Security Now! #595 - 01-17-17

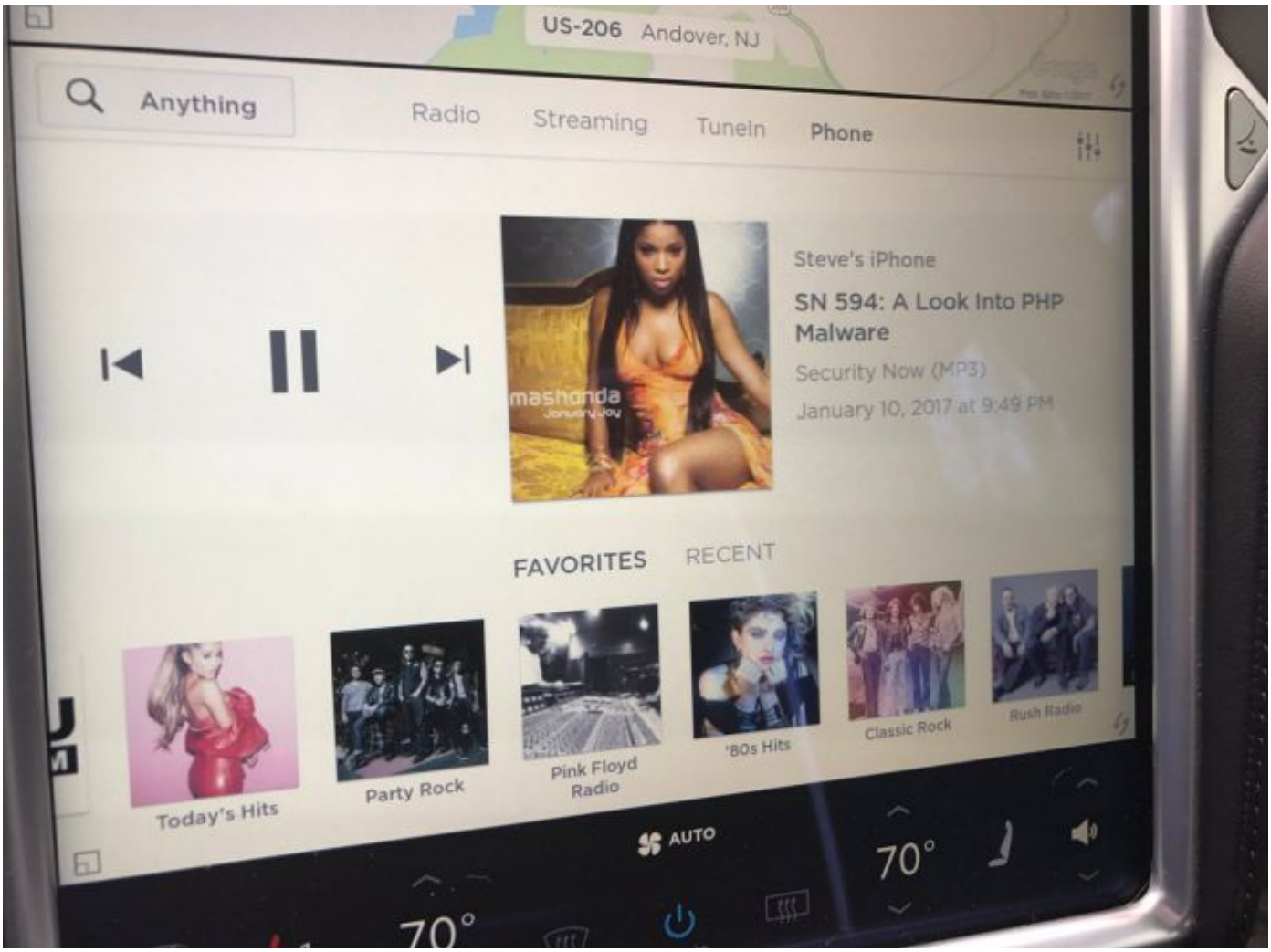
What's up with WhatsApp?

This week on Security Now!

A classic bug at GoDaddy bypassed domain validation for 8850 issued certificates, could flashing a peace sign compromise your biometric data?, it's not only new IoT devices that may tattle, many autos have been able to for the past 15 years, McDonalds get caught in a web security bypass, more famous hackers have been hacked, Google uses AI to increase image resolution, more on the value or danger of password tricks, and... does WhatsApp incorporate a deliberate crypto backdoor?

Steven Minnick (@stevenminnick) / 1/12/17, 6:49 AM

@SGgrc Steve, you look fantastic! Best album art from @TeslaMotors for your podcast so far.



Security News

The Righteous Bug of the New Year bites GoDaddy, and allows the issuance of 8850 certificates without proper domain validation.

- First note: GoDaddy handled the entire mess perfectly and responsibly.
  (But it's still wonderful!) (Unlike like "Wo-be-gone" -- WoSign)

- Google Groups: Mozilla.dev.security.policy
  - Wayne Thayer, Senior Internet Product & Technology Leader at GoDaddy
  - https://groups.google.com/forum/m/?hl=en#!msg/mozilla.dev.security.policy/Htujoyq-pO8/uRBcS2TmBQAJ

- January 10th: Incident Report – "Certificates issued without proper domain validation"

  On Friday, January 6th, 2017, GoDaddy became aware of a bug affecting our domain validation processing system. The bug that caused the issue was fixed late Friday [the same day].

  At 10 PM PST on Monday, Jan 9th we completed our review to determine the scope of the problem, and identified 8850 certificates that were issued without proper domain validation as a result of the bug. The impacted certificates will be revoked by 10 PM PST on Tuesday, Jan 10th, and will also be logged to the Google Pilot CT log. [Certificate Transparency]

- Detailed Description:
  On Tuesday, Jan 3rd, 2017, one of our resellers (Microsoft) sent an email to a GoDaddy notification account and two GoDaddy employees. Due to holiday vacations and the fact that the issue was not reported properly per our CPS, we did not become aware of the issue until one of the employees opened the email on Friday Jan 6th and promptly alerted management. The issue was originally reported to Microsoft by one of their own customers and was described as only affecting certificate requests when the DNS A record of the domain was set to 127.0.0.1. An investigation was initiated immediately and within a few hours we determined that the problem was broader in scope. The root cause of the problem was fixed via a code change at approximately 10 PM MST on Friday, Jan 6th.

  On Saturday, January 7th, we determined that the bug was first introduced on July 29th, 2016 as part of a routine code change intended to improve our certificate issuance process. The bug is related to our use of practical demonstration of control to validate authority to receive a certificate for a given fully-qualified domain name. In the problematic case, we provide a random code to a customer and ask them to place it in a specific location on their website. Our system automatically checks for the presence of that code via an HTTP and/or HTTPS request to the website. If the code is found, the domain control check is completed successfully.

  Prior to the [introduction of the] bug, the library used to query the website and check for the code was configured to return a failure if the HTTP status code was not 200 (success).

A configuration change to the library caused it to return RESULTS even when the HTTP status code was not 200.

Since many web servers are configured to include the URL of the request in the body of a 404 (not found) response, and the URL also contained the random code, any web server configured this way caused domain control verification to complete successfully.

We are currently unaware of any malicious exploitation of this bug to procure a certificate for a domain that was not authorized. The customer who discovered the bug revoked the certificate they obtained, and subsequent certificates issued as the result of requests used for testing by Microsoft and GoDaddy have been revoked.

Further, any certificate requests made for domains we flag as high-risk were also subjected to manual review (rather than being issued purely based on an invalid domain authorization).

We have re-verified domain control on every certificate issued using this method of validation in the period from when the bug was introduced until it was fixed. A list of 8850 potentially unverified certificates (representing less than 2% of the total issued during the period) was compiled at 10 PM PST on Monday Jan 9th. As mentioned above, potentially impacted certificates will be revoked by 10 PM PST on Tuesday Jan 10th and logged to a Google CT log.

Additional code changes were deployed on Monday Jan 9th and Tuesday 10th to prevent the re-issuance of certificates using cached and potentially unverified domain validation information. However, prior to identifying and shutting down this path, an additional 101 certificates were reissued using such cached and potentially unverified domain validation information, resulting in an overall total of 8951 certificates that were issued without proper domain validation as a result of the bug.

- Next Steps:
  While we are confident that we have completely resolved the problem, we are watching our system closely to ensure that no more certificates are issued without proper domain validation, and we will take immediate action and report any further issues if found. A full post-mortem review of this incident will occur and steps will be taken to prevent a recurrence, including the addition of automated tests designed to detect this type of scenario. If more information about the cause or impact of this incident becomes available, we will publish updates to this report.

  Wayne Thayer
  GoDaddy

Could flashing a peace sign lead to biometric theft?

- Camera resolutions are skyrocketing.
- I am sometimes taken aback when I zoom well into a photo and the image remains super-sharp.
- I do underestimate the amount of detail we capture in a photo.
- We've seen demonstrations of distant photos of a key being turned into a working physical object.

Cartapping: How Feds Have Spied On Connected Cars For 15 Years

- http://www.forbes.com/sites/thomasbrewster/2017/01/15/police-spying-on-car-conversations-location-siriusxm-gm-chevrolet-toyota-privacy/#49e1c5bf649b
- Thomas Fox-Brewster, who covers crime, privacy and security in digital and physical forms for Forbes, writes…

  The rapid spread of connected devices that can listen and locate has been a boon for law enforcement. Any new technology hooked up to the web has the potential to become a surveillance device, even if it's original purpose was benign, as shown in a 2016 Arkansas murder investigation where Amazon was asked to hand over audio from a suspect's Echo.

  But such information and much more, I've learned, has long been retrievable from cars. Indeed, court documents reveal a 15-year history of what's been dubbed "cartapping," where almost real-time audio and location data can be retrieved when cops order vehicle tech providers to hand it over.
  One of the more recent examples can be found in a 2014 warrant that allowed New York

police to trace a vehicle by demanding the satellite radio and telematics provider SiriusXM provide location information. The warrant, originally filed in 2014 but only recently unsealed (and published below in full), asked SiriusXM "to activate and monitor as a tracking device the SIRIUS XM Satellite Radio installed on the Target Vehicle for a period of 10 days." The target was a Toyota 4-Runner wrapped up in an alleged illegal gambling enterprise.

SiriusXM told FORBES it complied with the order and did so by switching on the stolen vehicle recovery feature of its Connected Vehicle Services technology, which is only available in a subset of cars it supplies (the satellite radios alone cannot be tracked as the telematics services can). The request was, then, akin to the police demanding Apple hand over a customer's location data by turning on the Find My iPhone feature. The company said it also worked sporadically with law enforcement to provide such information, noting it always required a valid warrant, estimating it receives five valid court orders a year to activate the stolen vehicle recovery feature to monitor a suspect. It declined to offer on-record comment.

The SiriusXM case got me thinking: what other providers were being asked to track cars and in what manner? It was little surprise to find General Motors (GM) had repeatedly worked with cops to hand over not just location but also audio, where conversations were recorded when the in-car cellular connection was switched on; its OnStar service is one of the best-known telematics providers on the market.

- The article goes on to cite specific example after example -- going back 15 years -- which were obtained from court document searches and followed-up by interviews.

  While my own beloved auto is 16 years old, and has absolutely zero connected technology, I'm sure I'm going to outlive it. When I update I imagine that whatever I get will be laden with conveniences the likes of which I only see when I travel in other people's cars or travel by air and rent a car. But I'm not overly concerned because my life is pretty boring and I take responsibility for my decisions and actions.

  But... if your life is a bit more sketchy, you should be aware that driving around in a new auto is not unlike being inside of a mobile teleconferencing endpoint... with camera and video running. And.. I have a stealth vehicle you might be interested in! <g>

Stealing passwords from McDonald's users

- https://finnwea.com/blog/stealing-passwords-from-mcdonalds-users
- Title: "Reflected XSS through AngularJS sandbox bypass causes password exposure of McDonald users"

- The display from a search query reads:
  'n' Search results were found for "<--------- search term ---------->"
  Then it has the term filled-into a form field
  Do you want to search ""<--------- search term ---------->" in news section?

- Recall our discussion last week about the dangerous power of scripting:
    - PHP is an interpreter running on the server. So if an attacker can arrange to cause a PHP-enabled server to display code they supply, all bets are off.
    - Similarly, JavaScript is an interpreter running on the client... but IF an attacker can arrange to control the page's contents, since the script is running within the context of the web server's domain, the "same origin policy" protects allow the script freedom to do whatever it wishes within that domain.
- McDonald's website implementation contains a distressing number of poor design decisions (which the full article elaborates but which I won't bog us down with here), all which can be leveraged thanks to the "friendliness" of echoing back user-supplied search text -- without URL-escaping it so that it DISPLAYS but will not run.
- And an even better policy, since mistakes can happen, is to avoid showing users what they provided. Web based online forums must do that by definition... but due to the well-known extreme danger of doing that, the danger of making a mistake is always in the forefront of the developer's mind.

Hacker Steals 900 GB of Cellebrite Data

- http://motherboard.vice.com/read/hacker-steals-900-gb-of-cellebrite-data
- Motherboard:
  Motherboard has obtained 900 GB of data related to Cellebrite [they were directly contacted by the hacker who extracted the data from Cellebrite], one of the most popular companies in the mobile phone hacking industry. The cache includes customer information, databases, and a vast amount of technical data regarding Cellebrite's products.

  The breach is the latest chapter in a growing trend of hackers taking matters into their own hands, and stealing information from companies that specialize in surveillance or hacking technologies.

  Cellebrite is an Israeli company whose main product, a typically laptop-sized device called the Universal Forensic Extraction Device (UFED), can rip data from thousands of different models of mobile phones. That data can include SMS messages, emails, call logs, and much more, as long as the UFED user is in physical possession of the phone.

  Cellebrite is popular with US federal and state law enforcement, and, according to the hacked data, possibly also with authoritarian regimes such as Russia, the United Arab Emirates, and Turkey.

  The data appears to have been taken, at least in part, from servers related to Cellebrite's website. The cache includes alleged usernames and passwords for logging into Cellebrite databases connected to the company's my.cellebrite domain. This section of the site is used by customers to, among other things, access new software versions.

Motherboard verified the email addresses in the cache by attempting to create accounts on Cellebrite's customer login portal. In the majority of cases, this was not possible because the email address was already in use. A customer included in the data confirmed some of their details.

The dump also contains what appears to be evidence files from seized mobile phones, and logs from Cellebrite devices.

Google's "RAISR": Rapid and Accurate Image Super Resolution

- https://arxiv.org/abs/1606.01299
- ABSTRACT:
  Given an image, we wish to produce an image of larger size with significantly more pixels and higher image quality. This is generally known as the Single Image Super-Resolution (SISR) problem. The idea is that with sufficient training data (corresponding pairs of low and high resolution images) we can learn a set of filters (i.e. a mapping) that when applied to a given image that is not part of the training set, will produce a higher resolution version of it, where the learning is preferably low complexity.

  In our proposed approach, the run-time is more than one to two orders of magnitude faster than the best competing methods currently available, while producing results comparable or better than state-of-the-art.

  A closely related topic is image sharpening and contrast enhancement, i.e., improving the visual quality of a blurry image by amplifying the underlying details (a wide range of frequencies).

  Our approach additionally includes an extremely efficient way to produce an image that is significantly sharper than the input blurry one, without introducing artifacts such as halos and noise amplification. We illustrate how this effective sharpening algorithm, in addition to being of independent interest, can be used as a pre-processing step to induce the learning of more effective upscaling filters with built-in sharpening and contrast enhancement effect.

Rodrigo Barrouin / 1/15/17, 4:59 AM

You said that a random character password was better than one with random words. Is this true of the diceware system even if longer?

If there exists ANY strategy for, in any way, short-circuiting a full brute force attack, then, while the system may still offer ***sufficient*** security, it does **not** offer maximum possible security. Even my own "Password Haystacks" method has been validly attacked by those who said that "haystack patterns" could be checked for... and those people are absolutely correct. Checking for "haystack patterns" would be a strategy that would be better than sheer, blind, brute force.

ONLY IF every character composing a string's password is equally probable, randomly chosen and without ANY interdependence among or between characters, do we have both maximum entropy for the length, and, by definition, no possible strategy for reducing the search space below that required for a full brute force attack.

Miscellany

- "Courtesy" of WikiLeaks:
  From:eryn.sepp@gmail.com
  To: john.podesta@gmail.com
  Date: 2015-02-19 00:35
- Subject: 2 things

  Though CAP is still having issues with my email and computer, yours is good to go.

  jpodesta
  p@ssw0rd

SpinRite

Tim D (near) Detroit Michigan
Subject: SN 594 SpinRite story (about the RAID 10 that was having problems until SpinRite)
:
Just wanted to note something I learned from an older and wiser geek than I;

He told me that about half of the very substantial price difference between a commercial (Compaq at the time) server and something with similar specs that you built out with consumer parts was that Compaq has a large supply of hard drives that are not managed in a first-in first-out manner.

Instead, your server would be assembled so that you didn't have two drives from the same manufacturing run in your array, because infant mortality in hard drives would tend to cluster around certain manufacturing runs. If one drive failed early and all drives were from the same manufacturing run, it is very likely that a second drive would fail before a new drive could be swapped in and the array rebuilt.

This is the reason I have always purchased drives for my Drobos one at a time.

What's up with WhatsApp?

Someday, hopefully, we'll have a widely recognized single understanding of the term "Backdoor".

We don't have that yet.

The confusion arises, partly because the term it's good for baiting clicks. A "backdoor" is a scary term partly because it seems sinister and also partly because it's a brilliant term. Everyone can readily visualize a "backdoor". As we saw during the FBI's request for a "we're not asking for a backdoor, we want a golden key for the front door" debacle last year... the proper use of the term requires an understanding of the INTENT.

UC Berkeley security researcher, Tobias Boelter, waited nine months after informing Facebook and WhatsApp of what he believed to be a security vulnerability in WhatsApp's implementation of the Signal protocol. When he was ignored and obtained no satisfaction, he went public and the Guardian dropped the bomb using the heavily freighted term... Backdoor.

The trouble was, and is... it was neither a mistake nor a secret... it is a deliberate and carefully considered design decision. This doesn't mean that it's the correct decision... but it does definitely mean that it's not a "backdoor" in the usual way that we still somewhat softly understand the term.

A backdoor IS an unknown and secret password embedded into router firmware that allows anyone who knows the secret incantation to obtain unofficial and unauthorized access to the device. That's a backdoor. But figuring out how a deliberate design decision feature can be abused may not be good... but it's not a backdoor.

So... shame on The Guardian for succumbing to the temptation to use that term. It achieved its goal, but at some increment of reputation cost to them.

So what is this all about????

It's another classic instance of a tradeoff between true security and security transparency.
Users say they want encryption for privacy.
We know that privacy requires authentication to be meaningful.
But edge-cases arise that forces a decision about whether to involve the user.

WhatsApp processes 15 trillion messages per year through its messaging servers.

WhatsApp is an implementation of the Open Whisper Systems "Signal" protocol.
In the past we discussed the Signal protocol's operation in detail. SN #555 is titled "WhatsApp":

In its introduction I summarized the news of the week and concluded: “… and the result of my deep dive into the Open Whisper Systems "Signal" communications protocol that's finally been fully integrated into the world's #1 multiplatform messaging system, WhatsApp, along with two things that MUST be done to get true security.”

Performing real time online messaging where both endpoints are guaranteed to be online for the conversation -- such as with a VPN -- is an easier problem to solve securely because, as we've discussed before, we have robust protocols, such as Diffie Hellman Key Agreement, which allow the two ends to securely negotiate a new shared secret in full view with their traffic being monitored. So long as reliable authentication is tied into the negotiation, to prevent tampering up to and including a man-in-the-middle attack, the endpoints are free to generate shared encryption keys at the start of every dialog and even to regenerate them periodically for long-lived communications.

But that's effectively "synchronous" communications where both ends are dynamically interacting. The plot thickens a great deal when we ask for the same sorts of guarantees from an asynchronous communications system.

In this scenario we want to send a secure message to the other endpoint... even if it's not currently online.

Signal provides for this, and WhatsApp inherits that provision, by asking clients to pre-generate a substantial block (of 100) public keys which will be escrowed on the messaging server. These keys are then distributed on an as-needed basis to anyone who wishes to obtain a short-lived ephemeral public key for the other endpoint.

And, if endpoints are offline, the messaging server must and does retain all unsent messages, in encrypted form, on the server while they await delivery.

Now here's the problem: What happens when that other endpoint changes phones, or SIM cards (as is more frequently done outside of the US) or when the application is reinstalled... WHILE THERE ARE STILL PENDING MESSAGES encrypted for its previously keyed identity?

This is where Signal and WhatsApp made different decisions: Will the other end's changed key require all future senders of messages to that new user to be notified?... or will the system somehow arrange to handle it transparently FOR the endpoint users?

Moxie and Open Whisper Systems opted for the safest, zero-compromise policy.

Facebook and WhatsApp... did not.

When an existing user appears with a new, changed key, that user supplies the messaging server with another new block of 100 public keys. The messaging server sees that one or more users have sent messages which were never delivered, and which are therefore sitting on the server encrypted with the changed key's obsoleted key. So the messaging server sends a message to the various senders, instructing them to RESEND those undelivered messages using one of the newly updated public keys.

And therein lies the tradeoff:
In Open Whisper System's Signal, the original sender's device will NOT silently comply with a request from the messaging server to re-encrypt and retransmit unreceived messages under the original intended recipient's new key. Signal notifies its user first and requires permission to do that.

WhatsApp considered this carefully and worried that in places where SIM cards are casually changed and where devices may be changed, notifying all senders of pending unreceived messages to the person changing their device would be excessively burdensome. So they made a different decision in WhatsApp's implementation of Signal. ONLY IF the user can change the security default of "notify me when my contacts keys change" will ANY notification EVER be seen... and only then AFTER the fact, when WhatsApp has already re-encrypted and resent the undelivered and pending messages.

So... while it is certainly not a "Backdoor," Tobias was probably right, on balance, to shine a light on the issue... because this feature COULD indeed be used to monitor conversations. Tobias further alleges that a modification of the messaging server's protocol -- entirely at the server-side, thus requiring no change to any existing clients -- could prevent the "message delivered successfully" notification from being returned to the sender, thus allowing not just the last message, but a lengthy conversation, to be surreptitiously monitored.

What we're learning from stories of Amazon Echo and IoT-connected devices having their records subject to court-ordered searches is that modern law enforcement WILL indeed turn over every possible rock and attempt to obtain information from every nook and cranny available. Now they also know that the #1 most popular encrypted private messaging system in the world, by a long shot, CAN be coerced to spy upon its users. Who among us thinks that's not going to cause Facebook and WhatsApp a great deal of grief?

As we originally recommended here, Signal is a better choice than WhatsApp. But Threema remains the only tool I would choose if I actually needed to have maximally private conversations. Threema lacks all of those extra convenience bells and whistles... which is exactly the point. It makes YOU responsible for authenticating the person with whom you are communicating. And no matter how much fancy technology tries and promises to lift that burden from you, you abdicate and delegate that responsibility at your peril.

Links:
http://gizmodo.com/theres-no-security-backdoor-in-whatsapp-despite-report-1791158247
http://www.bbc.com/news/technology-38609854
https://9to5mac.com/2017/01/13/whatsapp-encryption-backdoor-vulnerability/
https://www.theguardian.com/technology/2017/jan/16/whatsapp-vulnerability-facebook

~30~