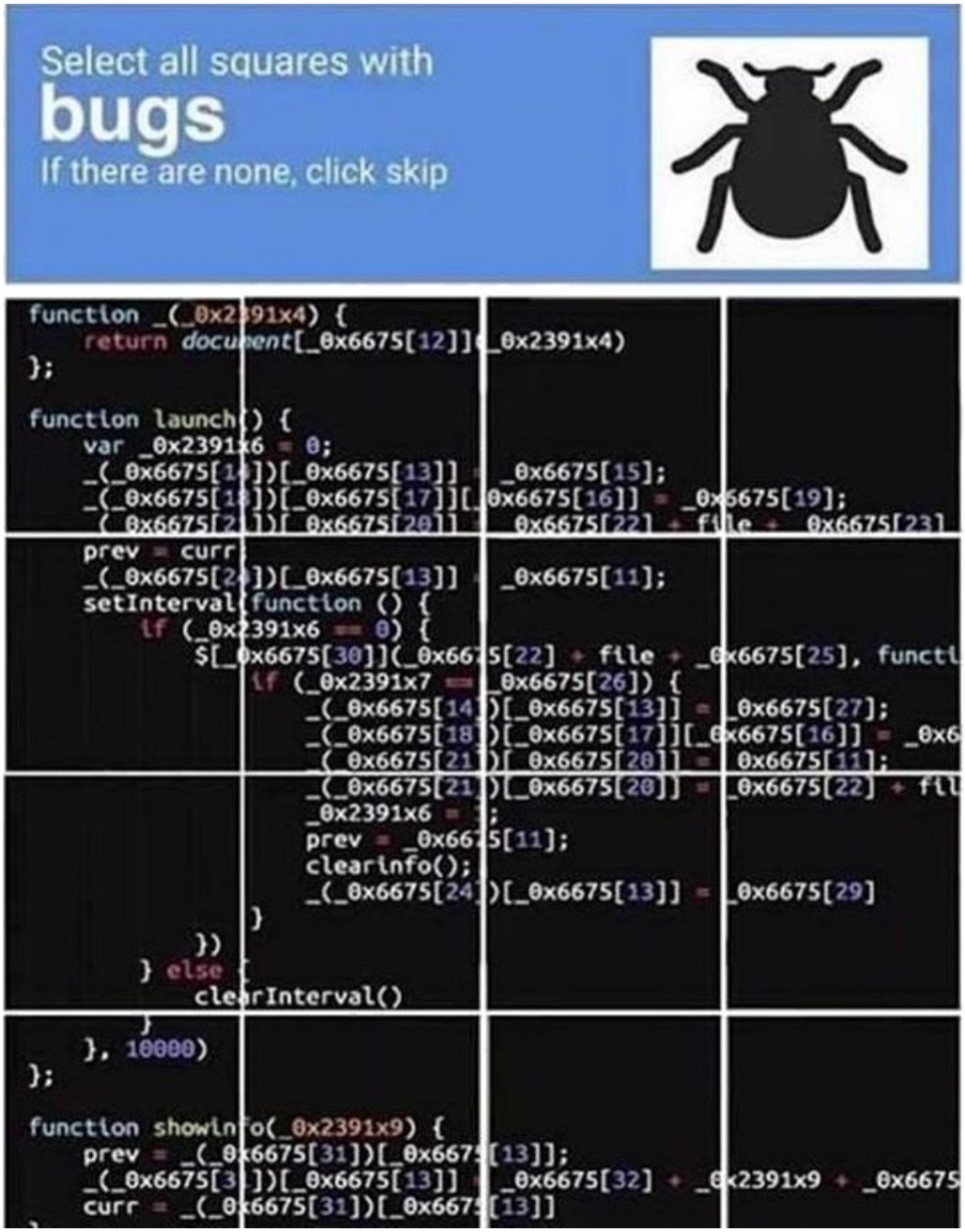
Security Now! #753 - 02-11-20

Promiscuous Cookies

This week on Security Now!

This week we offer some welcome news about Microsoft A/V under Windows 7, we follow even more blow-by-blow consequences of January's final updates for Windows 7, we look at a worrisome exploitable Bluetooth bug Google just fixed in Android and what it means for those not fixed, we update on the ClearView AI face scanning saga, we take a peak into data recovery from physically destroyed phones, we entertain yet another whacky data exfiltration channel, and we conclude by looking at the consequences of the recent changes to make cookies mess promiscuous.



Security News

Whoa! We get to REMAIN with Security Essentials under Windows 7!

This observation we credit to Elaine. When sending last week's transcript to me, she added:

> Hi Steve,
>
> I've been using MS Security Essentials since it came out, through XP and 7. Wikipedia says: "Although support for Windows 7 ended on January 14, 2020 and MSE is no longer available to download, Microsoft will continue to update virus definitions for existing users until 2023."
>
> Guess I'm good for a while. I still get new definitions every night.
>
> And I can confirm that I, too, am still getting nightly updates and that my MSE is continuing to scan and protect my Win7 machine!

https://support.microsoft.com/en-us/help/17150/windows-7-what-is-microsoft-security-essentials

Microsoft says:

> Microsoft Security Essentials reached end-of-service on January 14, 2020 and is no longer available as a download. Microsoft will continue to release signature updates (including engine) to service systems currently running Microsoft Security Essentials until 2023.
>
> ***Why is Microsoft Security Essentials no longer available?***
> Windows 7 is no longer supported and availability of new installations of Microsoft Security Essentials has ended. We recommend all customers move to Windows 10 and Windows Defender Antivirus for our best security option.
>
> ***Will Microsoft Security Essentials running on my system continue to run?***
> Yes, we will continue to provide signature updates for Microsoft Security Essentials until 2023.

It MUST BE that our listeners have been waving their arms at me in Twitter. I've been so focused and busy that I haven't been keeping up. So… thank you to everyone who was almost certainly trying to let me know that there would be NO NEED to go find some other A/V, nor to go "Commando" and hope for the best.

Microsoft drops a fix for the wallpaper stretch black screen

https://support.microsoft.com/en-us/help/4539602/wallpaper-set-to-stretch-is-displayed-as-black

Last Friday, February 7th, in an out-of-cycle update, Microsoft dropped a fix to repair the Windows desktop bitmap image stretch problem that was introduced with last month's final Patch Tuesday update.

The update was titled: *"Wallpaper set to Stretch is displayed as black in Windows 7 SP1 and Server 2008 R2 SP1"*

Summary
This update resolves the following issue:

Addresses an issue that might cause your wallpaper that is set to Stretch to display as black. Important Before you apply this update, see the "Prerequisites" section.

Known issues in this update
We are currently not aware of any issues that affect this update.

How to get this update
Microsoft Update Catalog
To get the stand-alone package for this update, go to the Microsoft Update Catalog website.

Prerequisites
You must have the following updates installed before you apply this update. If you use Windows Update, these updates will be offered automatically as needed.

You must have the SHA-2 update (KB4474419) that is dated September 23, 2019 or a later SHA-2 update installed and then restart your device before you apply this update. If you use Windows Update, the latest SHA-2 update will be offered to you automatically. For more information about SHA-2 updates, see 2019 SHA-2 Code Signing Support requirement for Windows and WSUS.

You must have the servicing stack update (SSU) (KB4490628) that is dated March 12, 2019 or a later SSU update installed. For more information about the latest SSU updates, see ADV990001 | Latest Servicing Stack Updates.

Important You must restart your device after you install these required updates and before you apply any Monthly Rollup, Security-Only Update, Preview of Monthly Rollup, or stand-alone update.

That update was titled: *"Wallpaper set to Stretch is displayed as black in Windows 7 SP1 and Server 2008 R2 SP1"* … which is significant because today we learned that...

"Can't Boot This!"
Windows Server 2008 R2 won't boot after installing the KB4539602 update!

Believe it or not, on any instance of Windows server 2008 R2 which is lacking those prerequisite updates I noted above, the consequence of attempting to install KB4539602 isn't a notice of an update failure, or a nice mention that some prerequisite updates are missing. No… the result is a fully BRICKED server!

For reasons only Microsoft knows, attempting to fix the desktop wallpaper stretching issue introduced the previous month on Windows Server 2008 R2 results in the deletion of two critical boot files "winload.efi" and "winload.exe" from the server's C:\windows\system32\ directory.

Windows 2008 R2 servers have been getting bricked left and right since Friday and the community finally figured out what was going on. Those two files need to be copied back into the C:\Windows\System32\ directory from another installation, or the system must be rolled back using the system imaging command. You can boot into "System Recovery" then issue the following command against the proper system drive letter:

dism.exe /image:C:\ /cleanup-image / revertpendingactions

Or, boot into System Recovery and, as I mentioned, copy those two files from another instance.

But these are not the only consequences of January's troubled final update.

"Can't Shut This!"...
We know that the "final" (?) Patch Tuesday update for Windows 7 broke desktop wallpaper stretch, treating longtime Windows users to a black screen. But another more serious and less cosmetic problem has since surfaced: You gotta love the irony of this one since Microsoft has been frantically working to push everyone off of Windows 7 and over to Windows 10. But now some Windows 7 users are being told: "You don't have permission to shut down this computer."

A number of workarounds have been found, such as logging off rather than shutting down, then using the power switch in the lower-right corner of the screen... and several others. For example, there's a Group Policy tweak. But since this glitch is more than simply cosmetic it'll be interesting to see whether Microsoft issues another out-of-cycle manual download patch, or whether Windows 7 might be treated to something… today for Tuesday's February patches.

Win10 Firefox users being "reminded" about Edge
While I'm on the topic of Microsoft and Windows 10... I suppose anyone who hasn't deliberately turned off the "suggestions" option for their Windows 10 Start Menu may have left it on because they are interested in what Microsoft might have to suggest. And, after all, the OS is now free and is intended to be a source of "significant marketing and profit opportunities moving forward" as we'll recall Microsoft once bragged in an annual report to shareholders a few years back...

Still, it's not for the sake of running Windows that we run Windows, exciting and harrowing though it can sometimes be. No... Windows is a means to an end. It exists to host and launch other programs. It's an operating system. So it does seem a bit unseemly for people with their start menu suggestions still enabled, to receive the following selective notice when Firefox is their deliberately chosen browser. The notice reads: "Still using Firefox? Microsoft Edge is here."

Every week these notes are authored in Google Docs on Chrome alongside a Firefox browser with a long vertical column of tabs and an instance of the ThoughtManager Desktop outliner app. I'm working towards a tentative peace with Windows 10 because Microsoft hasn't left me with any practical choice. But I don't need any help choosing the best tool for the job. I'm delighted that Edge has incorporated Chromium. But to answer your question... Yes, Microsoft, I'm still using Firefox.

Last week Google closed an Android RCE flaw in the BlueTooth daemon.
The flaw was just patched by Android's February security update.

We've been encountering Bluetooth flaws recently. And while they are not good because they are potentially "hands off" and "at a distance" the deliberately lower power and short range operation of Bluetooth tends to limit their exportability and severity. WiFi vulnerabilities are worse and Internet TCP flaws are worse still. But in this case a critical vulnerability was found and fixed in the Bluetooth implementation on Android devices which could allow attackers to launch remote code execution (RCE) attacks without any user interaction.

Last Thursday, after the patch had been pushed out, researchers revealed additional (but not all) details behind the critical Android flaw, tracked as CVE-2020-0022. It poses a potential critical-severity threat to Android versions Pie (9.0) and Oreo (8.0, 8.1), which account for almost two-thirds of Android devices today when they have enabled Bluetooth... As they likely would.

Against those versions, researchers said that a remote attacker within Bluetooth range can silently execute arbitrary code with the privileges of the Bluetooth daemon. The flaw is worrisome because no user interaction is required and only the Bluetooth MAC address of the target devices needs to be known to launch an attack. And, for many devices, the Bluetooth MAC address can be deduced from the WiFi MAC address which devices promiscuously broadcast.

The same vulnerability impacts Google's most recent Android version, Android 10. However, with Android 10, the severity rating is moderate rather than critical because the impact is not a RCE, but only a denial of service resulting from the crash of the Bluetooth daemon. And, Android versions older than 8.0 were not tested but they might also be affected.

The flaw's discoverers said that once they are "confident" all patches have reached the end users, they will publish a technical report on the flaw that includes a description of the exploit as well as proof-of-concept code.

The trouble is, all of us here know that a great many Android devices running Oreo and Pie are never going to receive an update. So they will now, and probably forever, be vulnerable to the possibility of an engineered proximity takeover and malware installation. And completely

descriptive documentation with a working proof of concept will be available shortly. This is PRECISELY the sort of powerful and persistent vulnerability hostile powers love to find and exploit.

We've said it before but it bears repeating: Today's smartphones are seen by bad guys as a huge target of opportunity. And just as no one today wants to use an operating system that's no longer receiving security updates, people should be reluctant in the extreme to use any smartphone whose manufacturer does not have a solid track record of providing updates. It's true that such after sale support comes at a cost. The cheapest phones won't have it. But in this case you really are getting something valuable for the additional money.

Jonathan Knudsen, senior security strategist at Synopsys, said: "CVE-2020-0022 can be exploited by anyone within range of your vulnerable phone who can determine your Bluetooth MAC address, which is not difficult. As a user, keeping current with updates and applying them in a timely manner is important. Unfortunately, many vulnerable, slightly older phones will not have continuing software update support from the manufacturer, which means users are faced with two unattractive options: either disable Bluetooth entirely, or get a newer phone."

The February patch roundup for Android included patches for 25 bugs with 19 of those vulnerabilities rated high. An additional 4 bugs were also rated high, but they were specifically tied to Qualcomm chipsets used inside Android devices.

The forecast appears cloudy for ClearView AI

Last week we talked about Clearview AI, the facial recognition company that's scraped the web for three billion faceprints and made them available to 600 police departments so they could identify people within seconds.

Since then ClearView has increased their collection of cease-and-desist letters from major US social media players. The first one was received from Twitter a few weeks ago when Twitter told Clearview to stop collecting its data and to delete whatever it's got. In addition, Facebook has similarly demanded that Clearview stop scraping photos because the action violates its policies. And now Google and YouTube are telling ClearView to stop violating their policies against data scraping.

What's Clearview's take on all this? Defiance. In an interview Wednesday morning on "CBS This Morning", Clearview AI founder and CEO Hoan Ton-That told listeners to trust him: "The technology is only to be used by law enforcement, and only to identify potential criminals." Ton-That claims that the results are 99.6% accurate and also that it's his right to collect public photos to feed his facial recognition app: "There is also a First Amendment right to public information. So the way we have built our system is to only take publicly available information and index it that way."

We know from last week that Illinois, with their BIPA (Biometric Information Privacy Act) feels that doing so is illegal. And YouTube's statement read: *"YouTube's Terms of Service explicitly forbid collecting data that can be used to identify a person. Clearview has publicly admitted to doing exactly that, and in response we sent them a cease and desist letter."*

As for Facebook, the company said last Tuesday that it has demanded that Clearview stop scraping photos because the action violates its policies. Clearview's response to Facebook's review of its practices may cause Facebook to take action. Facebook said: "We have serious concerns with Clearview's practices, which is why we've requested information as part of our ongoing review. How they respond will determine the next steps we take."

For their part, in addition to asserting a right under the US 1st amendment protection permitting access to publicly available data, Ton-That defended Clearview as being a Google-like search engine. He said: "Google can pull in information from all different websites. If it's public and it can be inside Google's search engine, it can be in ours as well."

Google disagreed saying that ClearView isn't at all like their search engine. They said: "There's a big difference between what we do and the way you're shanghaiing everybody's face images without their consent." Google wrote: "Most websites want to be included in Google Search, and we give webmasters control over what information from their site is included in our search results, including the option to opt-out entirely. Clearview secretly collected image data of individuals without their consent, and in violation of rules explicitly forbidding them from doing so."

The question to ask appears to be: When is public information not public?

Back in 2016, hiQ, a San Francisco startup, was marketing two products which depended upon whatever data LinkedIn's 500 million members had made public: "Keeper" identified employees who might be ripe for being recruited and "Skills Mapper" summarized a LinkedIn member's skills.

In that instance, hiQ was amassing public information, grabbing the same material that anyone could get from LinkedIn without having to log in. Any browser would display the same information hiQ was vacuuming up, organizing and reselling. When sufficiently analyzed inferences could be made to alert companies when their pivotal employees might be interviewing for another position or more.

So when is public information not public? Apparently that's when the social media firms that collect it insist that it's not public.

In the case of hiQ, LinkedIn sent a cease-and-desist letter alleging that it was violating serious anti-hacking and anti-copyright violation laws: the Computer Fraud and Abuse Act (CFAA), the Digital Millennium Copyright Act (DMCA), and California Penal Code § 502(c). LinkedIn (which had been exploring how to do the same thing with its own data that hiQ had achieved) also noted that it had blocked hiQ from accessing its data.

Then just this past September 2019, an appeals court told LinkedIn to back off and that it had no legal right to interfere with hiQ's profiting from its users' publicly available data. The court protected data scraping of public data in what at first looks like a major legal precedent... but it's actually a lot less clear. Our friends at the Electronic Frontier Foundation (EFF) wrote:

"While this decision represents an important step to putting limits on using the CFAA to intimidate researchers with the legalese of cease and desist letters, the Ninth Circuit sadly left

the door open to other claims, such as trespass to chattels or even copyright infringement, that might allow actors like LinkedIn to limit competition with its products."

And, unfortunately, even with this ruling the CFAA is broadly written and is subject to multiple conflicting interpretations across the federal circuits. This makes it likely that the Supreme Court will ultimately be forced to resolve the meaning of key terms within the CFAA such as "without authorization." There is nothing worse than broadly-written legislation. It can be the case that legislation is the result of a compromise and that the way it is was the only way it could be written at all. But that merely pushes the problem out to the courts to resolve.

So the question of the use and reuse of publicly available data may come down to biometrics.

The EFF's surveillance litigation director, Jennifer Lynch said that Clearview is the latest example of why we need laws that ban, or at least pause, law enforcement's secretive use of facial biometric recognition. She cited many cases of what she called law enforcement's – and Clearview's – abuse of facial recognition, stating: "Police abuse of facial recognition technology is not theoretical: it's happening today. Law enforcement has already used 'live' face recognition on public streets and at political protests."

As we've observed before, this is all being enabled by the incredible reductions in cost. The cost of processing power, the cost of massive data storage, and the cost and presence of ubiquitous networked communications. We didn't have this ten years ago. What are we going to have in another ten years?

The NIST is testing methods of recovering data from smashed smartphones

It makes sense when you think about it. There's been a lot of discussion through the years about how best irreversibly kill a hard drive. One of my favorites, since modern hard drive platters are often glass, is to smash the drive hard on either of its faces. If you then hear the sound of many tiny bits of glass rattling 'round inside, you can be quite certain that no one will be obtaining any data that was ever stored on those platters.

But what about with an entirely solid state smartphone? Bad guys have tried smashing their phones, soaking them in water and, shooting them with a gun, and subjecting them to many other forms of torture. So how effective are such measures?

It turns out that many criminals have discovered to their chagrin that reducing their devices to smashed plastic and glass means nothing if the device's little black epoxy memory chips remain in working order. Forensic engineers who work with police to gather evidence have become quite adept at performing these feats of posthumous data extraction. With more and more evidence now sitting on smartphones, a better understanding of what works and what doesn't has been growing into an issue of some urgency.

So the US National Institute of Standards and Technology (NIST) recently conducted a series of tests using 10 popular Android smartphones which had accumulated a mix of data during simulated use.

The NIST engineers and their forensic partners then attempted to extract the data from the

surviving memory chips using various methods to compare with the original data set. In some cases the chips could be left on the original motherboard and accessed via the "JTAG" serial interface. JTAG is an industry standard interface used for testing and programming electronics. In other cases the chips were physically removed from their original motherboards and interconnecting to directly -- sort of "in vitro" data extraction. NIST wrote:

"The comparison showed that both JTAG and chip-off extracted the data without altering it, but that some of the software tools were better at interpreting the data than others, especially for data from social media apps."

Which raises a good point… It's one thing to have access to the raw unencrypted data. But you still need to be able to make heads or tails of what you have.

Either way, it's a big ask. NIST stated that neither technique is easy, especially extracting data using JTAG, and that's before factoring the shortage of trained forensics people and the subtle differences between different data extraction software and the diversity of smartphones. NIST's forensic expert Rick Ayers said:

"Many labs have an overwhelming workload, and some of these tools are very expensive. To be able to look at a report and say, this tool will work better than that one for a particular case can be a big advantage."

What's interesting is that NIST was using and comparing two commercial systems that we have spoken of many times. The first of two available reports is titled: "Test Results for Binary Image (Joint Test Action Group (JTAG), Chip-Off) Decoding and Analysis Tool: Cellebrite Universal Forensic Extraction Device (UFED) Physical Analyzer v7.20.0.123"

https://www.dhs.gov/sites/default/files/publications/cellebriteuniversalforensicextractiondeviceufedphysicalanalyzerv7.20.0.123.pdf

And the second report is titled: "Test Results for Binary Image (JTAG, Chip-Off) Decoding and Analysis Tool: Paraben's Electronic Evidence Examiner – Device Seizure (E3:DS) v2.3.12037.16428"

https://www.dhs.gov/sites/default/files/publications/testresultforbinaryimagejtagchipoffdecodingandanalysistool.pdf

We have, of course, encountered "Cellebrite" many times in the past. They are a favorite of law enforcement for performing evidentiary forensic data extraction from smartphones.

Test Results for Binary Image (JTAG, Chip-Off) Decoding and Analysis ToolTool Tested:

Physical Analyzer Software Version: v7.20.0.123
Supplier: Cellebrite, Inc.
Address: 7 Campus Drive, Suite 210 Parsippany, NJ 07054
Telephone: (415) 361-4077
Website: http://www.cellebrite.com

Results Summary
Cellebrite's "Physical Analyzer" is a versatile mobile forensic solution that runs on existing hardware. It comes with a suite of applications, peripherals and accessories. "Physical Analyzer" was tested for its ability to decode and analyze binary images created by performing Chip-Off and JTAG data extractions from supported mobile devices.

Except for the following anomalies, the tool was able to decode and report all supported data objects completely and accurately for all mobile devices tested.

Stand-alone Files:
Stand-alone files (i.e., audio, documents, graphics, video) (Device: HTC One Mini_Chip-off)

Social Media Data:
Social media related data (i.e., Twitter) (Devices: LG K7_Chip-off, ZTE 970_Chip-off,)
Social media related data (i.e., Facebook) (Devices: HTC One XL_Chip-off, HTC Desire S_Chip-off, HTC One XL_JTAG, HTC Desire S_JTAG)

GPS-related Data: GPS related data (i.e., longitude, latitude coordinates, routes, addresses, etc.) (Device: HTC One Mini_Chip-off)

4.1 Chip-Off Data Extractions
The internal memory contents for Chip-Off binary images were decoded and analyzed with PA v7.20.0.123.

All test cases pertaining to the acquisition of supported Android devices were successful with the exception of the following.

- Stand-alone files (i.e., audio, documents, graphics, video) were not reported for the HTC One Mini.
- Twitter social media data was partially reported i.e., account related information for the LG K7 and ZTE 970.
- Facebook social media data was partially reported i.e., account related information for the HTC One XL and HTC Desire S.
- GPS related data (e.g., waypoints, longitude, latitude, routes) were not reported for the HTC One Mini.

Notes: Devices defined in the table below with an '*' e.g., HTC One XL*, both Chip-Off and JTAG data extractions were performed.
When performing the Chip-off data extraction, it appeared the HTC One Mini had suffered water damage, which may lead to differences in the data reported for the JTAG compared to Chip-off.

- Deleted Contacts, Calendar, Memo/Note entries were recovered for the HTC Desire 626, ZTE 970, Moto-E, Samsung S2, HTC One XL and Samsung S4.
- Deleted Contacts and Calendar entries were recovered for the LG K7 and HTC Desire S.
- Deleted Contac's and Memo entries were recovered for the HTC One Mini.
- Deleted Call logs were recovered for the LG K7, Moto-E, Samsung S2, Samsung S4 and HTC Desire S.
- Deleted SMS entries were recovered for the HTC Desire 626, LG K7, ZTE 970, Moto-E, Samsung S2, HTC One XL, Samsung S4, HTC One Mini and HTC Desire S.
- Deleted bookmark entries were recovered for the HTC Desire 626, ZTE 970, Moto-E, Samsung S2, HTC One XL, and HTC Desire S.

The NIST report did spend some time on what they termed the "Encryption barrier"

These techniques allow forensics teams to retrieve data but of course have no bearing on their ability to bypass any encryption that has been applied to it. Despite reports that specific tools can do this already, as with any data extraction it remains a skilled and time-consuming undertaking. That's why the US Government keep returning to the issue, in October 2019 even publicly asking Facebook to delay its end-to-end encryption rollout until it can be shown that this doesn't get in the way of investigators in a hurry.

Meanwhile, the long-running battle with Apple goes on despite the company cooperating by providing iCloud backups connected to the shooting in Pensacola in December. But NIST wrote: "Even supposing a bypass for encryption were at hand, the reality is that criminals still often damage their devices and delete their backups. If politicians underestimate the problems this poses, NIST doesn't. But it won't deliver quick answers. Smartphone forensics faces a long road ahead."

And, of course, we can project a bit farther into the future: With each generation our smart phones become more smart and less phone. The density of their storage and circuitry soars, their circuit boards acquire more layers while growing ever thinner, and overall the devices become more fragile and prone to inadvertent mishap, let alone attempts at their deliberate destruction at the hands of someone who is hoping to render their contents irretrievable.

From our "Yet Another Data Exfiltration Technique of the Week" department...
The paper is titled: "BRIGHTNESS: Leaking Sensitive Data from Air-Gapped Workstations via Screen Brightness"

https://arxiv.org/pdf/2002.01078.pdf

Abstract—Air-gapped computers are systems that are kept isolated from the Internet since they store or process sensitive information. In this paper, we introduce an optical covert channel in which an attacker can leak (or, exfiltrate) sensitive information from air-gapped computers through manipulations of the screen brightness. This covert channel is invisible and it works even while the user is working on the computer. Malware on a compromised computer can obtain sensitive data (e.g., files, images, encryption keys and passwords), and modulate it within the screen brightness, invisible to users. The small changes in the brightness are invisible to humans but can be recovered from video streams taken by cameras such as a local security camera, smartphone camera or a webcam. We present related work and discuss the technical and scientific background of this covert channel. We examined the channel's boundaries under various parameters, with different types of computer and TV screens, and at several distances. We also tested different types of camera receivers to demonstrate the covert channel. Lastly, we present relevant countermeasures to this type of attack.

These are the guys from the Department of Software and Information Systems Engineering at Ben-Gurion University and the Department of Electrical and Electronics Engineering, Shamoon College of Engineering, both located in Israel.

As we'll recall, this topic appears to be something of a hobby horse for them. In years previous we have covered their serious research into all manner of air-gapped computer data exfiltration. They have come up with ways to get data out through PC speakers, blinking LEDs, infrared lights in surveillance cameras, and even by modulating the rotation rate of a computer's cooling fans. That was their famous "Fansmitter" research. Yes... Where there's a will there's a way.

My first reaction was to wonder what computer containing data worthy of exfiltration would tolerate having its screen within the view of a camera of any kind. But in their 7-page paper they tackle the problem with their usual aplomb, just as they tackled the question of how many bits per second can we transmit by the sound of varying fan speed?

Nevertheless, undeterred, they conclude by writing:

> CONCLUSION
> In this paper we present an optical covert channel in which data is concealed on the LCD screen brightness, but is invisible to users. We presented a malware scheme that can exfiltrate sensitive data from isolated (air-gapped) computers. The attack model consists of (a) contaminating the target network, a task which has been demonstrated to be within the capabilities of a modern advanced persistent threat (APT), and (b) using a camera to take videos of the computer's display, a task that can be performed by a malicious insider or visitor, or by exploiting a surveillance camera. We exploit the limitations of bare human vision, concerning brightness perception, using sufficiently low values of contrast between the brightness levels. Consequently, the current results demonstrate the feasibility of our covert channel, while outlining its boundaries. Notably, this kind of covert channel is not monitored by existing data leakage prevention systems.

Promiscuous Cookies

This matters because Chrome 80 appeared last week with its implementation of the updated handling of the optional “SameSite” enforcement cookie property.

So let’s back up a bit, first: We noted early last summer that this change was potentially coming. Last May, in an IETF Draft, Google proposed a change of the default for non-specified behavior that will improve by tightening cookie handling, but which might also break some existing working systems. The IETF Draft document Abstract introduces the idea as follows:

Abstract
This document proposes two changes to cookies, inspired by the properties of the HTTP State Tokens mechanism proposed in [I-D.west-http-state-tokens]. First, cookies should be treated as "SameSite=Lax" by default. Second, cookies that explicitly assert "SameSite=None" in order to enable cross-site delivery should also be marked as "Secure".

So, first of all, let’s stop to look at the way things are now:

https://tools.ietf.org/id/draft-ietf-httpbis-rfc6265bis-03.html#rfc.section.5.3.7

5.3.7. The SameSite Attribute

If the attribute-name case-insensitively matches the string "SameSite", the user agent MUST process the cookie-av as follows:

1. Let enforcement be "None".
2. If cookie-av's attribute-value is a case-insensitive match for "Strict", set enforcement to "Strict".
3. If cookie-av's attribute-value is a case-insensitive match for "Lax", set enforcement to "Lax".
4. Append an attribute to the cookie-attribute-list with an attribute-name of "SameSite" and an attribute-value of enforcement.

Note: This algorithm maps the "None" value, as well as any unknown value, to the "None" behavior, which is helpful for backwards compatibility when introducing new variants.

5.3.7.1. "Strict" and "Lax" enforcement

Same-site cookies in "Strict" enforcement mode will not be sent along with top-level navigations which are triggered from a cross-site document context. As discussed in Section 8.8.2, this might or might not be compatible with existing session management systems. In the interests of providing a drop-in mechanism that mitigates the risk of CSRF attacks, developers may set the SameSite attribute in a "Lax" enforcement mode that carves out an exception which sends same-site cookies along with cross-site requests if and only if they are top-level navigations which use a "safe" (in the [RFC7231] sense) HTTP method.

Lax enforcement provides reasonable defense in depth against CSRF attacks that rely on unsafe HTTP methods (like POST), but does not offer a robust defense against CSRF as a general category of attack:

Attackers can still pop up new windows or trigger top-level navigations in order to create a "same-site" request (as described in section 2.1), which is only a speedbump along the road to exploitation.

1. Introduction

The HTTP State Tokens proposal ([I-D.west-http-state-tokens]) aims to replace cookies with a state management mechanism that has better security and privacy properties. That proposal is somewhat aspirational: it's going to take a long time to come to agreement on the exact contours of a cookie replacement, and an even longer time to actually do so.

While we're debating the details of a new state management primitive, it seems quite reasonable to reevaluate some aspects of the existing primitive: cookies. When we can find consensus on some aspect of HTTP State Tokens, we can apply those aspirations to cookies, driving incremental improvements to state management in the status quo.

Based on conversations at [HTTP-Workshop-2019] and elsewhere, I'd suggest that we have something like agreement on at least two principles:

1. HTTP requests should not carry state along with cross-site requests by default (see Section 8.2 of [RFC6265bis]).

2. HTTP requests should not carry state over non-secure channels (see Section 8.3 of [RFC6265bis], and [RFC7258]).

With those principles in mind, this document proposes two changes that seem possible to deploy in the near-term. User agents should:

1. Treat the lack of an explicit "SameSite" attribute as "SameSite=Lax". That is, the "Set-Cookie" value "key=value" will produce a cookie equivalent to "key=value; SameSite=Lax". Cookies that require cross-site delivery can explicitly opt-into such behavior by asserting "SameSite=None" when creating a cookie.

2. Require the "Secure" attribute to be set for any cookie which asserts "SameSite=None" (similar conceptually to the behavior for the "__Secure-" prefix). That is, the "Set-Cookie" value "key=value; SameSite=None; Secure" will be accepted, while "key=value; SameSite=None" will be rejected.

3.1. "Lax" by Default

The processing algorithm in Section 5.3.7 of [RFC6265bis] treats the absence of a "SameSite" attribute in a "Set-Cookie" header as equivalent to the presence of "SameSite=None". Cookies are therefore available for cross-site delivery by default, and developers may opt-into more security by setting some other value explicitly. Ideally, we'd invert that such that developers who accepted the risks of cross-site delivery (see Section 8.2 of [RFC6265bis]) could opt into them, while developers who didn't make any explicit choice would be protected by default.

We could accomplish this goal by first altering the processing algorithm, replacing the current step 1:

1. Let "enforcement" be "None".

with the following two steps:

1. Let "enforcement" be "Lax".
2. If cookie-av's attribute-value is a case-insensitive match for "None", set "enforcement" to "None".

And then by, altering step 13 of the cookie storage model (Section 5.4 of [RFC6265bis]) from:

13. If the cookie-attribute-list contains an attribute with an attribute-name of "SameSite", set the cookie's same-site-flag to attribute-value (i.e. either "Strict", "Lax", or "None"). Otherwise, set the cookie's same-site-flag to "None".
to:

13. If the cookie-attribute-list contains an attribute with an attribute-name of "SameSite", set the cookie's same-site-flag to attribute-value. Otherwise, set the cookie's same-site-flag to "Lax".

This would have the effect of mapping the default behavior in the absence of an explicit "SameSite" attribute, as well as the presence of any unknown "SameSite" value, to the "Lax" behavior, protecting developers by making cross-site delivery an explicit choice, as opposed to an implicit default.

3.2. Requiring "Secure" for "SameSite=None"

Cookies sent over plaintext HTTP are visible to anyone on the network. As section 8.3 of [RFC6265bis] points out, this visibility exposes substantial amounts of data to network attackers. We know, for example, that long-lived and stable cookies have enabled pervasive monitoring [RFC7258] in the past (see Google's PREF cookie [pref-cookie]), and we know that a secure transport layer provides significant confidentiality protections against this kind of attack.

We've seen that changing anything will tend to break something, and there are concerns about this, too. Troy Hunt of "Have I been Pwned" fame blogged about this coming change on January 3rd. His post -- from which I obtained the title of today's podcast -- is titled: "Promiscuous Cookies and Their Impending Death via the SameSite Policy"

https://www.troyhunt.com/promiscuous-cookies-and-their-impending-death-via-the-samesite-policy/

> Cookies like to get around. They have no scruples about where they go save for some basic constraints relating to the origin from which they were set. I mean have a think about it:
>
> If a website sets a cookie then you click a link to another page on that same site, will the cookie be automatically sent with the request? Yes.
>
> What if an attacker sends you a link to that same website in a malicious email and you click that link, will the cookie be sent? Also yes.
>
> Last one: what if an attacker directs you to a malicious website and upon visiting it your browser makes a post request to the original website that set the cookie - will that cookie still be sent with the request? Yes!
>
> Cookies just don't care about how the request was initiated nor from which origin, all they care about is that they're valid for the requested resource. "Origin" is a key word here too; those last two examples above are "cross-origin" requests in that they were initiated from origins other than the original website that set the cookie. Problem is, that opens up a rather nasty attack vector we know as Cross Site Request Forgery or CSRF. Way back in 2010 I was writing about this as part of the OWASP Top 10 for ASP.NET series and a near decade on, it's still a problem.

Troy then gives some examples of how a POST providing both the old and a new password which

carries a promiscuous cookie can be abused. It's in the link I provided if anyone wants to see the details. He shows how it's possible to use CSRF request verification tokens as a means to thwart this sort of attack, but they are a mess… and anyone who has recently been writing secure web applications knows. Troy explains:

> There are two anti-forgery tokens passed in the request, one in a cookie and one in the body, both called "__RequestVerificationToken". They're not identical but they're paired such that when the server receives the request it checks to see if both values exist and belong together. If not, the request is rejected. This works because whilst the one in the cookie will be automatically sent with the request regardless of its origin, in a forged request scenario the one in the body would need to be provided by the attacker and they have no idea what the value should be. The browser's security model ensures there's no way of the attacker causing the victim's browser to visit the target site, generate the token in the HTML then pull it out of the browser in a way the malicious actor can access. At least not without a cross site scripting vulnerability as well and then that's a whole different class of vulnerability with different defences.
>
> This, frankly, is a mess. Whilst it's relatively easy to implement via frameworks such as ASP.NET, it leaves you wondering - do cookies really need to be that promiscuous? Do they need to accompany every single request regardless of the origin? No, they don't.

Google is leading the pack with last week's update which has taken the first step toward hardening the default unspecified handling of cookies. Microsoft plans to follow, and Mozilla has indicated that it supports the idea, too.

Broken OpenID Apps

Microsoft early on warned that the SameSite changes could break sites and applications that rely on OpenID-based federation.

Erik Anderson wrote, in a July 23 Chromium Forum post on the use of cookies with SameSite: "We love the intent and spirit of this change, but we fairly quickly determined that this breaks a large number of our sites leveraging Azure Active Directory (AAD) and Microsoft Account (MSA) authentication using the contract as defined here. We suspect that other OpenID-based federated auth providers may have similar scenarios and be broken."

Google warned IT pros in its Oct. 23 Chromium blog post that there could be problems with internal applications and single sign-on implementations:

Enterprise IT administrators may need to implement special policies to temporarily revert Chrome Browser to legacy behavior if some services such as single sign-on or internal applications are not ready for the February launch.

IT Pro Warnings

Microsoft has issued a specific warning about the coming SameSite changes. Effects could be felt when using Microsoft Teams client applications. There are considerations for sites that use ASP.NET. Exchange Server, SharePoint Server and the Skype for Business client all will need to have the latest updates installed.

Microsoft warned in a Microsoft Teams and SameSite cookies document that "Applications running in the Teams desktop client are incompatible with the SameSite=None attribute, and they will not work as expected." The document offered a couple of "workaround" options. It also explained that the Secure attribute needs to be used when the SameSite attribute's value is set to None to assure that third-party cookies won't get rejected.

For sites using ASP.NET or ASP.NET Core, Microsoft warned in an Oct. 18 ASP.NET blog post that the new SameSite changes will be in effect with ".NET 4.7.2 and in .NET Core 2.1 and above" and they could break OpenIDConnect log-ins. Updates to .NET that were released back in December and November added support for the new SameSite behavior.

So, we see another example of “Security is hard and change is even harder.” But tightening up the default behavior of cross-site cookies will clearly be a good thing moving forward. And if we’ve learned anything is that UNTIL there is some actual breakage no one will change anything. So Google is biting the bullet and is being willing to break a few eggs in order to force some changes that will, in the end, significantly improve the security of web applications for everyone.