Security Now! #631 - 10-03-17

Private Contact Discovery

This week on Security Now!

This week we discuss some aspects of iOS v11, the emergence of browser hijack crypto currency mining, new information about the Equifax hack, Google security research and Gmail improvements, breaking DKIM without breaking it, concerns over many servers in small routers and aging unpatched motherboard EFI firmware, a new privacy leakage bug in IE, a bit of miscellany, some long-awaited closing the loop feedback fro our listeners, and a close look into a beautiful piece of work my Moxie & Co on Signal.

Security Failure: EpiPen's Database Of Everyone Who's Allergic To Bees Has Been Obtained By Bees

Security News

A brief rant about the evolution of iOS

- The buggiest iOS every released.
- EVERY device has done utterly bizarro things, repeatedly.
- Older device are REALLY SLOWING DOWN over time with iOS advancement.

Apple's FaceID Security
https://images.apple.com/business/docs/FaceID_Security_Guide.pdf

Employs a "TrueDepth" camera system to map 3D facial geometry.

Confirms "attention" with pupil-tracking to detect the direction of gaze.

The iPhone X includes dedicated neural network hardware that Apple calls a "Neural Engine", which can perform up to 0.6 teraops / second used for Face ID.

Face ID adapts to changes in your appearance (Steve shaves and regrows his moustache.)

The ENTIRE Face ID system is a LOCAL authenticator. Nothing ever leave the user's device. Unlike Amazon's Echo system where the user's speech IS sent to the cloud for speech recognition, ALL facial recognition is performed on the device and NO FACIAL information ever leaves the device.

Apple says this:

> To use Face ID, you must set up iPhone X so that a passcode is required to unlock it. When Face ID detects and matches your face, iPhone X unlocks without asking for the device passcode. Face ID makes using a longer, more complex passcode far more practical because you don't need to enter it as frequently. Face ID doesn't replace your passcode, but provides easy access to iPhone X within thoughtful boundaries and time constraints. This is important because a strong passcode forms the foundation of your iOS device's cryptographic protection.

You can always use your passcode instead of Face ID, and it's still required under the following circumstances:

- The device has just been turned on or restarted.
- The device hasn't been unlocked for more than 48 hours.
- The passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and Face ID has not unlocked the device in the last 4 hours.
- The device has received a remote lock command.
- After five unsuccessful attempts to match a face.
- After initiating power off/Emergency SOS by pressing and holding either volume button and the side button simultaneously for 2 seconds.

The probability that a random person in the population could look at your iPhone X and unlock it using Face ID is approximately 1 in 1,000,000 (versus 1 in 50,000 for Touch ID).

For additional protection, Face ID allows only five unsuccessful match attempts before a passcode is required to obtain access to your iPhone. The probability of a false match is different for twins and siblings that look like you as well as among children under the age of 13, because their distinct facial features may not have fully developed. If you're concerned about this, we recommend using a passcode to authenticate.

Tech Details:

- Once it confirms the presence of an attentive face, the TrueDepth camera projects and reads over 30,000 infrared dots to form a depth map of the face, along with a 2D infrared image.

- This data is used to create a sequence of 2D images and depth maps, which are digitally signed and sent to the Secure Enclave.

- To counter both digital and physical spoofs, the TrueDepth camera randomizes the sequence of 2D images and depth map captures, and projects a device-specific random pattern. (Note that this has the effect of locking the captured data to the specific phone, preventing data captured from one phone from being used to spoof another phone, since each phone's scanning sequence is per-device.)

- Facial images captured during normal unlock operations aren’t saved, but are instead immediately discarded once the mathematical representation is calculated for comparison to the enrolled Face ID data.

- More details about locking/unlocking key management: And Apple did everything right.

- Application developers have access to Face ID recognition APIs.

The Emergence of Browser Hijack CryptoCurrency Mining
https://www.techdirt.com/articles/20170928/11262538304/showtime-wont-explain-why-website-was-hijacking-user-browsers-to-covertly-mine-cryptocurrency.shtml

Ads are mining, compromised sites are mining.

What's being done is that the user's processor power is being hijacked to roll the dice with a vanishingly small chance of success. But if a sufficient number of rolls occur, any machine might get lucky. The annoyance is that the user whose machine just got lucks doesn't get any reward for the use of their processor's power... the hijackers win the prize.

Bitcoin: Solving the SHA256 hash problem.

Once upon a time it was feasible for a single user to mine crypto currency.

The trouble is that, in California at least, even with the most efficient mining equipment, the cost of the electrical power required to perform the mathematical calculations has grown higher than the dollar value returned.

The consequence on mobile might be greater power consumption.

It's an annoyance, but it's not a security/privacy breach per se.

Judge: FBI Doesn't Have to Reveal How It Unlocked iPhone Used by San Bernardino Terrorist.
https://thehackernews.com/2017/10/apple-fbi-iphone-unlock.html

Terrorist Syed Farook / San Bernardino 2015 mass shooting killed 14 people.

Apple refused to help feds access data on the locked iPhone.

The FBI reportedly paid over a million dollars to an unnamed vendor to unlock the shooter's iPhone.

Remember that last year, James Comey the former FBI Director, indirectly disclosed that the agency reportedly paid around $1.3 Million for the hacking tool that helped the agency break into Farook's iPhone 5C.

The Associated Press, USA Today, and Vice Media all sued the FBI last year under the Freedom of Information Act (FOIA) to obtain information about the name of the company who helped to unlock the phone and how much it was paid.

Since taxpayer dollars were used, the argument was that we, the public, had a right to know how those dollars were being spent.

In a ruling, Saturday, U.S. District Judge Tanya S. Chutkan of the District of Columbia disagreed.

She ruled that the information about the vendor and the hacking tool used were exempt from mandatory disclosure under the government transparency law.

"It is logical and plausible that the vendor may be less capable than the FBI of protecting its proprietary information in the face of a cyber attack."

"The FBI's conclusion that releasing the name of the vendor to the general public could put the vendor's systems, and thereby crucial information about the technology, at risk of incursion is reasonable."

Regarding the cost of the hacking tool, Tanya agreed with the US government that revealing the price the government paid for unlocking iPhone could harm national security:

"Releasing the purchase price would designate a finite value for the technology and help adversaries determine whether the FBI can broadly utilize the technology to access their encrypted devices."

Translation: The FBI asked to keep this information private an it got what it asked for.

The Equifax Hack Has the Hallmarks of State-Sponsored Pros
https://www.bloomberg.com/news/features/2017-09-29/the-equifax-hack-has-all-the-hallmarks-of-state-sponsored-pros

Investigations into the massive breach aren't complete, but the intruders used techniques that have been linked to nation-state hackers in the past.

Bloomberg provided a FABULOUS wrap-up of what's currently known. Since I know that our listeners' insterest in this ranges from "holy crap tell me more" to "move along, these are not the droids you're looking for", I will only tease Bloomberg's in-depth coverage from the link in the show notes:

> Nike Zheng, a Chinese cybersecurity researcher from a bustling industrial center near Shanghai, probably knew little about Equifax or the value of the data pulsing through its servers when he exposed a flaw in popular backend software for web applications called Apache Struts. Information he provided to Apache, which published it along with a fix on March 6, showed how the flaw could be used to steal data from any company using the software.
>
> The average American had no reason to notice Apache's post but it caught the attention of the global hacking community. Within 24 hours, the information was posted to FreeBuf.com, a Chinese security website, and showed up the same day in Metasploit, a popular free hacking tool. On March 10, hackers scanning the internet for computer systems vulnerable to the attack got a hit on an Equifax server in Atlanta, according to people familiar with the investigation.
>
> Before long, hackers had penetrated Equifax. They may not have immediately grasped the value of their discovery, but, as the attack escalated over the following months, that first group—known as an entry crew—handed off to a more sophisticated team of hackers. They homed in on a bounty of staggering scale: the financial data—Social Security numbers, birth dates, addresses and more—of at least 143 million Americans. By the time they were done, the attackers had accessed dozens of sensitive databases and created more than 30 separate entry points into Equifax's computer systems. The hackers were finally discovered on July 29, but were so deeply embedded that the company was forced to take a consumer complaint portal offline for 11 days while the security team found and closed the backdoors the intruders had set up.
>
> The handoff to more sophisticated hackers is among the evidence that led some investigators inside Equifax to suspect a nation-state was behind the hack. Many of the tools used were Chinese, and these people say the Equifax breach has the hallmarks of similar intrusions in recent years at giant health insurer Anthem Inc. and the U.S. Office of Personnel Management; both were ultimately attributed to hackers working for Chinese intelligence.

Google Researcher Publishes PoC Exploit for Apple iPhone Wi-Fi Chip Hack
https://thehackernews.com/2017/09/apple-iphone-wifi-hacking.html

Another problem has been identified and reported in the Broadcom WiFi chip firmware used by many devices. A PoC (proof of concept) has been developed which affects all iOS devices up to v10.3.3.

It's not necessary to upgrade to iOS v11.0.1, but that will work. So long as you have v10.3.3 you're okay.

Google Will Retool User Security in Wake of Political Hack
https://www.bloomberg.com/news/articles/2017-09-29/google-is-said-to-retool-user-security-in-wake-of-political-hack

The implications of Russian involvement in US Presidential election meddling have been generating news. Facebook and Twitter have been examining the extent of their unwitting involvement. And now Google is getting ready to bump up the security of their gMail offering:

Details are still sketchy, and the Google personnel requested anonymity since the plans are not yet officially public.

But the upshot appears to be that Google' gMail service will be adding a "supper secure" service ostensibly named the "Advanced Protection Program" to be marketed to corporate executives, politicians and others with heightened security concerns.

The APP will require the use of TWO separate physical security keys of some kind and people familiar with the program stated that: "The new service will block all third-party programs from accessing a user's emails or files stored on Google Drive. The program will be updated with new features to protect user data on an on-going basis."

Breaking DKIM - on Purpose and by Chance
http://noxxi.de/research/breaking-dkim-on-purpose-and-by-chance.html

Except it didn't "break" it. It should have been called: Spoofing DKIM.

Remember that DKIM is a system whereby a mail domain -- such as GRC.COM -- is able to public its PUBLIC key. And then use its matching secret private key to sign outgoing eMail. Thus a recipient server is able to lookup the domain's public key and use it to validate that the incoming email was actually received from the indicated source.

DKIM protects the "mail envelope" and it's possible for a clever hacker to DISPLAY something outside of the envelope so that what DKIM is protecting is not what the user sees. This allows a DKIM-aware client to say "Yes! This is authentic"... while what the user is seeing is not.

This is not good.

DNSMASQ Vulnerabilities
Yesterday: "Behind the Masq: Yet more DNS, and DHCP, vulnerabilities" / October 2, 2017
https://security.googleblog.com/2017/10/behind-masq-yet-more-dns-and-dhcp.html

Dnsmasq provides network infrastructure for small networks: DNS, DHCP, router advertisement and network boot.

It's a set of lightweight, small footprint servers suitable for resource constrained routers and firewalls.

It's been widely used for tethering on smartphones and portable hotspots, and to support virtual networking in virtualisation frameworks. Supported platforms include Linux, Android, *BSD, and Mac OS X. Dnsmasq is included in most Linux distributions and the ports systems of FreeBSD, OpenBSD and NetBSD. Dnsmasq provides full IPv6 support.

The attacks are all local exploits, have been responsibly disclosed and patched. Since it's servers -- such as small routers, etc. -- and not clients, such as IoT endpoint devices, things like home routers and access points, Apple and Google nodes, etc.

So... keep an eye out for firmware updates to our routers. The danger is not remote, but it'll be better to get it tightened up.

Critical Code in Millions of Macs Isn't Getting Apple's Updates
https://www.wired.com/story/critical-efi-code-in-millions-of-macs-is-not-getting-apple-updates/
https://arstechnica.com/information-technology/2017/09/an-alarming-number-of-macs-remain-vulnerable-to-stealthy-firmware-hacks/

EFI firmware vulnerabilities are very difficult to exploit, but very powerful if exploited, and difficult for average users to deal with. It's not like running an A/V tool to scan for malware. EFI firmware vulnerabilities open the door for virtually undetectable super-rootkit-style PRE-BOOT attacks.

But, really, typical users have little to fear.

A small client application that uses the Duo Labs EFIgy API to inform you about the state of your Macs EFI firmware

http://efigy.io

ALL motherboards have these real and potential vulnerabilities, and an advantage to Apple Macs is that there's one point of responsibility. But it's also a more homogeneous monoculture compared with Windows and Linux machines.

Overall, the attention being given to this is, as always, a good thing. :)

Internet Explorer bug leaks whatever you type in the address bar
https://arstechnica.com/information-technology/2017/09/bug-in-fully-patched-internet-explorer-leaks-text-in-address-bar/
A bug in IE allows JavaScript to intercept the "Enter" navigation AWAY from a page and to obtain whatever the user has entered into the URL / Search field. Thus... any code running on a site where you ARE can determine where you went, and why.

Miscellany

Katie (@MercyGrace)
What'd you think? Did you enjoy it? Was it Star Trek or a CBS knockoff?

(And "The Orville" -- painfully dumb.)

SpinRite

Steven Perry, CISSP
Location: Charlotte, NC
Subject: SpinRite saves a Drobo
Date: 28 Sep 2017 08:20:32
:
Hi Steve,

Have a SpinRite testimonial for you. You could title this one SpinRite saves a Drobo. I have a Drobo FS NAS device that is about 5 years old now. When I setup the Drobo, I made sure that each of the 5 2TB WD Black drives passed SpinRite before they were installed in to the Drobo. This past Sunday the first drive bit the dust. I swapped out my spare 2TB drive (that had also passed SpinRite) and during the rebuild process one of the other drives went offline and Drobo was setting off alerts that too many drives had been removed. I was able to safely shut down the Drobo, remove the drive it thought was missing, run it through a level 2 pass from SpinRite which took about 11 hours to complete. The drive was then returned to the Drobo, powered on to find that Drobo was now seeing the drive and was able to successfully rebuild the new drive and all is good again. I did try to SpinRite the first drive that failed to find it was really dead and that it would not even spin up for SpinRite to detect the drive. Once again, SpinRite saves the day! I am eagerly awaiting the updates so I can run it on my Macs!

Steven Perry, CISSP

Closing The Loop

Mark Havas (@techuntangler)
@SGgrc Regarding your printing 2FA codes to have them on paper... Would you keep the images in a password manager like 1password? TA!

(We keep circling around this. Everyone wants a shortcut. Everyone wants more ease of use. But OFFLINE means OFFLINE. Not online. If it's offline it CANNOT be hacked. If it's online, it CAN be!)

Jamie (@LinkLayer)
@SGgrc My CS Prof anticipated fuzzing in 1976. He said to test input handling with punch cards from the trash and swept up from the floor.

Michael Johnston (@michaelsmusic)
@SGgrc How a segment on taking a laptop on a flight out of the USA, working, and returning. The best way to do this with no border hassles?

1. Clone your drive before you leave.
2. Optionally add Veracrypt to the travel instance.
3. Use strongly encrypted drive during your trip.
4. Before returning home, move any new documents to the cloud.
5. Remove encrypted drive (which may not contain malware).
6. Restore original drive that never left home.
7. Recover cloud-stored work.
8. Once sure all's well, wipe travel drive.

If drive exchanges are not feasible:
1. Use "Image For Windows" to make offline image.
2. Optionally encrypt drive for travel.
3. Restore from image upon return.

Tom Terrific Krauska (@tomterrific1947)
@SGgrc Portable Dog Killer in Cuba? Did Russia steal your plans or use something similar on our diplomats in Cuba?

James Parsons (@jparsons_net)
@SGgrc If #SQRL requires passwd re-entry every session, it encourages weak passwds—especially on mobile. Special chars r difficult on phones

NiGHTS62 (@nights62)
@SGgrc You referred to bugs found in Chrome via DOM Fuzzing in as Zero-Day bugs. Doesn't Zero-Day mean it was found being exploited?
@SGgrc also said with this DOM fuzzing technology you too can find Zero-days in browsers for about $1000. I think you mean bugs not zero-day

dpmanthei (@dpmanthei)
@SGgrc Payment API: Does this mean all adopting online retailers can STOP storing our credit cards? Breaches would be far less harmful!

Emile Mercier (@Emile_Mercier)
@SGgrc as I understand it , an acronym is pronounceable, like PIE. TNO is an initialism, not an acronym afaik

Shankar Kulumani (@skulumani)
@SGgrc I've got an Xfinity router/modem combo. How do I keep my devices secure? Another router or something else?

Private contact discovery for Signal

Last Thursday, Moxie posted "Technology preview: Private contact discovery for Signal"

Matthew Green (@matthew_d_green)
Private contact discovery for Signal. Make no mistake: what Moxie is doing here is going to revolutionize messaging: https://signal.org/blog/private-contact-discovery/

Encrypted data vs metadata leakage -- eavesdroppers might not know WHAT was said, but knowing WHO it was said to is still VITALLY useful information.

This is now referred to as a user's "Social Graph" and it's a source of significant privacy leakage.

How does a new Signal user with an existing address book determine which of his/her contacts are also Signal users WITHOUT making any disclosure to a third party eavesdropper?

An example that does not work:

The central Signal server could have a single master list containing the hash of every subscriber's phone number. So just a list of SHA256 gibberish.

A new Signal user could hash every phone number in their contacts list, scramble their order, and send them to the central server. The server would only return those that match a hash in its master hash list. The client then rehashes each phone number in its contact list and checks the server's returned "hash list" to identify the contacts it has on common... and marks those as Signal users.

The problem, of course, is that while a hash is a one-way function, there is insufficient input entropy from a phone number to prevent is from being brute-forced.

If the central server cannot be trusted NOT TO REVERSE the contact hashes, there is no assurance of privacy.

Moxie's solution: Arrange to enforce server-side trust.

SGX: Software Guard Extensions.
Intel is developing the Intel® Software Guard Extensions (Intel® SGX) technology, an extension to Intel® Architecture for generating protected software containers. The container is referred to as an enclave. Inside the enclave, software's code, data, and stack are protected by hardware enforced access control policies that prevent attacks against the enclave's content. In an era where software and services are deployed over the Internet, it is critical to be able to securely provision enclaves remotely, over the wire or air, to know with confidence that the secrets are protected and to be able to save secrets in non-volatile memory for future use. This paper describes the technology components that allow provisioning of secrets to an enclave. These components include a method to generate a hardware based attestation of the software running inside an enclave and a means for enclave software to seal secrets and export them outside of the enclave (for example store them in non-volatile memory) such that only the same enclave software would be able un-seal them back to their original form.

SGX is similar to the ARM TrustZone.

A "Secure Enclave" whose code content can be protected from modification and signed for verification.

Available on Skylake and later processors.

It was originally designed for DRM to prevent alteration of client-side code.

But it CAN be used server side to provide strong privacy guarantees.

Moxie: However, we can invert the traditional SGX relationship to run a secure enclave on the server. An SGX enclave on the server-side would enable a service to perform computations on encrypted client data without learning the content of the data or the result of the computation.

SGX contact discovery -- Private contact discovery using SGX is fairly simple at a high level:
Run a contact discovery service in a secure SGX enclave.
Clients that wish to perform contact discovery negotiate a secure connection over the network all the way through the remote OS to the enclave.
Clients perform remote attestation to ensure that the code which is running in the enclave is the same as the expected published open source code.
Clients transmit the encrypted identifiers from their address book to the enclave.
The enclave looks up a client's contacts in the set of all registered users and encrypts the results back to the client.

Since the enclave attests to the software that's running remotely, and since the remote server and OS have no visibility into the enclave, the service learns nothing about the contents of the client request. It's almost as if the client is executing the query locally on the client device.

BUT...

Although the RAM inside the SGX enclave is encrypted, the access patterns within the memory cannot be obscured.

Oblivious RAM (ORAM).

> An Oblivious RAM (ORAM) simulator is a compiler that transforms algorithms in such a way that the resulting algorithms preserve the input-output behavior of the original algorithm but the distribution of memory access pattern of the transformed algorithm is independent of the memory access pattern of the original algorithm. The definition of ORAMs is motivated by the fact that an adversary can obtain nontrivial information about the execution of a program and the nature of the data that it is dealing with, just by observing the pattern in which various locations of memory are accessed during its execution. An adversary can get this information even if the data values are all encrypted.

Moxie & Co. spent a great deal of time working out how to perform hash table lookup operations which were efficient and also whose access patterns would not create an side-channel leakage.

They wind up developing a "Cache Line Aware" solution which allows them to create a hash lookup table under full memory-access scrutiny while providing an external observer with NO information about the layout of the resulting table entries. While the table creation is time-consuming, it only needs to be done once. Then this lookup table can be used with high efficiency without revealing any useful information to an observer.

By pushing the inefficiencies of "Oblivious Access" into the setup and teardown sections of batch processing, the critical section remains fast enough for the entire process to scale to over a billion users.

Moxie: We've put all of this together into a full contact discovery service, scalable to billions of users, that allows Signal users to discover their social graph without revealing their contacts to the Signal service. Like everything we build, the contact discovery service is open source.

https://github.com/whispersystems/contactdiscoveryservice