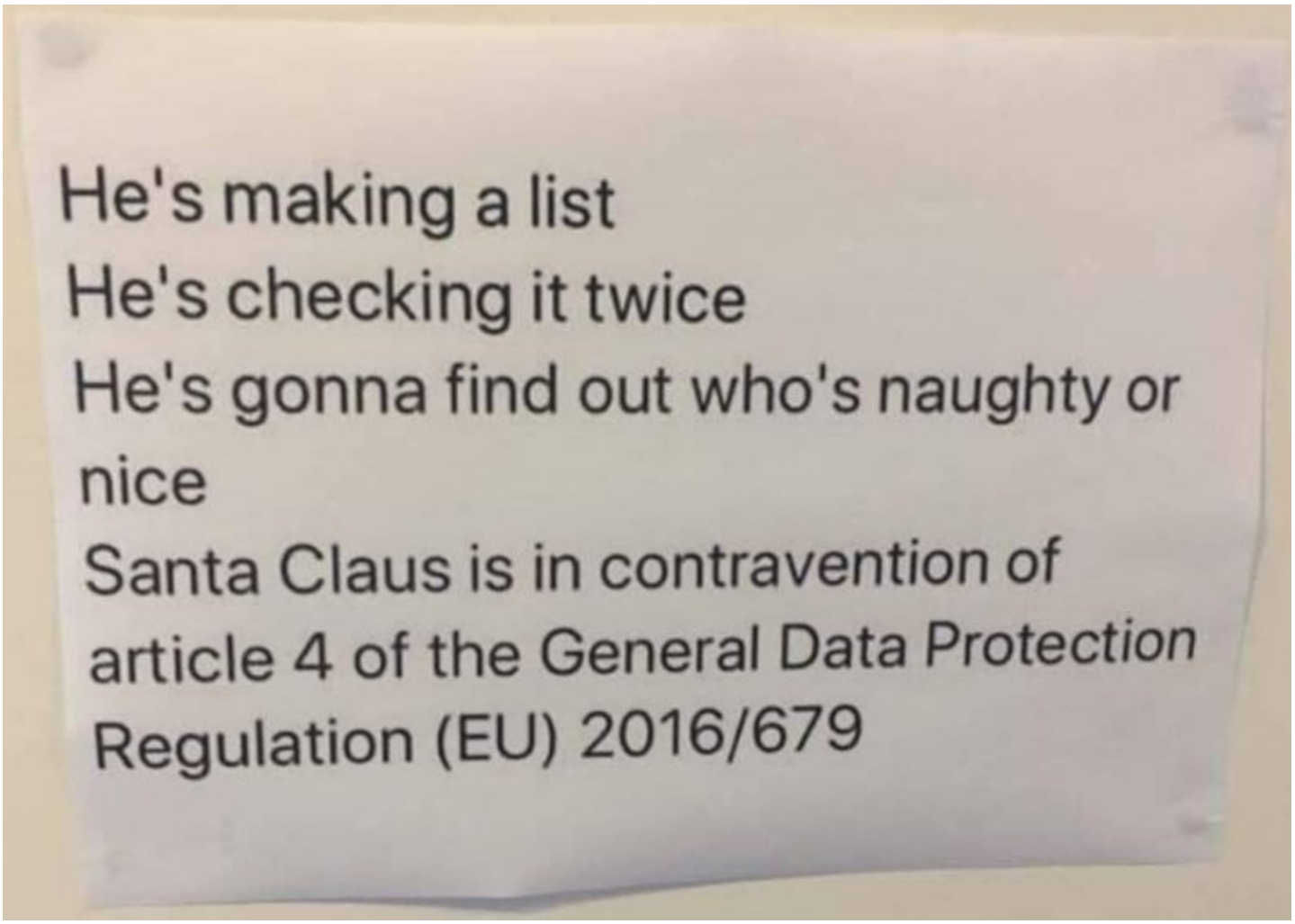
Security Now! #694 - 12-18-18

“The SQLite RCE Flaw”

This week on Security Now!

This week we look at Rhode Island's response to Google's recent API flaw, Signal's response to Australia's anti-encryption legislation, the return of PewDiePie, US border agents retaining traveler's private data, This Week in Android hijinks, confusion surrounding the Windows v5 release, another Facebook API mistake, the 8th annual most common passwords list (AKA "how's monkey doing?"), why all might not be lost if someone is hit with drive encrypting malware, Microsoft's recent 4-month run of 0-day vulnerability patches, the Firefox 64 update, a reminder of an awesome train game for iOS, Mac and Android, some closing the loop feedback with our listeners, and a look at a new and very troubling flaw discovered in the massively widespread SQLite library... and what we can do.



Security News

Rhode Island Government entity sues Google after latest Google+ API leak

Not everyone was as happy as I was over Google's conduct in handling their most recent API leak. Within a day of this announcement of the second leak a class action lawsuit was filed by the Employees Retirement System of Rhode Island (ERSRI), a government-owned investment fund that provides retirement, disability, survivor, and death benefits to state and municipal employees, and public school teachers.

When reading this I was thinking: "So... they're suing Google over an API flaw where there is no evidence that any information was stolen? How was anyone harmed?"

Well, the ERSRI suite accuses Google of intentionally misleading shareholders and federal regulators by failing to disclose its Google+ data leaks in a timely fashion. And although the lawsuit was filed immediately following the second leak report, the lawsuit cites both this most recent and the last October Google+ API incidents.

Just to refresh everyone's memory... Google announced the first at the start of October when the company revealed that a Google+ API bug could have been used to collect the data of over 500,000 users. And also recall that there was apparently a bit of a cover up within Google with the decision too withhold disclosure.

The second incident was announced last Monday, when Google said that a Google+ API update introduced another potential for data leakage which the company patched within a week. Google said this second leak wasn't abused to harvest user data, but if someone did, the data of 52.5 million users was exposed.

So, silly me... I couldn't see any grounds for a lawsuit since where was the damage?

ERSRI officials are claiming that Google's recent string of leaks and supposedly delayed disclosures have harmed Google's own company's stock value, which has, in turn, incurred loses to investors, such as itself.

Rhode Island's General Treasurer Seth Magaziner said: "Google had an obligation to tell its users and investors that private information wasn't being protected. Instead, Google executives decided to hide the breaches from its users and continued to mislead investors and federal regulators. This is an unconscionable violation of public trust by Google, and we are seeking financial restitution on behalf of the Rhode Island pension fund and other investors."

And it does seem that things are becoming more litigious, and it's not clear that this is a good thing. This is the second class-action suit aimed at Google because of its Google+ API leaks. The first one hit the day after the first API leak disclosure. And both Facebook and Marriott were sued shortly after announcing data breaches, even faster than Google, within hours.

I wanted to bring this up, and focus our attention upon it a bit, because as we know, responsible and timely disclosures of mistakes, such as this second one by Google which they caught and killed quickly and quietly, then disclosed, and which as we also know anyone can make no matter how careful they are, form an important piece of good public security conduct.

If companies start being sued the instant they acknowledge that they made and fixed a mistake, where no one was hurt, over the reputation damage their own responsible disclosure incurs, then it's foreseeable that companies are more likely to go back to the bad old days where they were much less forthcoming and less willing to work with others.

I hope that doesn't happen.

Setback in the Outback
https://signal.org/blog/setback-in-the-outback/

Joshua Lund / @joshualund
Sysadmin, programmer, privacy enthusiast, security fan, writer, occasional cyclist... and one of the Signal developers posted last week at signal.org:

[Lightly edited for the podcast] Like many others, we have been following the latest developments in Australia related to the "Assistance and Access" bill with a growing sense of frustration. The widespread adoption of strong cryptography and end-to-end encryption has given people around the world the ability to protect their personal information and communicate securely. Life is increasingly lived online, and the everyday actions of billions of people depend on this foundation remaining strong.

Attempting to roll back the clock on security improvements which have massively benefited Australia and the entire global community is a disappointing development.

More than eight years have passed since we released the public beta of what is now known as Signal. Throughout the entire development process, the project has faced resistance from people who struggle to understand end-to-end encryption or who seek to weaken its effects. This is not a new dynamic.

We can't include a backdoor in Signal, but that isn't a new dynamic either.

By design, Signal does not have a record of your contacts, social graph, conversation list, location, user avatar, user profile name, group memberships, group titles, or group avatars. The end-to-end encrypted contents of every message and voice/video call are protected by keys that are entirely inaccessible to us. In most cases now we don't even have access to who is messaging whom.

Everything we do is open source and anyone is free to verify or examine the code for each release. Reproducible builds and other readily accessible binary comparisons make it possible to ensure the code we distribute is what is actually running on user's devices. People often use Signal to share secrets with their friends, but we can't hide secrets in our software.

Everyone benefits from these design decisions – including Australian politicians. For instance, it has been widely reported that Malcolm Turnbull, the 29th Prime Minister of Australia, is a Signal user. He isn't alone. Members of government everywhere use Signal. Even if we disagree with Christian Porter, we would never be able to access his Signal messages, regardless of whether the request comes from his own government or any other government.

Although we can’t include a backdoor in Signal, the Australian government could attempt to block the service or restrict access to the app itself. Historically, this strategy hasn’t worked very well. Whenever services get blocked, users quickly adopt VPNs or other network obfuscation techniques to route around the restrictions.

If a country decided to apply pressure on Apple or Google to remove certain apps from their [regional] stores, switching to a different region is trivial on both Android and iOS. [And] Popular apps are widely mirrored across the internet. Some of them can even be downloaded directly from their official website.

One of the myriad ways that the “Assistance and Access” bill is particularly terrible lies in its potential to isolate Australians from the services that they depend on and use every day. Over time, users may find that a growing number of apps no longer behave as expected. New apps might never launch in Australia at all.

Technology organizations looking to open offices in a new country could decide that "AEST" (Australian Eastern Standard Time) isn’t such a great time zone after all. As remote work continues to become more prevalent, will companies start saying “goodbye” instead of “g’day” to applicants from Australia [who are unable to sufficiently secure and encrypt their corporate communications]?

This doesn’t seem like smart politics, but nothing about this bill seems particularly smart.

We remain committed to fighting mass surveillance worldwide. We encourage users in Australia to reach out to their representatives and express their opposition to the Assistance and Access Bill.

So... It's been five years since Edward Snowden's summer of 2013 revelations of classified NSA documents which apparently and convincingly detailed the breadth and depth of the NSA's technological surveillance efforts and capabilities.

Five years. During that short time, the world, our world, has changed rather dramatically. It's almost quaint to imagine using Firesheep to grab unprotected browser session cookies in a coffee shop to impersonate the logged-on user. We've grown up a lot. Edward didn't cause that to happen by himself... but he did probably cut the timing by several years. Perhaps in half.

Now, here we are, five years later. Enough incidents like the San Bernardino terrorist shooting, where the FBI was initially locked out of Syed Farook’s iPhone and Apple refused to help... and governments are finally beginning to stir.

As this 2018 year winds down to its end, I suspect that 2019 and 2020 are going to be the years where the you-know-what hits the fan and we see a collision between tech companies with their encryption and global governments. We live in interesting times.

Believe it or not, we have not escaped 2018 without a second mention of "PewDiePie"
Two recap for our listeners (and for Leo who was spared our first coverage of this due to his trip back East)...

A big fan of YouTube's sensation and top subscription magnet "PewDiePie", whose Twitter handle is @HackerGiraffe, was concerned that PewDiePie might be about to slip into 2nd place, eclipsed by T-Shirt, I mean, "T-Series"...

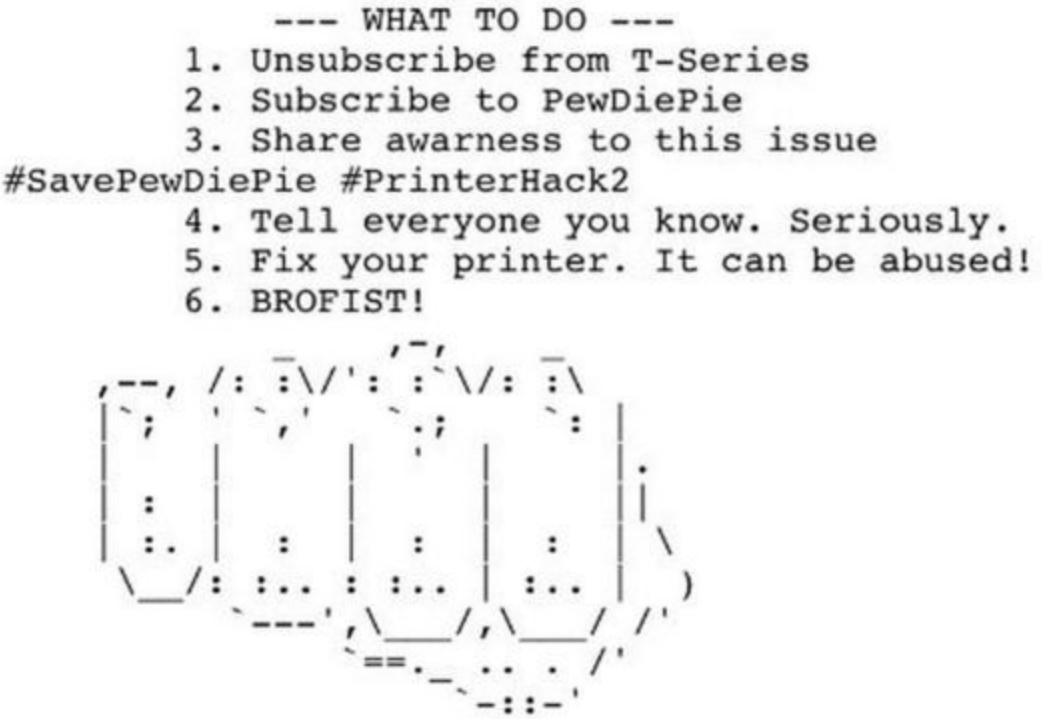
So @HackerGiraffe zipped over to Shoran to get the IPs of some exposed printers. He reported finding around 800,000 printers apparently exposed, of which he grabbed 50,000 IPs. He then used PRET -- the PRinter Exploitation Toolkit -- on Github – which gives hackers the ability to access files, damage the printer, or access the internal network.

And he then sent out a plea for recipients of his illicitly printed page to please subscribe to PewDiePie and, while you're at it, please unsubscribe from T-Series.

We're talking about this again, two podcasts later, @HackerGiraffe is back at it again.

And he's claiming, though it hasn't been verified and it's unclear how it might be, that this one doubled the Shodan-derived printer inventory to 100,000 printers.

@HackerGiraffe tweeted that he'll be taking a university exam at around the time, Monday morning, that employees will be arriving at work to potentially find the printouts.



"PewDiePie is in trouble and he needs your help to defeat T-Series!

--- WHAT IS GOING ON ---

PewDiePie, the currently most subscribed to channel on YouTube, is at stake of losing his position as the number one position by an Indian company called T-Series that simply uploads videos of Bollywood trailers and campaigns.

--- WHAT TO DO ---

1. Unsubscribe from T-Series.
2. Subscribe to PewDiePie.
3. Share awarness [sic] of this issue.
   #SavePewDiePie #PrinterHack2
4. Tell everyone you know. Seriously.
5. Fix your printer. It can be abused.
6. BROFIST!

Speaking to the BBC anonymously online, HackerGiraffe said: "I've been trying to show that 'hacking' isn't a game or toy, it can have serious real-life consequences. We really want people to pay attention to this because causing physical damage is very much a possibility."

He explained that flaws in the printers' firmware meant that he could continuously force data to be written to their chips.

"These chips have a limited lifetime of 'writes'. If you keep the loop on enough, the chip will fry and the printer will no longer function."
(At the moment, PewDiePie has around 75 million subscribers to T-Series 70 million.)

And... as if all this wasn't enough, the Wall Street Journal website was also defaced to say: "Wall Street Journal would like to apologize to pewdiepie. Due to misrepresentation by our journalists, those of whom have now been fired, we are sponsoring pewdiepie to reach maximum subscribers and beat Tseries to 80 million."

Border agents are copying travelers' data... then leaving it on USB drives
I have absolutely zero contraband of any sort of my phone. But it's mine. It's private. As are my conversations with my friends and family. And I feel a bit proprietary toward them. Maybe I'm just being dumb. But the idea that someone unknown to me is going to rifle through my private messages and photos and browsing history, etc. is more than mildly annoying. And not because I have anything whatsoever to hide... just because it's mine.

So when I then read, as was in the news this week as a result of a report from the Office of Inspector General (OIG) published on the Department of Homeland Security (DHS) website last week... that our own US Customs and Border Protection agents are retaining the data that they copy from user's devices, that's even more annoying.

We've touched on this from time to time, but for the past three years, since 2015 when the Trade Facilitation and Trade Enforcement Act of 2015 (TFTEA) went into effect, CBP agents are allowed to carry out warrantless device searches at all 328 ports of entry in the US.

This means that agents are allowed to manually (visually) inspect any travelers' devices, such as smartphones or laptops, without a reason, and they're supposed to look for suspicious content related to terrorism, child pornography, or anything that might hint of a crime.

And, moreover, in 67 selected ports, CBP agents are also allowed to copy device data onto a USB thumb drive and uploaded it onto a search platform called an Automated Targeting System (ATS) on which more complex searches are carried out against the user's copied data.

According to the recently published OIG report, CBP agents have not been deleting user data from these USB thumb drives after they've loaded the data onto the ATS, as standard procedure dictates. The OIG report said: "We physically inspected thumb drives at five ports of entry. At three of the five ports, we found thumb drives that contained information copied from past advanced searches [which had not been deleted]."

This, of course, means CBP agents could still access the user's possibly-sensitive information even after he or she had been released to enter the US. The data was permanently exfiltrated.

The OIG recommended that the CBP OFO improve its procedures and train agents accordingly.

What this means for me, since it would not be any inconvenience for me, and we've talked about this before, is that I'll travel with a burner iPhone 6 that I wipe and reset when crossing borders. If I'm questioned about why the phone is blank I'll just explain that this is my travel phone for emergencies, and that I leave my main good phone at home so it's not a risk of loss or theft while travelling to strange lands.

This Week in Android Hijinks:
https://www.welivesecurity.com/2018/12/11/android-trojan-steals-money-paypal-accounts-2fa/
ESET's posting is titled: "Android Trojan steals money from PayPal accounts even with 2FA on"

ESET researchers discovered a new Android Trojan using a novel Accessibility-abusing technique that targets the official PayPal app, and is capable of bypassing PayPal’s two-factor authentication

There is a new Trojan preying on Android users, and it has some nasty tricks up its sleeve.

First detected by ESET in November 2018, the malware combines the capabilities of a remotely controlled banking Trojan with a novel misuse of Android Accessibility services, to target users of the official PayPal app.

At the time of writing, the malware is masquerading as a battery optimization tool, and is distributed via third-party app stores.

"Optimization Android" v1.0

After being launched, the malicious app terminates without offering any functionality and hides its icon. So it really has become a Trojan. From then on, it watches what the user does... with the primary goal of stealing money from the victim's PayPal accounts. This requires the activation of a malicious Accessibility service. This request is presented to the user as being from the innocuous-sounding “Enable statistics” service:

Use Enable statistics?
Enable statistics needs to:
o Observe your actions / Retrieve notifications when you're interacting with an app.
o Retrieve window content / Inspect the content of a window you're interacting with.
[CANCEL] [OK]

The user now has a Trojan on their machine which is able to intercept any launch of PayPal to maliciously transfer 1000 Euro or other local currency whenever the user logs into PayPal. The use of two factor authentication doesn't prevent this since the Trojan simply waits for the 2FA to be complete and the logon to completely succeed. Then, using malicious accessibility features, the Trojan executes hidden PayPal transactions as if they were the user without the knowing. The entire process requires just seconds.

ESET has notified PayPal of the existence of the Trojan and of the PayPal account to which funds are being transferred. The hack will fail only if the user has no balance and no card connected to their PayPal account.

Our takeaway from this, I think, is just a reminder that our Smartphones have become very smart. And we've seen multiple instances on both iOS and Android platforms where the addition of important features, such as those required in support of accessibility, can be abused in clever ways that was never their intent.

Safe use of desktop PCs requires some care about what apps are loaded from where -- such that both Microsoft and Apple are starting to create controlled "Stores" for their desktop platforms. Asking users to be constantly vigilant in their actions is easy to say. But if an app is doing things behind the user's back secretly, the only place where vigilance can be effective is preventing the app from getting into the device in the first place.

The Wordpress organization recently released their major v5 of Wordpress.
And as they have done since their v1.0 "Miles Davis" release on January 3rd, 2004, this one was also named after a famous musician...

1.0	Miles Davis	January 3, 2004
2.0	Duke Ellington	December 31, 2005
2.1	Ella Fitzgerald	January 22, 2007
2.2	Stan Getz	May 16, 2007
2.7	John Coltrane	December 10, 2008
3.2	George Gershwin	July 4, 2011
3.7	Count Basie	October 24, 2013
3.8	Charlie Parker	December 12, 2013
4.0	Benny Goodman	September 4, 2014
4.3	Billie Holiday	August 18, 2015
4.7	Sarah Vaughan	December 6, 2016

And with v5.0 we have...

5.0	Bebo Valdés	December 6, 2018

Of whom Wordpress say: This release is named in homage to the pioneering Cuban jazz musician Bebo Valdés.

I mention this because the tech press, always searching for a click-inducing headline, picked up on the most worrisome-sounding worst-case scenario, which was also very unlikely to occur, stating in their headlines that one of the bugs "might allow Google to index user passwords."

https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/

So on December 13th, one week after the December 6th release of v5.0, Wordpress released v5.0.1 as a security patch to fix seven recently identified problems... but, significantly, these are NOT directly related to the v5.0 release but date as far back as the "Count Basie" release v3.7 in October of 2013.

After studying the various problems and standing back a bit, what really appears to have happened is that Wordpress stuck to their major v5.0 release schedule... and SEPARATELY released, coincidentally, a closely-timed security update for all releases for the past five years, which now included v5.

So these were fixes for some longstanding problems. After enumerating the seven problems found, in their disclosure Wordpress wrote: "Thank you to all of the reporters for privately disclosing the vulnerabilities, which gave us time to fix them before WordPress sites could be attacked."

The one of the seven that triggered the headlines was a discovery by "Team Yoast" that the user activation screen could be indexed by search engines in some uncommon configurations, leading to exposure of email addresses, and in some rare cases, default generated passwords.

I wanted to clarify this, however, so that Wordpress users of ANY release in the last five years will update their installations with the appropriate patches for the releases of Wordpress that they are running. This was NOT a v5-created problem. Wordpress wrote: "WordPress versions 5.0 and earlier are affected by the following bugs, which are fixed in version 5.0.1. Updated versions of WordPress 4.9 and older [back to v3.7] releases are also available, for users who have not yet updated to 5.0."

For anyone who's interested in v5.0, the big news there is a brand new extremely flexiblee block-based editor.

Facebook Photo Bug

Last Friday Facebook announced another security incident affecting millions of its customers. For the 12-day perdiod from September 13 to September 25, 2018, a buf existed in one of its APIs which exposed the private photos of nearly 6.8 million users. Their forensic analysis of the bug's impact revealed that up to 1,500 apps built by 876 developers could have had accessed the private photos of these 6.8 specific million users.

https://developers.facebook.com/blog/post/2018/12/14/notifying-our-developer-ecosystem-about-a-photo-api-bug/

In Facebook's Tomer Bar's posting about this last Friday under the title: "Notifying our Developer Ecosystem about a Photo API Bug", he said:

"Our internal team discovered a photo API bug that may have affected people who used Facebook Login and granted permission to third-party apps to access their photos. We have fixed the issue but, because of this bug, some third-party apps may have had access to a broader set of photos than usual for 12 days between September 13 to September 25, 2018.

[As we know, the title of last week's podcast was "Internal Bug Discovery." So I'll note that, despite Facebook's current residence being a doghouse, they, like Google, clearly have an internal "Red Team" of some sort whose job it is to find their own mistakes before anyone else does. Rather than faulting them for making a mistake, I salute them, as I did Google, for finding and fixing it themselves. This is not a role anyone wants to outsource to hackers.]

Anyway, Tomer continued to explain: "When someone gives permission for an app to access their photos on Facebook, we usually only grant the app access to photos people share on their timeline. In this case, the bug potentially gave developers access to other photos, such as those shared on Marketplace or Facebook Stories. The bug also impacted photos that people uploaded to Facebook but chose not to post. For example, if someone uploads a photo to Facebook but doesn't finish posting it - maybe because they've lost reception or walked into a meeting - we store a copy of that photo for three days so the person has it when they come back to the app to complete their post.

Currently, we believe this may have affected up to 6.8 million users and up to 1,500 apps built by 876 developers. The only apps affected by this bug were ones that Facebook approved to access the photos API and that individuals had authorized to access their photos.

We're sorry this happened. Early next week we will be rolling out tools for app developers that will allow them to determine which people using their app might be impacted by this bug. We will be working with those developers to delete the photos from impacted users.

We will also notify the people potentially impacted by this bug via an alert on Facebook. The notification will direct them to a Help Center link where they'll be able to see if they've used any apps that were affected by the bug. (See example of user notification below)

We are also recommending people log into any apps with which they have shared their Facebook photos to check which photos they have access to.

We are building ever-more-complex and capable systems. And these systems will be imperfect. I fully expect that someday we will figure out how to economically make these systems far more "correct." But until we get there, the best we can do is find and fix their problems as they arise.

Oh..... the passwords we use...
"SplashData" the publisher of the password management applications TeamsID, Gpass, and SplashID, has just released their eighth annual list of "Worst Passwords of the Year."

Appearing for the first time on the list, in 23rd place, is "donald" and, somewhat sadly, our long time podcast favorite "monkey" has slipped five rungs down to 18th place... while "princess" is debuting in the 11th slot.

So what do we have? The top two have not budged...

1 123456 (Rank unchanged from last year)
2 password (Unchanged)

Then we have the slightly less popular, yet growing in popularity for those less lazy:

3 123456789 (Up 3)

while "12345678" drops down one rung to 4th place.

4 12345678 (Down 1)

"12345" with just five digits retains its 5th place position.

5 12345 (Unchanged)

6 111111 (New)
7 1234567 (Up 1)
8 sunshine (New)
9 qwerty (Down 5)
10 iloveyou (Unchanged)

11 princess (New)
12 admin (Down 1)
13 welcome (Down 1)
14 666666 (New)
15 abc123 (Unchanged)
16 football (Down 7)
17 123123 (Unchanged)
18 monkey (Down 5)
19 654321 (New)
20 !@#$%^&* (New)
21 charlie (New)
22 aa123456 (New)
23 donald (New)
24 password1 (New)
25 qwerty123 (New)

SplashData estimates almost 10% of people have used at least one of the 25 worst passwords on this year's list, and nearly 3% of people have used the worst password, 123456.

According to SplashData, the over five million leaked passwords evaluated for the 2018 list were mostly held by users in North America and Western Europe.

And we should note that passwords leaked from hacks of adult websites were not included in this report... doubtless being NSFW.

After being hit by drive encrypting malware, all might not be lost
Just because SOME malicious drive encryption is done right, doesn't mean that all of it has all been done right. Far from it, apparently.

This past week Bleeping Computer covered a number of instances where, indeed, all is not lost... and in fact nothing need be lost...
"How to Decrypt the InsaneCrypt or Everbe 1 Family of Ransomware"
https://www.bleepingcomputer.com/ransomware/decryptor/how-to-decrypt-the-insanecrypt-or-everbe-1-family-of-ransomware/

InsaneCrypt or the Everbe 1.0 Ransomware is a family of ransomware infections that were based off of an open source project. This ransomware family is distributed through possibly spam and hacking into Remote Desktop Services, but it has not been confirmed.

The good news is that the variants of this ransomware family can be decrypted for free using a decryptor created by Michael Gillespie and Maxime Meignan. In order to use the decryptor a victim just needs to have an encrypted file and an unencrypted version of the same file. This can typically be achieved through the sample pictures provided by Windows.

"How to Decrypt HiddenTear Ransomware Variants"
https://www.bleepingcomputer.com/ransomware/decryptor/how-to-decrypt-hiddentear-ransomware-variants/

If you have been infected with a HiddenTear Ransomware variant, then you are in luck as a program called HiddenTearDecrypter has been created by Michael Gillespie that allows you recover your encryption key without having to pay the ransom.

HiddenTear is the name of a ransomware family whose full source code was published on GitHub. This allowed attackers to download the source code and create their own ransomware variants that could be used to infect victims.

Due to the source code's wide availability, there are many ransomware infections under different names that utilize the same HiddenTear code base. As the original code was decryptable, this means all other ransomware created from the same code are decryptable as well.

Some of the HiddenTear variants supported by this tool include:

8lock8, AnonCrack, Assembly, Balbaz, BankAccountSummary, Bansomqare Wanna, Blank, BloodJaws, Boris, CerberTear, CryptConsole2, CryptoKill, CyberResearcher, Data_Locker, Dev-Nightmare 2xx9, Diamond, Domino, Donut, dotRansom, Executioner, Executioner2, Executioner3, Explerer, FlatChestWare, Frog, Fuck_You, Gendarmerie, Horros, JobCrypter, Jodis, J-Ransomware, J-Want-To-Cry, Karmen, Kraken 2.0, Kratos, LanRan, Lime, Lime-HT, Luv, Matroska, MireWare, MoonCrypter, MTC, Nobug, Nulltica, onion3cry, OpsVenezuela, Paul, PayOrDie, Pedo, PGPSnippet, Poolezoor, Pransomware, Predator, Qwerty, Random6, Random6 2, Randion, RansomMine, Rootabx, Saramat, Shrug, ShutUpAndDance, Sorry, Symbiom, TearDr0p, Technicy, The Brotherhood, TheZone, tlar, TotalWipeOut, TQV, Ton, VideoBelle, WhiteRose, WhiteRose2, Zalupaid, ZenCrypt, Zenis, ZeroRansom, Zorro

If you are infected with a HiddenTear Ransomware infection, you can use the following guide to decrypt your files for free.

So, our takeaway here is, many wannabe attackers started with the source code of a readily decryptable malware... and have used it widely. So all is not lost if you or a friend or family member might be hit by "Donut", "PayOrDie" or "Sorry".

Microsoft is encountering a run of 0-Day vulnerabilities.
Last week's Patch Tuesday closed a 0-day vulnerability that was being actively exploited by two different state-sponsored cyber-espionage groups who were also behind the 0-Day that Microsoft patched the month before, in November.

And before that, in October, Kaspersky Lab discovered a zero-day (CVE-2018-8453) being used for elevation of privilege by the FruityArmor state-sponsored cyber-espionage group.

And before that, in September, Microsoft patched a zero-day (CVE-2018-8440) that was in use by hackers who were using it to install and spread a backdoor.

Mostly, these appear to be high-end, highly targeted, state-level espionage attacks. The discoverers of these 0-Day vulnerabilities don't want to have them discovered since they are too useful for their targeted attacks. So they are not being used in widespread campaigns.

Still... the spate of 0-Days affecting this increasingly old and creaky operating system is a bit worrying. It is coming under increasing pressure and it's not holding up as well as we would like or hope. Listening to Paul and MaryJo on Wednesday is quite sobering. So it's not just me!

Firefox 64 Released

An interesting note is that Firefox's full distrust of Symantec SSL/TLS certificates was postponed from the previous release 63 until this release 64 because Mozilla's telemetry had indicated that too many of its users were still encountering websites that were still using Symantec certs. (Just re-up with DigiCert, people!)

Bleeping Computer provides a nice summary of what's new:

- Contextual Feature Recommender (CFR)
  Contextual Feature Recommender is a new feature that will display recommended extensions that correspond to a particular site you are visiting. For example, if you are visiting Facebook it will recommend a Facebook extension. At this time, Mozilla is recommending three extensions: Facebook Container, Enhancer for YouTube and To Google Translate.

  Fortunately, users not wanting these recommendations can disable them by going to the Firefox Options, scrolling down to the Browsing category, and unchecking the checkbox labeled "Recommend extensions as you browse".

- Multi-Tab Selection
  Multiple tabs can now be selected by pressing shift or ctrl+click. Once chosen, they can be all be bookmarked, moved, or pinned as a group.

- Native Windows 10 Share Support
  Firefox 64 has added native Windows 10 sharing support to allow sharing a web page using a variety of applications installed in Windows 10. To access this feature, click on the [...] menu in the address bar, then click on the Share button. This will open the native Windows 10 sharing dialog.

- Live bookmarks and RSS Feed subscriptions removed
  Mozilla has removed support for Atom and RSS feed subscriptions, including Live Bookmarks in this release. So fro mnow on, users wishing to subscribe to RSS feeds will need to install an addon to provide that feature.

- Task Manager now shows you energy usage per tab
  The Firefox Task Manager has been improved to show you how much energy a particular tab, addon, or other task is using. This is especially handy for spotting pages that are using in-browser mining... or just behaving badly. Goto: "about:performance" in the URL bar.

- Remove Extension context menu option
  It's now possible to right-click on the icon for any installed extension and select "Remove Extension" to remove the extension.

Miscellany

"Rails"

SpinRite

Kevin Sears / Eagle, WI
Subject: Holiday Hard Drive Song

A winter holiday remake... (the the tune "Let It Snow")

I Let It Go
by Kevin Sears (aka macdoctorwho)

Oh, an aging hard drive is frightful,
But my photos are so delightful.
Got no plan you know so,
I let it go, let it go, let it go...

It doesn't show signs of stoppin’,
Cuz error correction is workin’.
The green light is still “ON” so,
I let it go, let it go, let it go...

The drive wants to say “goodnight”,
But everythin’ seems to be alright.
Bits are aged five years or more,
Seems slower than it did before.

The spinning surface is dyin’,
And, S.M.A.R.T. says its goodbyin’.
I think I backed up a year ago so,
I let it go, let it go, let it go...

Now bits are just ones or zeroes,
Trying hard to be our data hero.
But soon they'll be unreadable…NO!
Cuz I let it go, let it go, let it…….

Closing The Loop

Mike S in New Hampshire
Subject: Router infection

I have a Netgear router (R6300v2 if you care)
I routinely clicked on the Router Update -> Check for new version link. It continued to tell me that there were no updates available.
Last week I started seeing very slow response from WiFi, so I checked again, and it said the same thing. I decided to check with Netgear, and their website had updates! I downloaded the update and applied it, and the performance went back to normal. So far, it's been good.
I'm thinking that some malware got into it, and wouldn't let it check for updates itself, but it couldn't stop me from finding updates on the Netgear website.
Thanks for a great podcast!

Chris Petersen / Ham Lake, Minnesota
Subject: It's even simpler

Regarding the unauthenticated UDP IoT protocol. Since it's IoT, we want to minimize processing and storage.

The IoT device only needs to return a pseudo-random number to the requestor. This can be of poor entropy. The IoT device is then paired with the requestor. Any further requests from that IP must include the same PRN or they are rejected. A request from another IP resets the "pairing" and the process is repeated.

We can do this because spoofing requires a response to a fake IP address. This fake address, the victim, won't be responding.

A short PRN of 32 bits or less should be sufficient, thus minimizing traffic sent to the victim.

Obviously not "perfect" security. But it's "good enough" security - and way better than what they're doing now.

73,
Chris, K9EQ

Critical SQLite Flaw Leaves Millions of Apps Vulnerable to Hackers

https://blade.tencent.com/magellan/index_en.html

This is bad:

<quote> Magellan is a remote code execution vulnerability discovered by Tencent Blade Team that exists in SQLite. As a well-known database, SQLite is widely used in all modern mainstream operating systems and software, so this vulnerability has a wide range of influence. After testing Chromium was also affected by this vulnerability, Google has confirmed and fixed this vulnerability. We will not disclose any details of the vulnerability at this time, and we are pushing other vendors to fix this vulnerability as soon as possible.

Q & A

(1) Am I affected by the vulnerability?
If you use a device or software that uses SQLite or Chromium. It may be affected, depending on whether there is a suitable attack surface.

[My eMail client, Thunderbird, uses SQLite... and eMail has a notorious attack surface! It's pretty much ALL attack surface!]

(2) What is the danger of this vulnerability?
Remote code execution, leaking program memory or causing program crashes.

(3) Does this vulnerability have exploit code?
Yes, we successfully exploited Google Home with this vulnerability, and we currently have no plans to disclose exploit code.

(4) What are the conditions for exploiting the vulnerability?
This vulnerability can be triggered remotely, such as accessing a particular web page in a browser,Or any scenario that can execute SQL statements.

(5) Has "Magellan" been abused in the wild?
We have not seen the case yet.

(6) Is there a workaround/fix?
We have reported all the details of the vulnerability to Google and they have fixed the vulnerability (commit). If your product uses Chromium, please update to the official stable version 71.0.3578.80(Release updates). If your product uses SQLite, please update to 3.26.0 (Release updates).The CVE number is pending.

My: Google Chrome is up to date: Version 71.0.3578.98 (Official Build) (64-bit)

December 14th: "Crash Chrome 70 with the SQLite Magellan bug"

https://worthdoingbadly.com/sqlitebug/

First of all... what is SQLite? ...

ZDNet:

A security vulnerability in the massively popular SQLite database engine puts thousands of desktop and mobile applications at risk.

[
I found it in four places on my system:
Networx: sqlite.dll 11/17/2026
Python: sqlite3.dll v3.21.0.0 / 6/27/2018
Abode Reader: sqlite.dll v1.0.0.1 2/27/2009
NovaPDF: sqlite.interop.dll v1.0.82.0 6/8/2018
]

Discovered by Tencent's Blade security team, the vulnerability allows an attacker to run malicious code on the victim's computer, and in less dangerous situations, leak program memory or cause program crashes.

Because SQLite is embedded in thousands of apps, the vulnerability impacts a wide range of software, from IoT devices to desktop software, and from web browsers to Android and iOS apps.

The bad news, according to Tencent Blade researchers, is that this vulnerability can also be exploited remotely by accessing something as simple as a web page, if the underlying browser support SQLite and the Web SQL API that translates the exploit code into regular SQL syntax.

Firefox and Edge don't support this API, but the Chromium open-source browser engine does. This means that Chromium-based browsers like Google Chrome, Vivaldi, Opera, and Brave, are all affected. A demo that crashes a Chrome tab is available here.

But while web browsers pose the biggest attack surface, other apps are also affected. For example, Google Home is also vulnerable.

"We successfully exploited Google Home with this vulnerability," the Tencent Blade team said in a security advisory this week.

Tencent Blade researchers said they reported this issue to the SQLite team earlier this fall. A fix was shipped out on December 1, with the release of SQLite 3.26.0. The fix was also ported inside Chromium, and later in Google Chrome 71, released last week.

Chromium-based browsers like Vivaldi and Brave are running the latest version of Chromium, but Opera is still one Chromium release behind, meaning it's latest release is still affected.

While it does not support Web SQL, Firefox, too, is affected, since it comes with a locally accessible SQLite database, meaning a local attacker could abuse this vulnerability to execute

code and more.

Eyal Itkin, a Check Point researcher, also pointed out that the vulnerability also requires an attacker having "the ability to issue arbitrary SQL commands so to corrupt the DB and trigger the vulnerability," which greatly reduces the number of vulnerable applications.

But even if the SQLite team shipped a fix, many apps are likely to remain vulnerable for years to come. Updating the underlying database engine to any desktop, mobile, or web app is a dangerous process, which sometimes can result in data corruption, and most programmers avoid it as long as possible.

App developers rarely update libraries and the component parts of their apps as it is, so the chances that this vulnerability will haunt the app ecosystem for years is pretty high.

Because of this reason, the Tencent Blade team said it would refrain for the time being from releasing any proof-of-concept exploit code. Nonetheless, other security researchers have already started combing the SQLite patch to reverse engineer it and see how the vulnerability works under the hood.

This SQLite vulnerability has not yet received a CVE identification number and Tencent researchers are using the "Magellan" codename to refer to it for now.

---

cd /
dir /s sqlite*.dll

Networx: sqlite.dll 11/17/2026
Python: sqlite3.dll v3.21.0.0 / 6/27/2018
Abode Reader: sqlite.dll v1.0.0.1 2/27/2009
NovaPDF: sqlite.interop.dll v1.0.82.0 6/8/2018

~30~