox running on current Windows Insider builds, but are expected to work with the upcoming Windows 10 stable release, 19H1, scheduled for release later this spring.

The Windows Defender Application Guard technology is a relatively new Windows Defender security feature that until now has only been available to Edge users."

So I'm thinking... *"Wow! Microsoft is willing to protect Chrome and Firefox users from malicious web content. How amazing is that??!!?"*

But then reading into this a bit further you encounter:
"When using Chrome or Firefox, administrators can establish a list of trusted websites and local resources that the user can access within those browsers. But when a user of Chrome or Firefox attempts to visit any URL not on the trusted list, Windows Defender Application Guard comes into effect by launching a sandboxed session of Edge (within a Hyper-V-enabled container) where the untrusted website will be loaded into a safe environment within the Edge browser and isolated from the rest of the underlying operating system."

I had to read that several times to be sure I wasn't missing something.

Microsoft's own blog posting from last Friday, March 15, 2019 2:02 pm was titled: "Announcing Windows 10 Insider Preview Build 18358"

https://blogs.windows.com/windowsexperience/2019/03/15/announcing-windows-10-insider-preview-build-18358/

The section about this is titled: "Windows Defender Application Guard as browser extensions in Google Chrome and Mozilla Firefox."

To extend our container technology to other browsers and provide customers with a comprehensive solution to isolate potential browser-based attacks, we have designed and developed Windows Defender Application Guard extensions for Google Chrome and Mozilla Firefox.

This way, any potential attack won't be able to reach and grab the user's data, or plant malware on the local operating system.

How it works:

The extensions for Google Chrome and Mozilla Firefox automatically redirect untrusted navigations to Windows Defender Application Guard for Microsoft Edge. The extension relies on a native application that we've built to support the communication between the browser and the device's Application Guard settings.

[In other words, Windows Defender Application Guard knows that the world is a scary place. And should you attempt to venture out there with Chrome or Firefox, this nifty new web browser extension will jump in to protect you from your wayward wanderings, taking you instead into Microsoft's proprietary Hyper-V VM where, from the safety of their Edge browser you can peer out into the gloom which is the Internet! How very thoughtful!]

When users navigate to a site, the extension checks the URL against a list of trusted sites defined by enterprise administrators. If the site is determined to be untrusted, the user is redirected to an isolated Microsoft Edge session. In the isolated Microsoft Edge session, the user can freely navigate to any site that has not been explicitly defined as trusted by their organization without any risk to the rest of system. With our upcoming dynamic switching capability, if the user tries to go to a trusted site while in an isolated Microsoft Edge session, the user is taken back to the default browser.

[How thoughtful! ... in the future, once the upcoming dynamic switching capability is in place, it will even return you to your original browser once you have navigated yourself back to safety and away from the scary Internet. That's SO much better than just fixing the bugs in Windows.]

AMD Believes SPOILER Vulnerability Does Not Impact Its Processors
https://www.bleepingcomputer.com/news/security/amd-believes-spoiler-vulnerability-does-not-impact-its-processors/

Just to confirm what we reported last week about "SPOILER", AMD assert that its processors are not impacted by the SPOILER vulnerability which employs speculative execution to improve the efficiency of memory and cache attacks such as Rowhammer. Specifically, AMD states that its processors are not affected by SPOILER because they do not use "partial address matches above address bit 11" when having to address load conflicts:

<quote> We are aware of the report of a new security exploit called SPOILER which can gain access to partial address information during load operations. We believe that our products are not susceptible to this issue because of our unique processor architecture. The SPOILER exploit can gain access to partial address information above address bit 11 during load operations. We believe that our products are not susceptible to this issue because AMD processors do not use partial address matches above address bit 11 when resolving load conflicts. </quote>

Closing The Loop

Lawrence in Philadelphia
Subject: Making Windows 10 Usable
Date: 17 Mar 2019 10:03:52
:
Steve, I've heard you mention at least a few times that, in making peace with Windows 10, you have done a bunch of things to strip out the junk and make it into a usable operating system. I would love some guidance on how to make it actually functional? Would you share some instructions on the show or provide some sort of checklist of what steps need to be taken? I dream of some tool like your "Never 10" program that simply fixes everything with one click but I suspect that this is not so easy and I know you're tied up with new versions of SpinRight and SQRL. Thanks for all your hard work and dedication.

(Never10 -- more than 3 Million downloads.)

(DNS Benchmark -- at 4,738,347 downloads and 2,142/day.)

A SpinRite user wrote: "Please don’t put off native usb support to later versions."

Open Source eVoting

https://motherboard.vice.com/en_us/article/yw84q7/darpa-is-building-a-dollar10-million-open-source-secure-voting-system

The headline on the Motherboard website read:
DARPA Is Building a $10 Million, Open Source, Secure Voting System

The system will be fully open source and designed with newly developed secure hardware to make the system not only impervious to certain kinds of hacking, but also allow voters to verify that their votes were recorded accurately.

For years security professionals and election integrity activists have been pushing voting machine vendors to build more secure and verifiable election systems, so voters and candidates can be assured election outcomes haven't been manipulated.

[I would argue that the ONLY POSSIBLE WAY to have voting machines is to have an absolute lack of proprietary technology. It must be 100% open and published and accessible. I have no problem with the likes of Diebold manufacturing instances of a fully open reference design system. But it must be public and provably implemented.]

Now, thanks to a new $10 million contract DARPA has launched to design and build a secure voting system that it hopes will be impervious to hacking… this has the chance to happen.

The first-of-its-kind system will be designed by an Oregon-based firm called Galois, a longtime government contractor with experience in designing secure and verifiable systems. The system will use fully open source voting software, instead of the closed, proprietary software currently used in the vast majority of voting machines, which no one outside of voting machine testing labs can examine. More importantly, it will be built on secure open source hardware, made from secure designs and techniques developed over the last year as part of a special program at DARPA. The voting system will also be designed to create fully verifiable and transparent results so that voters don't have to blindly trust that the machines and election officials delivered correct results.

But DARPA and Galois won't be asking people to blindly trust that their voting systems are secure—as voting machine vendors currently do. Instead they'll be publishing source code for the software online and bring prototypes of the systems to the Def Con Voting Village this summer and next, so that hackers and researchers will be able to freely examine the systems themselves and conduct penetration tests to gauge their security. They'll also be working with a number of university teams over the next year to have them examine the systems in formal test environments.

Linton Salmon is the program manager for DARPA's Microsystems Technology Office which is overseeing the project. In a phone call he told Motherboard: "Def Con is great, but [hackers there] will not give us as much technical details as we want [about problems they find in the systems]," "Universities will give us more information. But we won't have as many people or as high visibility when we do it with universities."

The systems Galois designs won't be available for sale. But the prototypes it creates will be available for existing voting machine vendors or others to freely adopt and customize without costly licensing fees or the millions of dollars it would take to research and develop a secure system from scratch.

Linton said: "We will not have a voting system that we can deploy. That's not what we do. We will show a methodology that could be used by others to build a voting system that is completely secure."

[.....]

Joe Kiniry is the principal scientist at Galois who is leading the project at his company. He said that Galois will design two basic voting machine types:

The first will be a ballot-marking device that uses a touch-screen for voters to make their selections. That system won't tabulate votes. Instead it will print out a paper ballot marked with the voter's choices, so voters can review them before depositing them into an optical-scan machine that tabulates the votes. Galois will bring this system to Def Con this year.

Many current ballot-marking systems on the market today have been criticized by security professionals because they print bar codes on the ballot that the scanner can read instead of the human-readable portion voters review. Someone could subvert the bar code to say one thing, while the human-readable portion says something else. Kiniry said they're aiming to design their system without barcodes.

The optical-scan system will print a receipt with a cryptographic representation of the voter's choices. After the election, the cryptographic values for all ballots will be published on a web site, where voters can verify that their ballot and votes are among them.

Kiniry said: "That receipt will not permit you to prove anything about HOW you voted, but it permits you to prove that the system accurately captured your intent and that your vote is in the final tally."

Members of the public will be able to use the cryptographic values to independently tally the votes to verify the election results so that tabulating the votes isn't a closed process solely in the hands of election officials.

"Any organization [interested in verifying the election results] that hires a competent software engineer [can] write their own tabulator," Kiniry said. "We fully expect that Common Cause, League of Women Voters and the [political parties] will all have their own tabulators and verifiers."

The second system Galois plans to build is an optical-scan system that reads paper ballots marked by voters by hand. They'll bring that system to Def Con next year.

~30~