Security Now! #478 - 10-21-14

Poodle Bites

Link Tracking Warning!
This document was first authored in Google Docs, then Downloaded as a PDF. So, Google has thoughtfully (ha!) added "tracking" redirections to all of the links here. (I have no idea why, but that's Google.) If that bothers you, simply copy the text of the link into your browser's URL field.

This week on Security Now!

- The FBI making noises about asking Congress to impose mandatory backdoors.
- Yubikey & Google's authentication news.
- Yesterday Apple launched ApplePay.
- In Depth coverage of "Poodle"... the latest chink in connection security.
    - (And why it's all a bunch of nonsense.)

Security News:

James Comey grumbles about time for legislative 'Fixed' for Phone Encryption By Apple, Google

- http://www.huffingtonpost.com/2014/10/16/james-comey-phone-encryption_n_5996808.html
- Huffington Post, Thursday: <quote>
  Last thursday, speaking at the Brookings Institution in Washington in his first major policy speech since taking over the FBI 13 months ago... FBI Director James Comey called for "a regulatory or legislative fix" for technology companies' expanding use of encryption to protect user privacy, arguing that without such a fix, "homicide cases could be stalled, suspects could walk free, and child exploitation victims might not be identified or recovered."

  Comey said he understood the "justifiable surprise" many Americans felt after former National Security Agency contractor Edward Snowden's disclosures about mass government surveillance, but he contended that recent shifts by companies like Apple and Google to make data stored on cell phones inaccessible to law enforcement went too far.

  Comey said: "Perhaps it's time to suggest that the post-Snowden pendulum has swung too far in one direction -- in a direction of fear and mistrust."said Comey... "Justice may be denied because of a locked phone or an encrypted hard drive."

  Comey said the FBI was seeing "more and more cases" in which law enforcement officials believed there was significant evidence on a laptop or phone they couldn't access due to encryption. It's not clear, however, that any of the cases he specifically referenced -- from a murder in Louisiana to a hit-and-run homicide in California -- could not have been solved with a traditional warrant to cellular service providers.

Matthew Green (assistant crypto professor at Johns Hopkins): "Law enforcement has access to more data than they've ever had. As a society we're just finally trying to get back to a point where it's a little more in line with what law enforcement would have been able to get back in the '80s."

For background, Huffington Post notes: Snowden's revelations have provoked a crisis abroad for major U.S. tech companies, which could lose billions as foreign customers leery of American software and devices compromised by the NSA turn to other providers. Comey said that he was "not trying to jump on the companies," like Apple and Google, that implement encryption systems closed off to law enforcement and that he believed they were "responding to a marketing imperative."

Google adds Yubikey 2nd-factor authentication to Chrome and Google assets.

- Chrome knows how to interact with the special, new, $18 "Security Key"
    - The "Security Key" and both "Neo's" ($50 and $60) are FIDO U2F-capable.

Yubico:

- Hardware lineup:
    - Five devices:
        - "Standard", "Nano", "Neo", "Neo-n" & "Security Key"
        - https://www.yubico.com/products/yubikey-hardware/

- FIDO U2F Security Key - $18 on Amazon.
    - Dedicated U2F device, U2F protocol only
    - Chrome browser under Windows, Mac, Linux

- YubiKey NEO - $50 on Amazon
    - Multi-Protocol authenticator w/support for FIDO U2F, OTP, etc.
    - NFC interface for contactless communication with compatible devices.
    - CCID compliant USB token, includes "Secure Element" and JavaCard.
    - Windows, Mac. Linux, Firefox, Chrome, etc.

- YubiKey NEO-n - $60 on Amazon.
    - Tucks into and nearly disappears into a USB slot.

SpinRite:

A note of thanks for the recent site licenses.

Poodle Bites:

TLS to SSL Downgrade attack...

- Some legacy servers don't understand TLS client opening handshakes and fail the connect.
- So client were forced to retry without TLS, on the assumption that server was old.
- That allows ANY MITM attackers to force SSL even when a server knows how to TLS.

SSLv3 has a choice of RC4 or CBC (cipher block chaining)

- RC4 is a nice stream cipher, but it suffers from "cold start" problems in its PRNG.
- CBC, being block-cipher based, requires a multiple of block length.

SSL got encryption and authentication the wrong way around:

- Moxie Marlinspike: "If you have to perform ANY cryptographic operation before verifying the MAC on a message you've received, it will somehow inevitably lead to doom.
    - http://www.thoughtcrime.org/blog/the-cryptographic-doom-principle/
- The only -- ONLY -- thing that can be safely done upon receiving a message is to verify the message's MAC to assure that it has not been tampered with. NOTHING ELSE IS SAFE.
- But... SSL authenticates BEFORE encrypting.
- ... which means that it decrypts THEN verifies the MAC (message authentication code).
- The original SSL designers didn't see any problem with that.
    - (Subtle crypto mistakes are SO easy to make.)

SSL returns a padding error which is different from a MAC validation error.

- "Padding check" happens at decryption, before the MAC can be checked.
- During CBC *decryption* the second to the last CIPHERTEXT is XORed with the output of the cipher to form the plaintext. In other words... this allows the attacker to DELIBERATELY flip-the-bits in the final block of plaintext.
- Thus... by intercepting and changing single bytes of the ciphertext sent by the client to the server, it's possible to use the decryption padding check

The attack requires:

- The users client (their browser), which knows and includes the site's HTTP authentication or logon session cookie into every connection to the remote website, to conduct a coordinated repetitious flood of the remote server with thousands of identical repeated queries.
- The attacker, positioned somewhere in the connection, but unable to see into the client's encryped data stream will, nevertheless, make careful alterations to bytes of the outgoing opaque stream as they whiz past, and then monitor the server's response to the WAY the stream is broken.
- Because the cracking occurs on cipher block boundaries, as individual successive client data bytes are finally determined, the signals its success to the client which then changes the upstream padding to cause successive header bytes to fall across cipher block boundaries.

To pull off an attack... attacker must:

- Obtain man-in-the-middle position, able to intercept and alter communications.
- Arrange to somehow inject malicious client-side JavaScript code into the user's browser.
- Communicate with the injected JavaScript in real time.

Coming up next...

- The "Official" fix, why GRC doesn't care, and why this entire thing is COMPLETELY bogus.

< < < SPONSOR INSERT HERE! > > >

The Official Fix: TLS_FALLBACK_SCSV

- "TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks"
- https://tools.ietf.org/html/draft-bmoeller-tls-downgrade-scsv-01
- A pseudo cipher suite value that allows clients to statical assert their understanding and support for TLS protocols.

Why GRC is not vulnerable.

- I've never used cookies. My eCommerce system is 100% cookie-free.

Why this ENTIRE THING is COMPLETELY bogus...

- If an attacker has some means for installing a malicious client which is capable enough to assist the MITM in this attack, there are FAR EASIER and FASTER means for the client to side-channel leak any site-specific script-accessible data to a MITM attacker.

- In other words... this is an attack that has been reverse-engineered from a true protocol weakness.

- And, yes, protocol weaknesses should always be fixed.

- But... any attacker who COULD arrange to perform the attack wouldn't NEED ANY PROTOCOL weakness to leak information much more easily, much faster and more efficiently to any MITM who was monitoring the connection.

- And, moreover, the attack now becomes much simpler:
  - (1) Inject malicious Javascript.
  - (2) Passively monitor communications for client's deliberate side-channel information leakage.