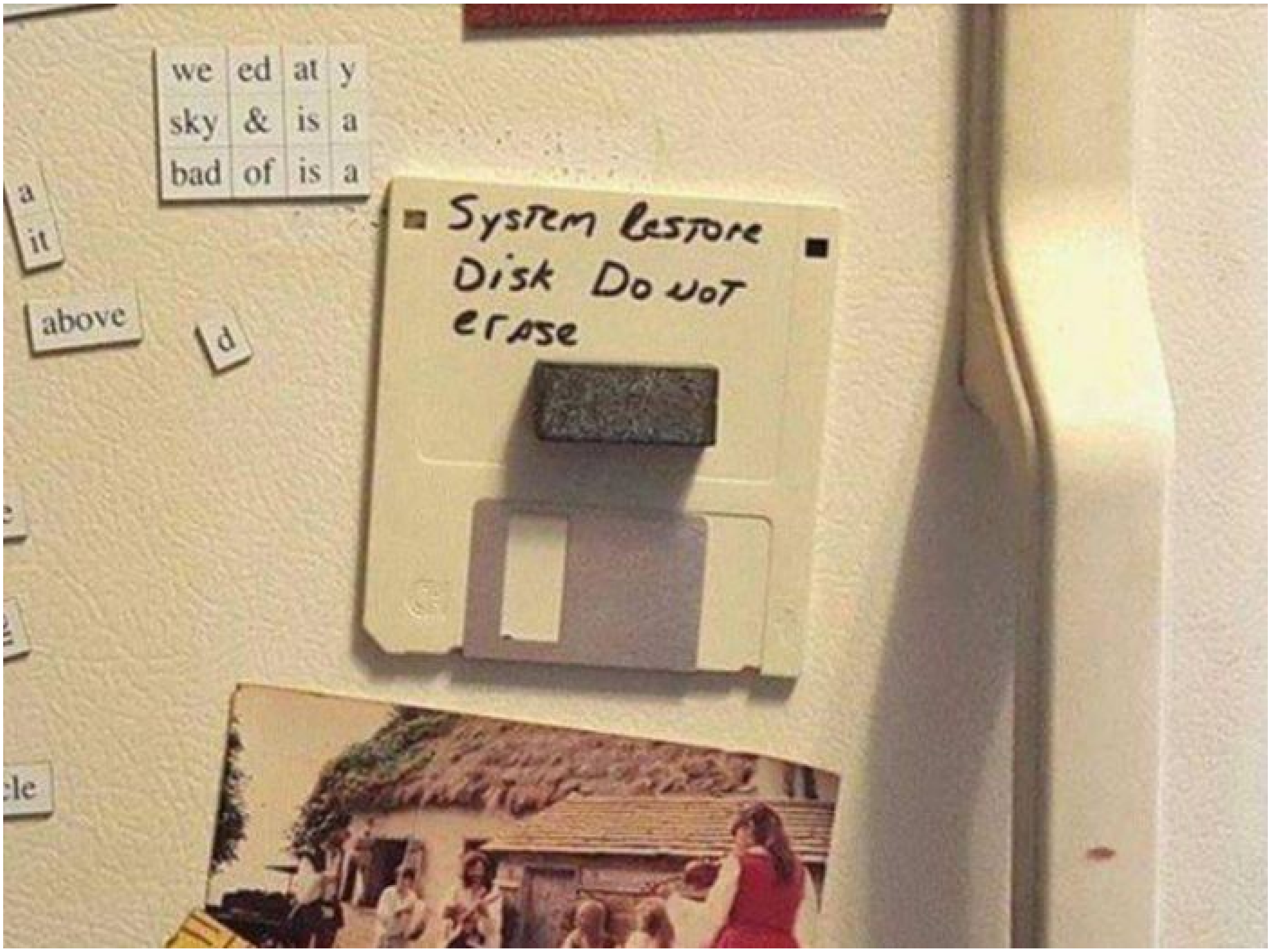
Security Now! #669 - 06-26-18
Cellular Location Privacy

This week on Security Now!

This week we examine some new side-channel worries and vulnerabilities, did Mandiant "hack back" on China?, more trouble with browsers, the big Google Firebase mess, sharing a bit of my dead system resurrection, and a look at the recent Supreme Court decision addressing cellular location privacy.

Our Picture of the Week



Security News

More side-channel worries:
OpenBSD, the branch of the UNIX project which prioritizes security above all else (and whose reputation is well deserved) has announced that it will be suspending all support for hyper-threading.

TLBleed
Theo de Raadt, who heads the OpenBSD project told iTWire last Thursday that he had spent about a month trying to discern the problems that he said "Intel would not disclose to us".

He said that a paper, due to be presented at the Black Hat USA 2018 conference in August "was shared with us".

https://www.blackhat.com/us-18/briefings/schedule/#tlbleed-when-protecting-your-cpu-caches-is-not-enough-10149
"TLBleed: When Protecting Your CPU Caches is Not Enough"
by Ben Gras

Ben and I have corresponded in the past following our coverage of their weaponizing of ROWHAMMER in “Flip Feng Shui”... and, in our episode #577 back in 2016:

STEVE: I received a tweet from one of the security research team at VU Amsterdam's group that's run by Professor Herbert Bos, who we talked about. They're the guys who did the Flip Feng Shui work which was so impressive. Ben Gras wrote: "As one of the authors, thank you for your knowledgeable and detailed exposition. We're honored by it and your kind words." And I replied, I said: "Oh, wow, thanks." And so in email he followed up, saying: "Thank you again for your Flip Feng Shui coverage. Because of more recent coverage, I am a bit of an expert on how well our work is understood; and I don't hesitate to rate your coverage at a 95th percentile rating for expertise and quality of exposition."

So... Ben and his team, presumably lead by their Professor Herbert Bos, have been up to more mischief. Their BlackHat talk abstract reads:

We present TLBleed, a novel side-channel attack that leaks information out of Translation Lookaside Buffers (TLBs). TLBleed shows a reliable side channel without relying on the CPU data or instruction caches. This therefore bypasses several proposed CPU cache side-channel protections, such as page coloring, CAT, and TSX. Our TLBleed exploit successfully leaks a 256-bit EdDSA key from cryptographic signing code, which would be safe from cache attacks with cache isolation turned on, but would no longer be safe with TLBleed. We achieve a 98% success rate after just a single observation of signing operation on a co-resident hyperthread and just 17 seconds of analysis time. Further, we show how another exploit based on TLBleed can leak bits from a side-channel resistant RSA implementation. We use novel machine learning techniques to achieve this level of performance. These techniques will likely improve the quality of future side-channel attacks. This talk contains details about the architecture and complex behavior of modern, multilevel TLB's on several modern Intel microarchitectures that is undocumented, and will be publicly presented for the first time.

So anyway... back to OpenBSD:

De Raadt said he could not be more specific about the nature of the vulnerability that had led to the disabling of hyperthreading as the paper was yet to be published.

"It was shared with us, allowing us to fix it. The solution is trivial, and temporary. Maybe Intel tells us a different way. I'm waiting.... I'm waiting.... I better not keep waiting," he said.

"Only Tier-1 companies received advance information, and that is not responsible disclosure – it is selective disclosure," De Raadt told iTWire at the time. "Everyone below Tier-1 has just gotten screwed."

Regarding the reason why hyperthreading had been disabled, de Raadt said the presentation detailed a further class of problems similar to Spectre, relating to Intel's version of simultaneous multi-threading and hyperthreading.

"In particular, it is ill-advised to run different security domains (address spaces) on a pair of hyperthread CPUs," he said. "Maybe there are other ways to resolve this problem, but Intel isn't sharing solutions with us. We have selected the expedient approach of disabling hyperthreading until we know more."

He said it was a "particularly difficult time to do clever things in the process scheduler, since we believe there are further speculative execution problems coming down the pipeline".

iTWire noted that OpenBSD has a good reputation for security and runs some of the public servers with the longest uptimes. De Raadt himself is obsessed with security, and has as his aim the provision of an operating system that has a secure default install. OpenBSD has had only two remote vulnerabilities in its default install since the project forked from NetBSD in 1996.

Said de Raadt: "Oftentimes mitigating risk due to a problem isn't the same as having all the information to weaponise it. I think weaponising this requires the whole paper (that is to be presented at Black Hat USA 2018 in August).

"I guess Intel and the clowncomputing companies are really going to hate this paper," he added, referring to cloud computing firms. "Alternative solutions are going to be thin on providing security guarantees, or make hyperthreading far less usable compared to increased complexity in scheduling."

In a detailed post last Tuesday, OpenBSD maintainer Mark Kettenis said such processor implementations could lead to Spectre-style timing attacks.

"SMT (Simultaneous multithreading) implementations typically share TLBs and L1 caches between threads," Kettenis wrote. "This can make cache timing attacks a lot easier, and we strongly suspect that this will make several Spectre-class bugs exploitable."

I've spoken of the TLB recently. This is the insane multi-level mapping between the logical presentation of main memory to applications and the actual physical layout of RAM. It's the mechanism underlying what we now glibly refer to as "virtual memory". TLBs is the mechanism which implements memory virtualization.

Hyperthreading offers a sort of "pseudo-core" by allowing instantaneous switching of a single-threaded core between two thread contexts. This was clever when it was invented and since it offers an inexpensive though modest performance boost, it has remained.

But, unlike truly separate cores, Hyperthreading means that two separately executing threads are sharing the same TLB memory mapping hardware, and the TLB lookup tables are INHERENTLY another form of caching. This allows one thread to force TLB content eviction and by then measuring the time required for its own operations to infer what the thread it shares the core with has been doing with its time.

This is not the end of the world... but in the longer term this year's revelations are going to require some serious rethinking of how SMT -- simultaneous multi-threading -- is implemented by future processor architectures, operating systems, and processor virtualization systems.

Until this year we have just been throwing everyone into one big homogeneous processing pool and saying "have fun everyone and get as much work done as you can." But that happy naiveté has been shown to be fundamentally unsafe.

Did FireEye's Mandiant group "hack back" at China's APT1?
"The Perfect Weapon: War, Sabotage, and Fear in the Cyber Age" by David E. Sanger, New York Times national security journalist and professor of national security policy at Harvard's Kennedy School of Government. Kindle: $15 / Hardback: $20 / Audible.

One verified purchaser and author wrote:

I came to this book the long way around. Knowing that I had just published a military thriller in which North Korea crashes the electrical grid for the greater DC area, my brother-in-law sent me a link to David Sanger’s recent interview on NPR. Listening to Mr. Sanger confirmed some of the scariest parts of my own research. I discovered that my fictional scheme for robbing the U.S. government of electrical power is uncomfortably similar to an actual cyber attack that flat-lined large segments of the Ukrainian grid in 2015. Far from being worst-case imaginary scenarios, some of the concepts I’ve written about have already played out in the real world, usually in countries distant from the United States and under circumstances that either don’t make the news or don’t create an impression on the public consciousness.

I burned through this book in less than a day. The Perfect Weapon has the page-flipping intensity of the best techno-thriller novels, with the gravitas of meticulously-sourced nonfiction. If I had to sum up this book in one word, it would be “terrifying.”

With true stories from the cyber sabotage of the Democratic Campaign Committee to the penetration of the White House computer networks, this book is a wake-up call for our technology-dependent civilization. I just hope we don’t hit the ‘snooze’ button and go back to sleep.

Bleeping Computer has some terrific coverage and a summary of the controversy:

US cyber-security firm FireEye has denied claims that have been ramping up on social media all last week about illegally "hacking back" a Chinese nation-state cyber-espionage group.

The claims and social media discussions started after the publication of "The Perfect Weapon: War, Sabotage, and Fear in the Cyber Age," a book authored by David Sanger, New York Times national security journalist.

In the book, Sanger recounted a series of events from 2013, in the lead up to FireEye publishing a report called "APT1, Exposing One of China’s Cyber Espionage Units."

At the time, the report was a landmark moment in the cyber intelligence community, as it exposed the activities of Chinese hackers in a depth of details like never before, even going as far as pinning the hacking on Unit 61398 of China's People Liberation Army (PLA), an attribution level unheard at the time.

Sanger's side of the events

But according to Sanger's book, FireEye might have obtained all these details while "hacking back," a term used to describe the practice of using offensive hacking techniques to breach an attacker's systems to determine his identity, what he stole, and even destroy some of the stolen data in order to protect the victim. The technique is illegal, under the terms of US law, and limited to approved US military personnel only.

In his book, Sanger claims Mandiant (former cyber-security firm behind the report and purchased by FireEye a year later in 2014), allowed him to sit down with its security researchers during one of these incidents.

Passages from Sanger's book are excerpted in this tweet:

Thomas Rid is a political scientist best known for his work on the history and risks of information technology in conflict. He is Professor of Strategic Studies at Johns Hopkins University's School of Advanced International Studies.

Thomas tweeted: Previously unreported details on some of the evidence & methods used by Mandiant to attribute China’s PLA Unit 61398 in the landmark 2013 APT1 report. Source: David Sanger’s new book https://twitter.com/RidT/status/1010479475157499909

<quoting from Sanger's book>
Ever resourceful, [Mandiant CEO Kevin Mandia's] staff of former intelligence officers and cyber experts tried a different method of proving their case. They might not be able to track the IP addresses to the Datong Road high-rise itself, but they could actually look inside the room where the hacks originated. As soon as they detected Chinese hackers breaking into the private networks of some of their clients—mostly Fortune 500 companies—Mandia's investigators reached back through the network to activate the cameras on the hackers' own laptops. They could see their keystrokes while actually watching them at their desks.

The hackers, just about all of them male and most in their mid twenties, carried on like a lot of young guys around the world. They showed up at work about eight-thirty a.m. Shanghai time, checked a few sports scores, emailed their girlfriends, and occasionally watched porn. then, when the clock struck nine, they started methodically breaking into computer systems around the world, banking on the keyboards until a lunch break gave them a moment to go back to the scores, the girlfriends, and the porn.

One day I sat next to some of Mandia's team, watching the Unit 61938 hacking corps at work; it was a remarkable sight. My previous mental image of PLA officers was a bunch of stiff old generals sitting around in uniforms with epaulets, reminiscing about the glory days with Mao. But these guys were wearing leather jackets or just undershirts, and probably saw Mao only if they visited his mausoleum in Tiananmen Square. "They were such bros," Andrew Scwartz, one of Mandia's communications specialists, recalled later.

FireEye claims it was a misrepresenation/misunderstanding

But in a statement released today, FireEye refutes these claims. The company says that Sanger mischaracterized what really happened, and might have simply misunderstood what he was shown that day when he was allowed to sit with Mandiant [now FireEye] employees.

FireEye says Sanger never observed real-time hacking, but only pre-recorded videos of APT1 (PLA Unit 61398) operators interacting with computers on the network of compromised companies.

Furthermore, FireEye says it obtained permission from these companies to leave the compromised PCs intact and observe what the hackers were doing, and that at no point its employees used offensive hacking techniques.

Specifically, Mr. Sanger suggests our "…investigators reached back through the network to activate the cameras on the hackers' own laptops." We did not do this, nor have we ever done this. To state this unequivocally, Mandiant did not employ "hack back" techniques as part of our investigation of APT1, does not "hack back" in our incident response practice, and does not endorse the practice of "hacking back."
[...]
The conclusion that we hacked back, while incorrect, is understandable.
[...]
To someone observing this video "over the shoulder" of one of our investigators, it could appear as live system monitoring. Nevertheless, Mandiant did not create these videos through "hacking back" or any hacking activity. All of these videos were made through information obtained via consensual security monitoring on behalf of victim companies that were compromised.
[...]
The videos Mr. Sanger viewed were from Windows Remote Desktop Protocol (RDP) network packet captures (PCAP) of Internet traffic at these victim organizations. Mandiant has never turned on the webcam of an attacker or victim system.
[...]
In short, we do not fight hackers by hacking, but by diligently and legally pursuing attribution with a rigor and discipline that the cause requires.

The company released one of the videos they recorded of APT1 hackers active on one of these compromised PCs.

In an emailed response, Sanger responded to Mandiant's statement from earlier in the day:

"Mandiant gave us extraordinary access to their investigation as we were preparing to write about Unit 61398 in late 2012, and the result was our story in the Times, and the company's report, in February, 2013. I spent considerable time with their investigators, and saw the images of the hackers as described in "The Perfect Weapon." Mandiant now says that all those images came from "consensual monitoring" — in other words, that everything they received, from code to message traffic to imagery, was visible because the hackers themselves were transmitting them in the course of breaking into the systems owned by Mandiant's clients. While that wasn't my understanding at the time, passive monitoring is reasonable explanation of how the company came to link the hacks to specific individuals, several of whom have since been indicted by the United States."

Some industry insiders told CyberScoop they were less than shocked by the claims. Broadly speaking, 'hacking back' to gain novel insight into an attacker is fairly well known, despite it being illegal under U.S. law. The practice is often the subject of commonplace rumors amongst the industry.

WebAssembly, Meltdown, Spectre & TLBleed

WebAssembly is a recently developed technology whose adoption across the browser landscape was surprisingly swifter than for other technologies we've seen previously. It is currently supported by all of the major browsers including Chrome Firefox, Edge and Safari. The probable rationale for its rapid adoption was likely that it would provide such a tangible and significant performance boost that none of the browsers wanted to be left behind and labelled slow.

As we've covered previously, WASM provides a compact binary language that a browser can quickly convert into machine code for direct execution on the host processor. It's a web standard that defines a binary format and a corresponding assembly-like text format for executable code in Web pages... and it enables running web-sourced code nearly as fast as running native machine code.

(Note: ALL CRYPTOCURRENCY MINING is being done with WebAssembly. This explains my often stated quandary about the inefficiency of JavaScript-based mining: It's WASM that makes browser hijack mining possible.)

As we know, exploitation of the Spectre and Meltdown vulnerabilities requires that code executing on processor being shared with target/victim code and data is able to precisely measure the length of time various operations take to complete.

We also know that the nature of information leakage is that someone you don't trust is able to determine something about your system. This is why cloud-based attacks with shared cloud servers are a concern. The other obvious opportunity would be web-based attacks where untrusted web page code is allowed to run on a target/victim's machine.

Consequently, the immediate reaction of all of the browser vendors was to reduce the resolution of the clocks and timers accessible to JavaScript code.

But now, security researchers have begun to worry that the WebAssembly feature development roadmap calls for the addition of a feature known as SharedArrayBuffers (SAB) which would allow clever WebAssembly code to obtain the high-precision timing information required to pull off high-resolution timing attacks... and this time it's unclear how to thwart the exploitation without seriously undermining the performance which WebAssembly is all about.

[[*Explain how a shared array buffer can be leveraged into providing high resolution timing without any timing API function.*]]

As a consequence of this, web browsers have temporarily disabled the SharedArrayBuffer API in JavaScript to thwart Spectre & Meltdown exploits.

In the WebAssembly discussions the developers are punting on SAB and are choosing not to implement the feature until -- hopefully -- the bigger underlying processor problems can be addressed.

This makes a mess, however, since that means that the presence of some core APIs in JavaScript and WASM might be dependent upon whether the underlying machine CPU microcode and OS have introduced their own mitigations for Meltdown and Spectre.

And then we have TLBleed from Ben Gras & Co... which may require the disabling of hyperthreading, or at least intelligent thread scheduling.

The Firebase Mess

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

https://firebase.google.com/

Firebase helps mobile app teams succeed.

Build apps fast, without managing infrastructure
Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

Backed by Google, trusted by top apps
Firebase is built on Google infrastructure and scales automatically, for even the largest apps.

One platform, with products that work better together
Firebase products work great individually but share data and insights, so they work even better together.

- Cloud Firestore: Store and sync app data at global scale
- Cloud Functions: Run mobile backend code without managing servers
- Authentication: Authenticate users simply and securely
- Cloud Storage: Store and serve files at Google scale
- Realtime Database: Store and sync app data in milliseconds

http://info.appthority.com/-q2-2018-mtr-download-Firebase-vulnerability

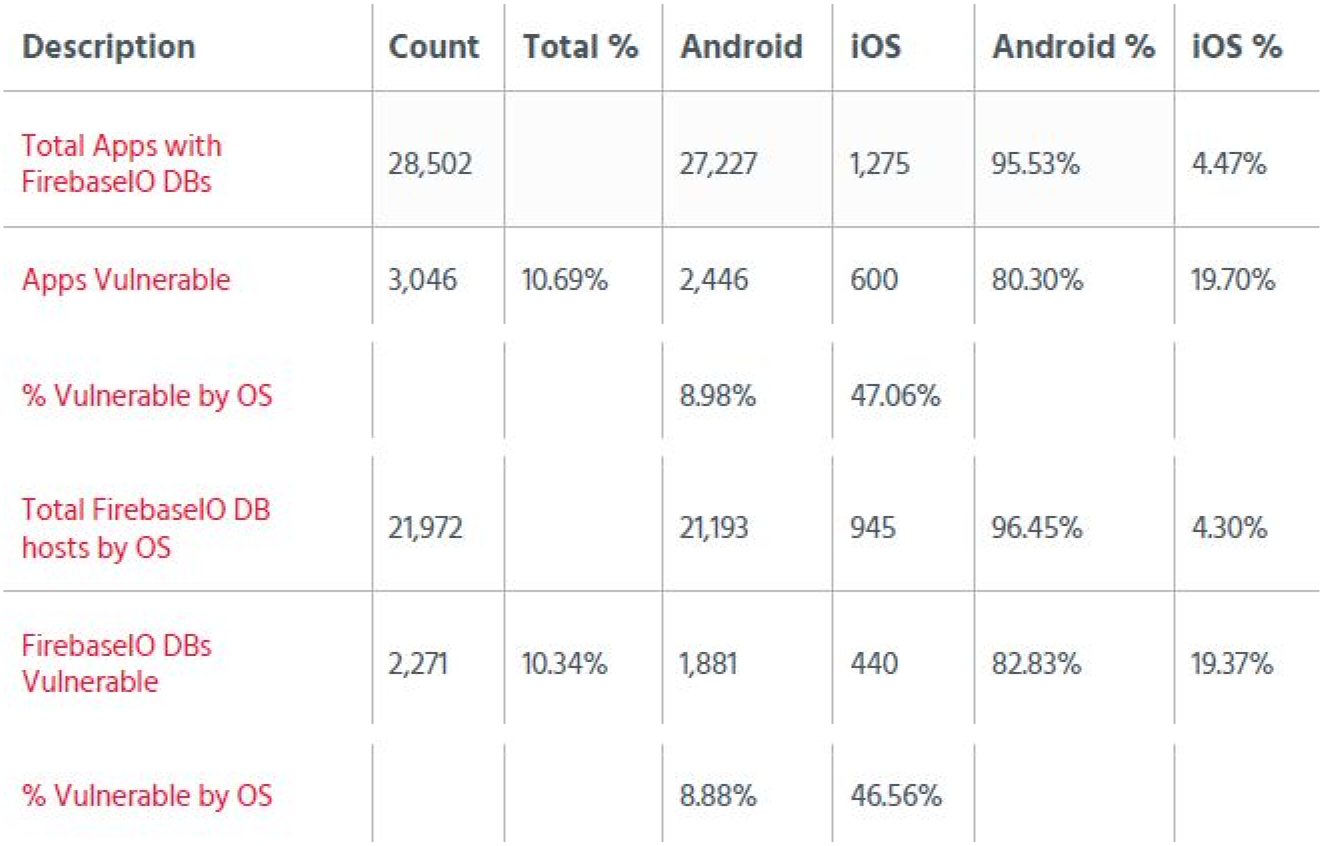
Appthority

In 2017, the Appthority Mobile Threat Team (MTT) discovered the HospitalGown vulnerability.

The vulnerability, named for data leaking through backend data stores that are unsecured, results from app developers' failure to properly secure back-end servers with firewalls and authentication, leading to data exposure. Our initial report in May 2017, revealed that weakly secured backend databases were being accessed via apps used by employees, partners and customers and resulted in numerous security risks, including extensive leaks of sensitive data, easier data access and exfiltration, and increased risks for spear phishing, social engineering, data ransom and other attacks.

This report focuses on the MTT's latest discovery, a new variant of the HospitalGown vulnerability which occurs when app developers fail to require authentication to a Google Firebase cloud database. Firebase is one of the most popular backend database technologies for mobile apps but does not secure user data by default, provide third-party encryption tools, or alert developers to insecure data and potential vulnerabilities. To secure data properly, developers need to specifically implement user authentication on all database tables and rows, which rarely happens in practice. Moreover, it takes little effort for attackers to find open Firebase app databases and gain access to millions of private mobile data app records.

The challenge for app developers is that Firebase does not provide adequate security capabilities out of the box. The only security feature available to developers is authentication and rule-based authorization. However, Firebase does not secure user data by default nor are third-party tools available to provide encryption for it. And finally, Firebase does not provide security checkup reports to identify insecure data and potential vulnerabilities. The only way to secure data in a Firebase data store is for the developer to explicitly implement user authentication on all database rows and tables. Moreover, it takes little effort for attackers to find open Firebase app databases. Once found, cybercriminals can gain access to millions of private mobile data app records by simply appending"/.json" to the server URL, e.g., "https://appname.firebaseio.com/.json". The result is a trove of data that is open to the public internet unless the developer explicitly imposes user authentication on each individual table or directory. Even when developers do implement authentication, they may not secure every database table.



Description	Count	Total %	Android	iOS	Android %	iOS %
Total Apps with FirebaseIO DBs	28,502		27,227	1,275	95.53%	4.47%
Apps Vulnerable	3,046	10.69%	2,446	600	80.30%	19.70%
% Vulnerable by OS			8.98%	47.06%		
Total FirebaseIO DB hosts by OS	21,972		21,193	945	96.45%	4.30%
FirebaseIO DBs Vulnerable	2,271	10.34%	1,881	440	82.83%	19.37%
% Vulnerable by OS			8.88%	46.56%		

Source: http://info.appthority.com/-q2-2018-mtr-download-Firebase-vulnerability

SCOPE OF THE THREAT

Out of a total of 2,705,987 apps analyzed, 27,227 Android apps and 1,275 iOS apps were found to be connected to a Firebase database. Of those connected apps, we found that:

- 1 in 11 Android apps (9%) and almost half of iOS apps (47%) that connect to a Firebase database were vulnerable

- More than 3,000 apps were leaking data from 2,300 unsecured servers. Of these, 975 apps were in active customer environments.

- 1 in 10 Firebase databases (10.34%) are vulnerable

- Vulnerable Android apps alone were downloaded over 620 million times

- Over 100 million records (113 gigabytes) of data was exposed

ORGANIZATIONS IMPACTED

- Most organizations are affected. We found that 62% of enterprises have at least one vulnerable app in their mobile environment.

- No type of organization is immune. Organizations with data exposed to this HospitalGown variant included banks, telecoms, postal services, ride sharing companies, hotels and educational institutions.

- Organizations worldwide were affected. Enterprises in the United States, Europe, the United Kingdom, Argentina, Brazil, Singapore, Taiwan, New Zealand, India, and China were exposed.

DATA LEAKED

Apps connected to unsecured Firebase databases have exposed more than 100 million data records, including:

- 2.6 million plain text passwords and user IDs

- 4 million+ PHI (Protected Health Information) records, including chat messages and prescription details

- 25 million GPS location records

- 50 thousand financial records including banking, payment and Bitcoin transactions

- 4.5 million+ Facebook, LinkedIn, Firebase and corporate data store user tokens

Regulated Data

Regulated data is being leaked: Some of the data leaked includes highly sensitive private information subject to regulatory requirements such as HIPAA, GDPR, and PCI. Such leaks can trigger expensive regulatory fines and breach notification requirements which can damage the company’s reputation. Exposed regulated data included:

• Medical information: Chat messages between patients and pharmaceutical sales representatives together with their prescriptions and orders leaked from a Mexicobased pharmaceutical app. Information about medical consultations, medical history, and diagnostic information was also exposed. This information is protected by HIPAA regulations and subject to breach notification requirements.

• Sensitive personal data: Email addresses, phone numbers, full names, geolocations, and Facebook OAuth tokens were all leaked in violation of data privacy protection laws such as GDPR.

• Vehicle license plate numbers: Some apps exposed vehicle license and registration numbers as well as geolocation data. California has numerous data privacy laws, some of which require companies that expose names accompanied by license plate numbers, and other automobile details to disclose these breaches.

• Credit card numbers: Credit card numbers, which were also exposed, are subject to protection in accordance with PCI DSS.

Personal Data

In some cases, leaked data may not be subject to regulatory requirements, but can nonetheless expose private or financial information. Leaked information of this type included:

- Private messages from a networking app that uses Artificial Intelligence (AI) for match making

- Voice recordings from a voice-based dating app

- Registered email addresses from an unofficial spy recording app that allows users to take videos when the app is not in the foreground

- A cryptocurrency wallet app leaked transaction history and total amount of bitcoins that users owned. This information could allow a bad actor to easily drain cryptocurrency accounts potentially containing hundreds of thousands of dollars in cryptocurrency.

Sensitive Enterprise Data

When internal company data is leaked, organizations can lose intellectual property (IP), damaging their viability and competitiveness. Forty percent of vulnerable apps installed are business related, increasing the risk for IP loss. The app leaks we found included:

- Corporate private keys and access credentials. With this information in hand, cybercriminals have free reign in a corporate network and can potentially exfiltrate sensitive intellectual property, such as patent information and plans for future products.

- Private conversations. Productivity apps leaked data about private business conversations.

- Sales info. Leaks of corporate sales details can give competitors important information about a company’s customers.

Miscellany

Dead system resurrection (from an offline drive)

http://www.ultrageek.com/?p=214

<quote> What if you had a perfectly good working hard drive but the machine that was running it was dead, DOA, not working, or had been upgraded with a new hard drive and OS? Connecting that to a machine and getting the files is easy enough but what about if you wanted to boot the OS back up? The Virtual Option is going to be your best bet but how? Sure there are plenty of options like Ghost that will keep your machine backed up with a VMDK option but how to accomplish this without spending money?

The process is generally referred to as Physical To Virtual or P2V for short. In doing some research on how to do this I've worked out a process that as long as you don't mind forking over a working email address you can download a couple [of free] tools to make this possible.

To accomplish this task you need to convert a drive containing a filesystem into a virtual drive image (vmdk) file to use with VM Ware Workstation or Player.

Create the VMX File used by the converter

Transform the image with the appropriate drivers to boot (HAL, HD, etc.)

- Disk2VHD (Mark Russinovich)
  The only tool I could find to take a mounted drive letter (filesystem) and produce a virtualized drive. (It creates a Windows VHD, but we fix that next.)

- StarWind's V2V converter - convert the VHD to a VMDK
  A free tool to convert among VM families. In this case from the Microsoft VHD to a VMWare VMDK.

- VMWare Converter
  Now we have a VMWare virtual machine... but it's setup to bootup from the previous actual physical (now dead) motherboard, =not= from a VMWare virtualized environment. But the "VMWare vCenter Converter Standalone" can do this.

Feedback Errata

Ciprian in Germany
Subject: Huge Cisco Security patches number
Date: 21 Jun 2018 02:00:51

Hello,
Cisco released again a tons of patches for "big iron routers", I got 24 individual emails.
What I wanted to remind is that Cisco had, is and will acquire companies for the products. And in those purchasing not all the "back doors" are known.
my2c
Regards,

P.S. Currently SpinRiting a huge 24 drive Dell enclosure to see what I can reuse. :)

Cellphone Location Privacy

Increasingly, local police and other law enforcement agencies have been obtaining court orders to compell local cell carriers to disclose the location data of crime suspects. Significantly, these have =not= been "probable cause search warrants" as permitted under the 4th amendment to the Constitution. These have simply been requests from courts for the business record documents of cellular carriers.

https://www.supremecourt.gov/opinions/17pdf/16-402_h315.pdf

TIMOTHY IVORY CARPENTER, PETITIONER v. UNITED STATES

CHIEF JUSTICE ROBERTS delivered the opinion of the Court.

This case presents the question whether the Government conducts a search under the Fourth Amendment when it accesses historical cell phone records that provide a comprehensive chronicle of the user's past movements.

[Opens with a perfect clear succinct explanation of cell tower location, how the increasing popularity of cellular communications has caused providers to increase the density of towers, thus increasing the accuracy of tower-based location, especially in urban areas.]

Summary of the decision:

Cell phones perform their wide and growing variety of functions by continuously connecting to a set of radio antennas called "cell sites." Each time a phone connects to a cell site, it generates a time-stamped record known as cell-site location information (CSLI). Wireless carriers collect and store this information for their own business purposes. Here, after the FBI identified the cell phone numbers of several robbery suspects, prosecutors were granted court orders to obtain the suspects' cell phone records under the Stored Communications Act. Wireless carriers produced CSLI for petitioner Timothy Carpenter's phone, and the Government was able to obtain 12,898 location points cataloging Carpenter's movements over 127 days-an average of 101 data points per day. Carpenter moved to suppress the data, arguing that the Government's seizure of the records without obtaining a warrant supported by probable cause violated the Fourth Amendment. The District Court denied the motion, and prosecutors used the records at trial to show that Carpenter's phone was near four of the robbery locations at the time those robberies occurred. Carpenter was convicted. The Sixth Circuit affirmed, holding that Carpenter lacked a reasonable expectation of privacy in the location information collected by the FBI because he had shared that information with his wireless carriers.

Held:

1. The Government's acquisition of Carpenter's cell-site records was a Fourth Amendment search. Pp. 4-18.

(a) The Fourth Amendment protects not only property interests but certain expectations of privacy as well. Katz v. United States, 389 S. 347, 351. Thus, when an individual "seeks to preserve something as private," and his expectation of privacy is "one that society is prepared to recognize as reasonable," official intrusion into that sphere generally qualifies as a search and requires a warrant supported by probable cause. Smith v. Maryland, 442 U. S. 735, 740 (internal quotation marks and alterations omitted). The analysis regarding which expectations of privacy are entitled to protection is informed by historical understandings "of what was deemed an unreasonable search and seizure when [the Fourth Amendment] was adopted." Carroll v. United States, 267 U. S. 132, 149. These Found-ing-era understandings continue to inform this Court when applying the Fourth Amendment to innovations in surveillance tools. See, e.g., Kyllo v. United States, 533 U. S. 27. Pp. 4-7.

(b) The digital data at issue-personal location information maintained by a third party-does not fit neatly under existing precedents but lies at the intersection of two lines of cases. One set addresses a person's expectation of privacy in his physical location and movements. See, e.g., United States v. Jones, 565 U. S. 400 (five Justices concluding that privacy concerns would be raised by GPS tracking). The other addresses a person's expectation of privacy in information voluntarily turned over to third parties. See United States v. Miller, 425 U. S. 435 (no expectation of privacy in financial records held by a bank), and Smith, 442 U. S. 735 (no expectation of privacy in records of dialed telephone numbers conveyed to telephone company). Pp. 7-10.

(c) Tracking a person's past movements through CSLI partakes of many of the qualities of GPS monitoring considered in Jones-it is detailed, encyclopedic, and effortlessly compiled. At the same time, however, the fact that the individual continuously reveals his location to his wireless carrier implicates the third-party principle of Smith and Miller. Given the unique nature of cell-site records, this Court declines to extend Smith and Miller to cover them. Pp. 10-18.

(1) A majority of the Court has already recognized that individuals have a reasonable expectation of privacy in the whole of their physical movements. Allowing government access to cell-site records-which "hold for many Americans the 'privacies of life,' " Riley v. California, 573 U. S. ___, ___-contravenes that expectation. In fact, historical cell-site records present even greater privacy concerns than the GPS monitoring considered in Jones: They give the Government near perfect surveillance and allow it to travel back in time to retrace a person's whereabouts, subject only to the five-year retention policies of most wireless carriers. The Government contends that CSLI data is less precise than GPS information, but it thought the data accurate enough here to highlight it during closing argument in Carpenter's trial. At any rate, the rule the Court adopts "must take account of more sophisticated systems that are already in use or in development," Kyllo, 533 U. S., at 36, and the accuracy of CSLI is rapidly approaching GPS-level precision. Pp. 12-15.

(2) The Government contends that the third-party doctrine governs this case, because cell-site records, like the records in Smith and Miller, are "business records," created and maintained by wireless carriers. But there is a world of difference between the limited types of personal information addressed in Smith and Miller and the exhaustive chronicle of location information casually collected by wireless carriers.

The third-party doctrine partly stems from the notion that an individual has a reduced expectation of privacy in information knowingly shared with another. Smith and Miller, however, did not rely solely on the act of sharing. They also considered "the nature of the particular documents sought" and limitations on any "legitimate 'expectation of privacy' concerning their contents." Miller, 425 U. S., at 442. In mechanically applying the third-party doctrine to this case the Government fails to appreciate the lack of comparable limitations on the revealing nature of CSLI.

Nor does the second rationale for the third-party doctrine-voluntary exposure-hold up when it comes to CSLI. Cell phone location information is not truly "shared" as the term is normally understood. First, cell phones and the services they provide are "such a pervasive and insistent part of daily life" that carrying one is indispensable to participation in modern society. Riley, 573 U. S., at ___. Second, a cell phone logs a cell-site record by dint of its operation, without any affirmative act on the user's part beyond powering up. Pp. 15-17.

(d) This decision is narrow. It does not express a view on matters not before the Court; does not disturb the application of Smith and Miller or call into question conventional surveillance techniques and tools, such as security cameras; does not address other business records that might incidentally reveal location information; and does not consider other collection techniques involving foreign affairs or national security. Pp. 17-18.

2. The Government did not obtain a warrant supported by probable cause before acquiring Carpenter's cell-site records. It acquired those records pursuant to a court order under the Stored Communications Act, which required the Government to show "reasonable grounds" for believing that the records were "relevant and material to an ongoing investigation." 18 U. S. C. §2703(d). That showing falls well short of the probable cause required for a warrant. Consequently, an order issued under §2703(d) is not a permissible mechanism for accessing historical cell-site records. Not all orders compelling the production of documents will require a showing of probable cause. A warrant is required only in the rare case where the suspect has a legitimate privacy interest in records held by a third party. And even though the Government will generally need a warrant to access CSLI, case-specific exceptions-e.g., exigent circumstances-may support a warrantless search. Pp. 18-22. 819 F. 3d 880, reversed and remanded.