Security Now! #608 - 04-18-17

News & Feedback Potpourri

This week on Security Now!

This week Steve and Leo discuss another new side-channel attack on smartphone PIN entry (and much more), Smartphone fingerprint readers turn out to be far more spoofable that we had hoped. All Linux kernels prior to v4.5 are vulnerable to a serious remote network attack over UDP, a way to prevent Google from tracking the search links we click (and to allow us to copy the links from the search results), the latest NSA Vault7 data dump nightmare, the problem with punycode domains, four years after the public UPnP router exposure, looking closely at the mixed blessing of hiding WiFi access point SSID broadcasts, some miscellany, and then a collection of quick "Closing The Loop" follow-ups from last week's "Proactive Privacy" podcast.

Gee… I wonder what digits this PIN uses...

Security News

"Stealing PINs via Mobile Sensors: Actual Risk versus User Perception"
Four researchers at the School of Computing Science, Newcastle University, UKK

ABSTRACT:
In the first part of this paper, we propose PINlogger.js which is a JavaScript-based side channel attack revealing user PINs on an Android mobile phone. In this attack, once the user visits a website controlled by an attacker, the JavaScript code embedded in the web page starts listening to the motion and orientation sensor streams without needing any permission from the user. By analysing these streams, it infers the user's PIN using an artificial neural network. Based on a test set of fifty 4-digit PINs, PINlogger.js is able to correctly identify PINs in the first attempt with a success rate of 82.96%, which increases to 96.23% and 99.48% in the second and third attempts respectively. The high success rates of stealing user PINs on mobile devices via JavaScript indicate a serious threat to user security.

In the second part of the paper, we study users' perception of the risks associated with mobile phone sensors. We design user studies to measure the general familiarity with different sensors and their functionality, and to investigate how concerned users are about their PIN being stolen by an app that has access to each sensor. Our results show that there is significant disparity between the actual and perceived levels of threat with regard to the compromise of the user PIN. We discuss how this observation, along with other factors, renders many academic and industry solutions ineffective in preventing such side channel attacks.

All of the most popular web browsers -- Chrome, Firefox, Safari, Opera & Dolphin -- support these functions and enable this PIN harvesting. Therefore any web page we visit can perpetrate this attack.

Geolocation data has previously been identified as a privacy concern, so modern web browser ask permission from their users before returning this information to a web server. (You may have noticed your browser saying "This website is requesting your location, click "ok" to allow or "cancel" to decline.) -- That REALLY annoys me. What I want is a third option labelled "FU".

At this time, JavaScript code running in a webpage does not require any user permission to access sensor data, such as device motion and orientation. And there is no notification while JavaScript is reading the sensor data stream. This allows browser-based attacks to be carried out covertly.

Following the W3C specifications, motion and orientation sensor data are available from a series of measured real time parameters:

- Device orientation, which provides the physical orientation of the device, expressed as three rotation angles in the device's local coordinate frame.
- Device's linear acceleration, which provides the physical acceleration of the device, expressed in x/y/z Cartesian coordinates in the device's local coordinate frame.
- Device acceleration-including-gravity, which is similar to acceleration except that it includes gravity as well.

- Device rotation rate which provides the rotation rate of the device about the local coordinate frame, expressed as three rotation angles.

How available is this data?

In some cases, such as Chrome and Dolphin on iOS, an inactive tab including the sensor listeners have access to the sensor measurements. And Safari allows inactive tabs to access the sensor data even when the minimized or the screen is locked.

Their user awareness study revealed that user are generally aware of the smartphone sensors they actively interact with. From most aware toward least aware, they are:

- 100% Camera & Touch Screen
- 97% Microphone, Bluetooth, GPS
- 93% WiFi
- 83% Fingerprint

But very little and decreasing awareness of:

- Rotation, Orientation, Ambient Light, Motion, TouchID, Device Temperature, NFC, Barometer, Ambient Temperature, Proximity, Gravity, Accelerometer, Gyroscope, Magnetic field, Ambient Humidity, Ambient Pressure, Hall Effect Sensor.

Modern smartphones are bristling with sensors... and thanks to the W3C, which keeps standardizing access to them through web pages, almost all of them can be covertly accessed by the JavaScript loaded into our browsers by untrusted web sites.

https://arxiv.org/pdf/1605.05549v1.pdf

Smartphone fingerprint readers really not that secure

Researchers from New York University and Michigan State University

Abstract:
This paper investigates the security of partial fingerprint-based authentication systems, especially when multiple fingerprints of a user are enrolled. A number of consumer electronic devices, such as smartphones, are beginning to incorporate fingerprint sensors for user authentication. The sensors embedded in these devices are generally small and the resulting images are, therefore, limited in size. To compensate for the limited size, these devices often acquire multiple partial impressions of a single finger during enrollment to ensure that at least one of them will successfully match with the image obtained from the user during authentication. Further, in some cases, the user is allowed to enroll multiple fingers, and the impressions pertaining to multiple partial fingers are associated with the same identity (i.e., one user). A user is said to be successfully authenticated if the partial fingerprint obtained during authentication matches any one of the stored templates. This paper investigates the possibility of generating a "MasterPrint", a synthetic or real partial fingerprint that serendipitously matches one or more of the stored templates for a significant number of users. Our preliminary results on an optical fingerprint dataset and a capacitive fingerprint dataset indicate that it is indeed possible to locate or generate partial fingerprints that can be used to impersonate a large number of users. In this regard, we expose a potential vulnerability of partial fingerprint-based authentication systems, especially when multiple impressions are enrolled per finger.

In their simulations, the researchers were able to develop a set of artificial "MasterPrints" that could match real prints similar to those used by phones as much as 65% of the time.

Discussion:

We all know, and Sherlock Holmes demonstrated, that a person's ENTIRE fingerprint is globally unique. But... think about this: How unique is a much smaller portion of an entire fingerprint? How unique CAN a smaller piece be?

Think of this as being like having a ultra-secure 20-digit PIN, but only needing to provide any three successive digits within that PIN.

What we're seeing is another instance where manufacturers desire to minimize their user's inconvenience by choosing to minimize their fingerprint recognition's false negatives. But the demonstrated result of this is a natural increase in false positive probability.

It occurs to me that I have never "attacked" my own iPhones' fingerprint reader system -- and that it virtually never refuses to unlock for me. I have I have never asked anyone I'm meeting with to TRY to use their fingers to unlock my phone. But after seeing this "obvious in retrospect" research, I'm going to start asking everyone I meet to see whether their fingerprint will unlock my phone. I'll report on what I learn, and if anyone listening to this is similarly curious, I'll be interested in learning of everyone else's findings.

The Linux kernel prior to v4.5 has a potential problem with receiving UDP

udp.c in the Linux kernel before 4.5 allows remote attackers to execute arbitrary code via UDP traffic that triggers an unsafe second checksum calculation during execution of a recv system call with the MSG_PEEK flag.

- https://nvd.nist.gov/vuln/detail/CVE-2016-10229
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10229
- Attack Vector (AV): Network
- Attack Complexity (AC): Low
- Privileges Required (PR): None
- User Interaction (UI): None

Google:

- Pixel
- Pixel XL
- Pixel C
- Nexus Player
- Nexus 5X
- Nexus 6
- Nexus 6P
- Nexus 9
- Android One
- Android

Android Security Bulletin—April 2017

- https://source.android.com/security/bulletin/2017-04-01
- CRITICAL REMOTE CODE EXECUTION

- "Remote code execution vulnerability in kernel networking subsystem"

- A remote code execution vulnerability in the kernel networking subsystem could enable a remote attacker to execute arbitrary code within the context of the kernel. This issue is rated as Critical due to the possibility of remote code execution in the context of the kernel.

Discussion:

There is virtually NO useful public information regarding the practical exploitability of this vulnerability. For this to cause damage, code running in user-space would need to be receiving UDP traffic but, for some reason, choose to "peek" at the next datagram rather than to obtain and consume it... which is the normal use case.

That said, a string search through the LINUX source tree does find hundreds of references to the MSG_PEEK flag. And given the massive installed base of Linux embedded in non-Google Android smartphones, our routers, televisions, DVRs and higher-end IoT devices -- which are much less easily updated than Google's devices, our desktops and servers, there can be ZERO DOUBT that would-be attackers are carefully examining every instance of Linux's user-space code for uses of MSG_PEEK that *can* be exploited.

John Gruber / Daring Fireball:
DirectLinks: Safari Extension That Circumvents Google and Facebook Link Redirects
Great little Safari extension from Canisbos:
This extension circumvents certain techniques used by Google and Facebook to track link clicks.

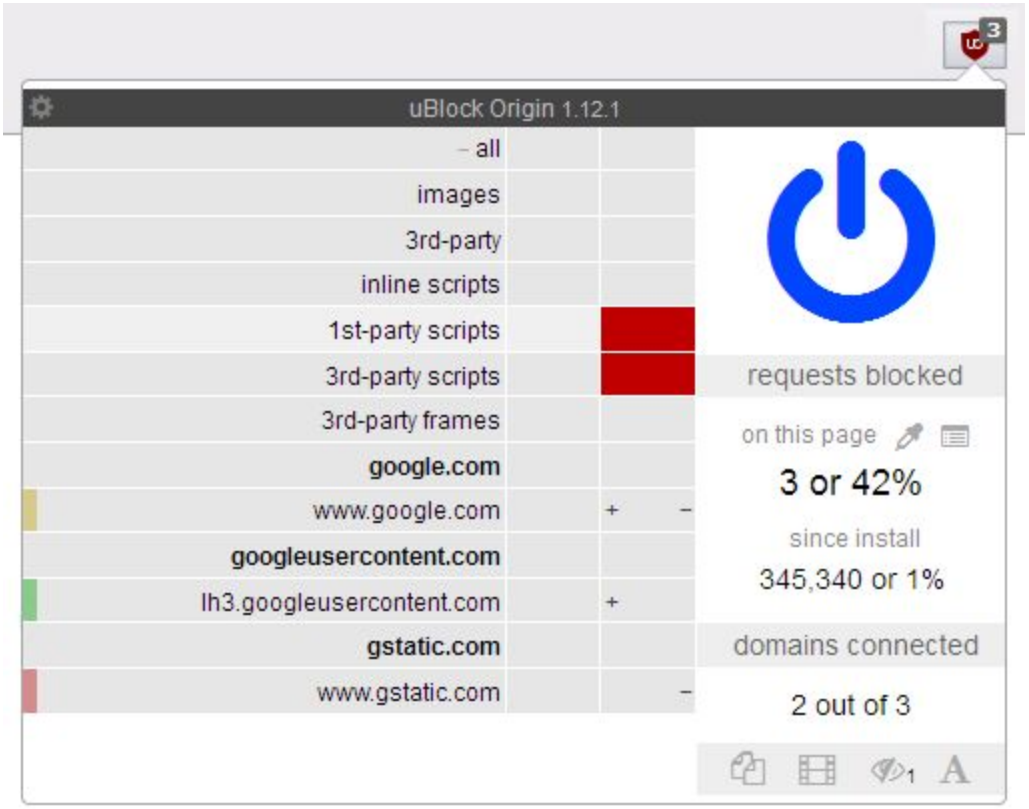
When you click a link in Google search results, Google uses JavaScript to replace the actual link with an indirect one, which they use for click tracking. Google then redirects the browser to the actual destination after logging the click. DirectLinks disables the JavaScript that replaces real links with indirect ones, so that when you click a search result link, Safari goes straight to the destination.

If you've ever tried dragging-and-dropping a URL from Google search results and getting a Google redirection URL instead of the actual URL you wanted (and Google's JavaScript will show the actual URL in the status field if you hover over the link, so it's impossible to tell that's what's going to happen), this extension is for you. There are obvious privacy benefits as well.

Discussion

After learning that Google's search results pages were being dynamically altered by JS -- rather than being delivered to our browser with pre-built, I simply used uBlock Origin to selectively block scripts on "www.google.com"... And it worked!

Gorhill's UI is so opaque that I needed to refresh my memory of how it worked since it has largely been a "set and forget" tool.



Security Now! #523 (September 1st, 2015) was "uBlock Origin".

Using the expanded UI, the lefthand column is global settings and the righthand column is per-site settings. So I just clicked the "red" in the righthand column and alll my Google search links are now tracking-free and also freely copyable.

NSA / Vault7 / ETERNALBLUE

<Typical Breathless Press Coverage> Last Friday, April 14th:
The ShadowBrokers, an entity previously confirmed to have leaked authentic malware used by the NSA to attack computers around the world, today released another cache of what appears to be extremely potent (and previously unknown) software capable of breaking into systems running Windows. The software could give nearly anyone with sufficient technical knowledge the ability to wreak havoc on millions of Microsoft users.

The leak includes a litany of typically codenamed software “implants” with names like ODDJOB, ZIPPYBEER, and ESTEEMAUDIT, capable of breaking into — and in some cases seizing control of — computers running version of the Windows operating system earlier than the most recent Windows 10.

The vulnerable Windows versions ran more than 65 percent of desktop computers surfing the web last month, according to estimates from the tracking firm Net Market Share.

Fixed in last month's (March) MS17-010 patch update.

https://blogs.technet.microsoft.com/msrc/2017/04/14/protecting-customers-and-evaluating-risk/

- "EternalBlue" Addressed by MS17-010
- "EternalRomance" Addressed by MS17-010
- "EternalSynergy" Addressed by MS17-010
- "EmeraldThread" Addressed by MS10-061
- "EsikmoRoll" Addressed by MS14-068
- "EternalChampion" Addressed by CVE-2017-0146 & CVE-2017-0147
- "EducatedScholar" Addressed by MS09-050
- "EclipsedWing" Addressed by MS08-067
- "ErraticGopher" Addressed prior to the release of Windows Vista

Links:
https://isc.sans.edu/forums/diary/ETERNALBLUE+Windows+SMBv1+Exploit+Patched/22304

The Punycode Problem

UNICODE -> PUNYCODE

- https://www.xn--80ak6aa92e.com
- Displays as: https://ww.apple.com
- Copies from the URL as: https://www.?????.com/

- Edge, IE and Safari are not vulnerable.
- Chrome is already fixed in Chrome 59.
- Firefox is still susceptible.
    - about:config -> search "puny"
    - "network.IDN_show_punycode" -- change to "true"
    - Test with:
    - https://www.xn--e1awd7f.com/
    - https://www.xn--80ak6aa92e.com

For additional terrific information:

- https://www.xudongz.com/blog/2017/idn-phishing/
- https://www.wordfence.com/blog/2017/04/chrome-firefox-unicode-phishing/
- https://www.engadget.com/2017/04/17/google-chrome-phishing-unicode-flaw/

Motherboard: Your Government's Hacking Tools Are Not Safe
https://motherboard.vice.com/en_us/article/your-governments-hacking-tools-are-not-safe
Joseph Cox, writing for Motherboard noted:

From Cellebrite, to Shadow Brokers, to the CIA dump, so many recent data breaches have shown there is a real risk of exposure of government hacking tools.

The hackers will get hacked.

Recent data breaches have made it startlingly clear hacking tools used by governments really are at risk of being exposed. The actual value of the information included in each of these dumps varies, and some may not be all that helpful in and of themselves, but they still highlight a key point: hackers or other third parties can obtain powerful tools of cyber espionage that are supposedly secure. And in most cases, the government does not appear to clean up the fallout, leaving the exploits open to be re-used by scammers, criminals, or anyone else—for any purpose.

Discussion...

As we know, this was my own takeaway from the Vault7 leaks. We have seen a clear pattern of our intelligence and law enforcement agencies inability to keep their own secrets. I DO NOT blame them, per se, since no large human-based organization can perfectly keeps secrets. ALL of the evidence continues to show this. And the CIA, NSA & FBI are not exceptions. This in turn argues against ANY form of "mono-key" technology for access into encrypted communications. Companies will likely be required to modify their technologies as that they can comply with legally issued court order and not be held in contempt of court and heavily fined. But its crucial that we maintain a heterogeneous encryption terrain and not force all companies to provide the equivalent of a generic encryption backdoor which would allow law enforcement to unilaterally obtain access to all "lawfully" encrypted data.

Exploit revealed for remote root access vulnerability affecting many router models
As we covered at the time, a little over four years ago, back in January of 2013, researchers from DefenseCode responsibly revealed the existence of a remote root access vulnerability in the default installation of some Cisco Linksys (now Belkin) routers.

DefenseCode:

- http://defensecode.com/whitepapers/From_Zero_To_ZeroDay_Network_Devices_Exploitation.txt

Our podcast listeners will recall that I immediately added a test for this to ShieldsUP!. Today, GRC's test for publicly exposed UPnP has found 51,854 positive (exposed) results. The ShieldsUP system maintains a 1024-IP deep MRU list to avoid multi-counting repeat tests.

Looking at the list of vulnerable routers (about 1/2 way into the DefenseCode disclosure) it's clear that pretty much all routers exposing UPnP publicly were vulnerable at the time. I'm not going to bother with a test for exploitability, since the UPnP was *NEVER* intended to be bound to any router's WAN interface. It is strictly a LAN-side internal network protocol.

Remember that all the implementors would have needed to do was set all UPnP packet TTL's to 1... since LAN don't route, and all UPnP packets would die at the first public router encountered.

Next-Generation Ad-Blocking
Three guys at Princeton and Stanford University's Jonathan Mayer (whom we've not heard much from for a while) have published a paper titled: "The Future of Ad Blocking: An Analytical Framework and New Techniques"
http://randomwalker.info/publications/ad-blocking-framework-techniques.pdf

It's 17 pages of academic review of the legal and technical history of advertisement blocking. Then they experiment with manipulating the DOM (the document object model) which describes the visual structure of the web page, to "cover up" with a white translucent overlay, those components that they heuristically determine are likely to be ads.

This is undetectable because all of the page's HTML assets are still fetched, but the user is presumably less annoyed because "visual distance" has been created between the page's non-advertising content and its advertising content.

The trouble, of course, is that the other benefits of traditional ad-blocking -- preventing tracking and malvertising -- are not blocked.

But perhaps a hybrid solution makes sense?

- Block everything unless the site actively objects.
- Then allow but "fade" the advertising content.

John Schneider (@09KR0058)

- @SGgrc Would you please address the pros/cons of hiding SSIDs on home routers in a future SN podcast? iOS 10 warns this is a bad thing.

Discussion

Doing this is mostly cosmetic and doesn't provide MUCH security.

All OSes that "connect automatically" broadcast the SSIDs (hidden or not) of all the networks they have connected to in the past. So while the non-broadcast access point's SSID is not being broadcast to anyone new, it IS being subsequently broadcast by all devices that once connected to it.

And... the over-the-air traffic also contains the non-broadcast SSID. So anyone actively sniffing radio traffic will STILL see the access point's SSID if any device is currently associated with it.

Windows 7 and subsequent can be configured not to automatically connect to hidden networks, but then the SSID and password won't be remembered and must be manually reentered every time.

iOS and macOS *always* connect to all known networks, hidden or not.

Miscellany

Burger King Ad deliberately ends by saying: "Ok Google, what is the Whopper burger?"
https://arstechnica.com/gadgets/2017/04/google-burger-king-feud-over-control-of-the-google-assistant/

From our "Epically Cool Ways to Waste Time" Department, we have:
"Generating Sequences of Primes in Conway's Game of Life"
Nathaniel Johnston, an Assistant Professor at the Mount Allison University at Sackville, New Brunswick, Canada
http://www.njohnston.ca/2009/08/generating-sequences-of-primes-in-conways-game-of-life/

<quote> One of the most interesting patterns that has ever been constructed in Conway's Game of Life is primer, a gun that fires lightweight spaceships that represent exactly the prime numbers. It was constructed by Dean Hickerson way back in 1991, yet arguably no pattern since then has been constructed that's as interesting. It seems somewhat counter-intuitive at first that the prime numbers, which seem somehow "random" or "unpredictable", can be generated by this (relatively simple) pattern in the completely deterministic Game of Life.

The gun works by firing lightweight spaceships westward, and destroying them via glider guns that emulate the Sieve of Eratosthenes. A lightweight spaceship makes it past the left edge of the gun at generation 120N if and only if N is a prime number (though for technical reasons, 2 and 3 are not outputted).

Justin (@linuxsysad)

- @SGgrc I have been watching your show for years. I know you love kindle. Which one do you own? How do you prefer reading your ebooks?

Chris Hall (@chall1600) Saturday, 4/15/17, 1:19 PM

- @SGgrc Ryk Brown's Frontiers Saga is amazing; on 3rd book since Tuesday this week. Highly recommend this to any Stak Trek fan hungry for more.

SpinRite

- Patrick McFarland (@pmcfarlandia)
  @SGgrc SpinRite saved the day again!
  My son's Xbox 360 hard drive was failing.
  Plugged it into a PC and ran SR on Level 4. Good as new!

SpinRite recovering a dead PC-based high-end arcade game...

Last Week Follow-ups

The unseen tracking token!... your eMail address!
Whoops!

"Which VPN Services Keep You Anonymous in 2017?"

- https://torrentfreak.com/vpn-services-anonymous-review-2017-170304/

- VPN services have become an important tool to counter the growing threat of Internet surveillance. Encrypting one's traffic through a VPN connection helps to keep online communications private, but is your VPN truly anonymous? We take a look at the logging policies of dozens of top VPN providers.

  Millions of Internet users around the world use a VPN to protect their privacy online.

Unfortunately, however, not all VPN services are as private as you might think. In fact, some are known to keep extensive logs that can easily identify specific users on their network.

This is the main reason why we have launched a yearly VPN review, asking providers about their respective logging policies as well as other security and privacy aspects. This year's questions are as follows:

1. Do you keep ANY logs which would allow you to match an IP-address and a time stamp to a user/users of your service? If so, what information do you hold and for how long?

2. What is the registered name of the company and under what jurisdiction(s) does it operate?

3. Do you use any external visitor tracking, email providers or support tools that hold information about your users/visitors?

4. In the event you receive a takedown notice (DMCA or other), how are these handled?

5. What steps are taken when a valid court order or subpoena requires your company to identify an active user of your service? Has this ever happened?

6. Is BitTorrent and other file-sharing traffic allowed (and treated equally to other traffic) on all servers? If not, why?

7. Which payment systems do you use and how are these linked to individual user accounts?

8. What is the most secure VPN connection and encryption algorithm you would recommend to your users?

9. How do you currently handle IPv6 connections and potential IPv6 leaks? Do you provide DNS leak protection and tools such as "kill switches" if a connection drops?

10. Do you offer a custom VPN application to your users? If so, for which platforms?

11. Do you have physical control over your VPN servers and network or are they hosted by/accessible to a third party? Do you use your own DNS servers?

12. What countries are your servers located in?

"bobbob1016" (@bobbob1016)

- @SGgrc SN607, privacy section doesn't mention not using Google.
  You say Google gets first party on redirect, but what if you're not using it.

Nick Bedford (@nickjbedford)

- @SGgrc I just encountered an issue with blocking third party cookies.
  I couldn't login to Jetstar Airlines account until I unblocked them :/

David Benedict (@bippy_b)

- @SGgrc listening to SN607.
  So would having your ISP NAT connections be a bonus now to your privacy?

johnmoehrke (@johnmoehrke)

- @SGgrc Oauth identity providers also use redirection…
  And get to track your movements... At least where Oauth is used

Ryan Dlugosz (@lbwski)

- @SGgrc important caveat to DNSCrypt: destination site names still leak if that site is using SNI to do HTTPS. Many sites use SNI these days.

Malcolm Hannan-Smith (@MHS3637)

- @SGgrc Re: SN607 "Proactive Privacy (Really!)" it is VERY important to live test an active #VPN, with a site like whoer.net
- @SGgrc Re: #VPN testing. I used proXPN VPN last year. It said the PC was "safe" but testing showed my real ISP IP Address was visible.

  https://whoer.net/

  Disable WebRTC
  Firefox: about:config / media.peerconnection.enabled -> False
  https://whoer.net/blog/article/how-to-disable-webrtc-in-various-browsers/