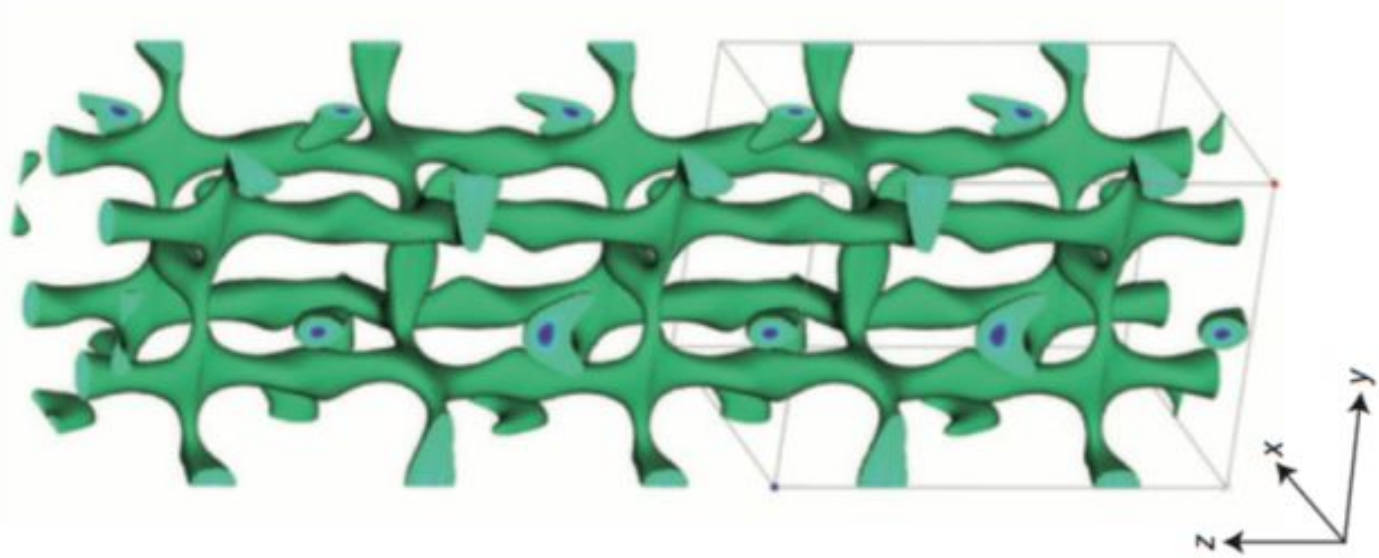
Security Now! #552 - 03-22-16

DROWN

This week on Security Now!

- FBI postpones today's court hearing.
- Matthew Green and four students poked a hole in iMessage.
- Another side channel attack, this one against mobile devices.
- Massive malvertising campaign hits many major sites.
- Lenovo back in the dog house… again!
- 2016 Pwn2Own competition results.
- Android Stagefright module even more unsafe than believed.
- Miscellaneous bits about Steve's chosen drive imaging solution, the return of AnyDVD, new solid-state battery tech… and
- A closer look at the DROWN vulnerability & attack (and why security is hard!)

A solid state Lithium (plus other stuff) crystal lattice



Security News

The FBI has postponed the court hearing... it can still be picked up later.

In their filing yesterday (Monday) afternoon, Federal prosecutors wrote: "On Sunday, March 20, 2016, an outside party demonstrated to the FBI a possible method for unlocking Farook's iPhone. Testing is required to determine whether it is a viable method that will not compromise data on Farook's iPhone. If the method is viable, it should eliminate the need for the assistance from Apple Inc. ('Apple') set forth in the All Writs Act Order in this case."

Jennifer Granick (@granick)

- Director of Civil Liberties for the Center for Internet and Society at Stanford Law School.
- "We won't know whether the government withdraws it motion until at least April 5th."
- RT: *Marcia Hofmann (@marciahofmann)*
    - Digital rights lawyer in private practice.
    - Adjunct prof @UCHastingsLaw, special counsel to @EFF.
    - Ex-@EFF & @epicprivacy. PGP
    - TWEET: "Let's be clear: the government hasn't dropped its case against Apple. A hearing was canceled, that's all."

John Paczkowski (@JohnPaczkowski)

- Managing Editor, BuzzFeed SF, http://www.buzzfeed.com/tech.
- Formerly Re/code & AllThingsD.
- Apple on the latest twist in #AppleVsFBI: Government has provided us with absolutely no information about this outside party hack.

Maybe John MacAfee gave them a call? <g>

iMessage -- and why crypto is difficult to get right

Matthew Green and four of his research students managed to poke a hole in Apple's iMessage encryption... which was (sort of) fixed yesterday with iOS v9.3
http://blog.cryptographyengineering.com/2016/03/attack-of-week-apple-imessage.html

- *Yesterday (Monday) Matthew Green wrote the following:*

Today's Washington Post has a story entitled "Johns Hopkins researchers poke a hole in Apple's encryption", which describes the results of some research my students and I have been working on over the past few months.

As you might have guessed from the headline, the work concerns Apple, and specifically Apple's iMessage text messaging protocol. Over the past months my students Christina Garman, Ian Miers, Gabe Kaptchuk and Mike Rushanan and I have been looking closely at the encryption used by iMessage, in order to determine how the system fares against sophisticated attackers. The results of this analysis include some very neat new attacks that allow us to -- under very specific circumstances -- decrypt the contents of iMessage attachments, such as photos and videos.

Now before I go further, it's worth noting that the security of a text messaging protocol may not seem like the most important problem in computer security. And under normal circumstances I might agree with you. But today the circumstances are anything but normal: encryption systems like iMessage are at the center of a critical national debate over the role of technology

companies in assisting law enforcement.

A particularly unfortunate aspect of this controversy has been the repeated call for U.S. technology companies to add "backdoors" to end-to-end encryption systems such as iMessage. I've always felt that one of the most compelling arguments against this approach -- an argument I've made along with other colleagues -- is that we just don't know how to construct such backdoors securely. But lately I've come to believe that this position doesn't go far enough -- in the sense that it is woefully optimistic. The fact of the matter is that forget backdoors: we barely know how to make encryption work at all. If anything, this work makes me much gloomier about the subject.

But enough with the generalities. The TL;DR of our work is this:

Apple iMessage, as implemented in versions of iOS prior to 9.3 and Mac OS X prior to 10.11.4, contains serious flaws in the encryption mechanism that could allow an attacker -- who obtains iMessage ciphertexts -- to decrypt the payload of certain attachment messages via a slow but remote and silent attack, provided that one sender or recipient device is online. While capturing encrypted messages is difficult in practice on recent iOS devices, thanks to certificate pinning, it could still be conducted by a nation state attacker or a hacker with access to Apple's servers. You should probably patch now.

- Apple protected iMessages by signing them... but NOT by authenticating them (!).
- At first blush, signing would at first appear to perform both functions, but it actually doesn't.
- … because it’s possible for an attacker to replace the authentic signature and re-sign with THEIR signature.
- This subtle hole allows an attacker to make small modifications to a message, send it to the original recipient under their signature, and observe the nature of the error returned.
- With time, an attacker can reverse engineer the entire, previously unknown, message.

- The iMessage protocol is fundamentally broken, so the "fix" is the horrific kludge of caching the most recent failures and detecting repeated attempts. Sort of an IDS (intrusion detection system) for iMessage… which is NOT what anyone wants.

- Matthew suggests that the only good long-term solution is for Apple migrate away from their current iMessage architecture over to something carefully designed... such as the WhisperSystems Signal messaging protocol.

Another side-channel attack... this time against mobile devices

- A group of security researchers at the University of Tel Aviv
- https://eprint.iacr.org/2016/230.pdf
- https://eprint.iacr.org/2016/230

Abstract:
We show that elliptic-curve cryptography implementations on mobile devices are vulnerable to electromagnetic and power side-channel attacks. We demonstrate full extraction of ECDSA secret signing keys from OpenSSL and CoreBitcoin running on iOS devices, and partial key leakage from OpenSSL running on Android and from iOS's CommonCrypto. These non-intrusive attacks use a simple magnetic probe placed in proximity to the device, or a power probe on the phone's USB cable. They use a bandwidth of merely a few hundred kHz, and can be performed cheaply using an audio card and an improvised magnetic probe.

In this paper we demonstrate the a side channel attack on Elliptic Curve Cryptography (ECC) running on a smartphone which simultaneously achieves the following properties:

- 1. Real-World Implementations. We attacked the ECDSA implementation of OpenSSL running on iOS devices (iPhone and iPad) as well as Android devices. In particular, we attacked the CoreBitcoin library, based on OpenSSL, which is used by popular Bitcoin wallets on iOS devices. We also attacked the built-in ECDSA implementation of iOS's CommonCrypto library.
- 2. Non-Invasive. The demonstrated attacks are non invasive and can be conducted by merely placing a magnetic probe in the proximity of the device, or using a power tap on its USB charging cable. The attack does not require any software to be installed on the device, and does not require opening the device's case (see Figures 1 and 9).
- 3. Cheap EM and power analysis. Our attack utilizes physical emanations (electromagnetic or power) at frequencies below 200 kHz, which is well below the GHz-scale processor clock speed. Consequently, our attack can acquire secret-key information using cheap, compact and readily available equipment, such as sound cards and improvised probes.

Moral: Attacks never get worse… they only get better.

Many big-name websites hit by a rash of malicious ads spreading... crypto ransomware.

https://www.trustwave.com/Resources/SpiderLabs-Blog/Angler-Takes-Malvertising-to-New-Heights/

- A new malvertising campaign appears to have exposed tens of thousands of visitors in just 24 hours.

- The New York Times, the BBC, MSN, AOL, my.xfinity.com, nfl.com, realtor.com, theweathernetwork.com, thehill.com, newsweek.com, answers.com, zerohedge.com, infolinks.com.
- Affected advertising networks include those owned by Google, AppNexis, AOL, and Rubicon.
- The "Angler" toolkit is being dropped onto victim computers.
- One JSON-based file being served in the ads has more than 12,000 lines of heavily obfuscated code. When researchers deciphered the code, they discovered it enumerated a long list of security products and tools it avoided in an attempt to remain undetected.
- The most widely seen infection domain name in the current campaign is brentsmedia[.]com. WHOIS records show it was owned by an online marketer until January 1, when the address expired. It was snapped up by its current owner on March 6... a day before the malicious ad onslaught started.
- What can users do? Reduce "attack surface" by uninstalling things like Adobe Flash, Oracle Java, Microsoft Silverlight, and other third-party browser extensions unless absolutely required.
- And… run with ads blocked by default, unblocking only when/as/if necessary.
- We need a stronger solution for quarantining our web browsers.

And speaking of the "Angler" exploit kit...

- https://labsblog.f-secure.com/2016/03/15/lenovo-startpage-pushed-angler
- F-Secure reports that they found a Lenovo site pushing Angler.
- "startpage.lenovo.com" was redirecting visitors to an instance of Angler.

Pwn2Own 2016

Researchers won a total of $460,000 in cash for disclosing 21 new vulnerabilities in:

- (This was about $100,000 LESS than the previous year's haul of $550,000.)
- Windows 10 (6 vulns, the most in any single target of the competition.) But...
- OS X (not far behind, with 5 vulns.)
- Flash
- Safar - 3 successful attacks.
- Edge - significantly more secure than previous versions of IE.
- Chrome - still the most secure of all browsers. All attacks failed, at least partially.
- Firefox was not present... because among the elite hackers it's not considered hard enough to beat.Firefox's codebase is aging, it doesn't have strong sandboxing and it needs a rewrite in Mozilla's "Rust" memory-safe language. It's aging add-on architecture needs to be scrapped and redesigned.

Android's poorly written "StageFright" media module just keeps on giving.

- Beautifully produced write-up: (truly, an AMAZING piece of work!)
  - https://www.exploit-db.com/docs/39527.pdf
- Last year Stagefright imperiled 950 million Android phones through MMS vulnerabilities.
  - Zimperium
  - MPEG-4 file format implementation mistake.
- Israeli security firm NorthBit
- Now "Metaphor" puts nearly 300 million Android phones at risk of drive-by attacks... by exploiting the SAME VULNERABILITY which, for some reason, Google never fixed!
- Proof-of-concept exploit works against Android versions 2.2 through 4.0, 5.0 and 5.1.
- Android v4.1 added ASLR -- and the Metaphor hack bypasses it.
- But it's even worse than that:
  - The new Stagefright hack leaks ASLR specifics leading to much more reliable code execution than was previously possible.
- v5.1 was thought to be immune, but no longer. (19% of Android phones.)
- Works "best" on:
  - Nexus 5 models with a stock ROM
  - Also works on the HTC One
  - LG G3
  - Samsung S5
- Depending upon the vendor, a drive-by attack requires anywhere from 20 seconds to two minutes to work.
- HOWEVER!!!! ---> the attack IS strictly device-specific, making a universal exploit infeasible.
- But... per-device payloads COULD be developed.
- In late news: Google has stated: "Android devices with a security patch level of October 1, 2015 or greater are protected because of a fix released for this issue (CVE-2015-3864) last year. As always, we appreciate the security community's research efforts as they help further secure the Android ecosystem for everyone."

Moral: Keep mobile devices updated, and only use mobile devices from reputable vendors who DO keep their devices updated, and only so long as they ARE being kept updated.

Miscellany

What's Steve's Drive Imaging solution?

Background:

- Microsoft declared that older OS versions would not be supported starting with Skylake (now!)
- Has since backed off of that due to the industry reacting as I did.
- So I built "my last PC"

Mentioned nightly images from SSD to mirrored 3TB drives.

"Image For Windows" by TeraByte

- https://www.terabyteunlimited.com/image-for-windows.htm
- Image for DOS & Linux
- 30-Day Trial

AnyDVD is back, sort of...

Someone wrote:

Hey Steve and Leo,

Steve you mentioned AnyDVD on SN549, if you have a license to any of Slysoft's products you can still update them as the developers have resurrected the software.

Slysoft, the company behind AnyDVD, has been closed down. But it appears the developers were not based in the same country (Antigua) as Slysoft.

The new site is located here: https://forum.redfox.bz (downloads top right of main site).

TorrentFreak explains what is going on:
https://torrentfreak.com/anydvd-is-back-but-dont-call-us-pirates-developer-says-160302
https://www.redfox.bz/download.html

New solid state lithium battery tech

- http://arstechnica.com/science/2016/03/new-lithium-battery-ditches-solvents-reaches-supercapacitor-rates/
- A team of academic researchers working with Toyota.
- Can charge and discharge at extremely high super capacitor rates.
- Typical batteries use a liquid (leakable) electrolyte.
- Typical fluid electrolyte interfaces the electrodes, dissolves the lithium ions, and allows ionic movement to and fro between the electrodes.
- No fluid electrolyte, so can operate across wide (-30 to 100C) range.
- Cannot not explode (at least in the way Li-Po batteries can.)
- 20 times the lithium ion density compared with a fluid suspension.
- Strong cycle life:

- Initial 10% loss of theoretical capacity after its 1st cycle.
- But 500 cycles later, only loses another 15%, retaining 75% of original capacity.

SpinRite

Using SpinRite on Solid State Media:

With solid state mass storage, as with electromagnetic spinning mass storage drives, manufacturers have brought their costs way down and made their products competitive mostly by cramming too much data into too little space... and in doing so they have traded away some of the reliability margin from their products. For the most part, today's spinning and solid state drives work well. But as SpinRite's owners learn, it's never fun to become a statistic.

When Flash drives and SSDs began to penetrate the market, we had no idea whether SpinRite's days might be numbered. We knew that spinning drives would still be around for a long time, but it was certainly possible that SpinRite's recovery technology might have nothing to offer drives where nothing was spinning. Then we started receiving reports (and not just a few!) from SpinRite's existing users who had used it to successfully recover their solid state drives!

It turns out that in order to cram too many bits into too small a space, the same "error correction codes" (ECC) that were developed for spinning drives were employed for solid state drives. The reason is: the solid state bits have been made so tiny that they are always on the verge of being unreadable. Just like a spinning drive's bits. So the same technologies GRC developed years ago for reading those unreadable bits still work on today's and tomorrow's solid state mass storage technologies. And that shows no sign of changing.

HOWEVER...
Unlike spinning disks, the process of changing any of a solid state drive's bits -- writing to the drive -- ever-so-slightly fatigues those changed bits. They become just a tiny bit less reliable. Over a long period of time, solid state drives can actually wear out and develop bad bits -- which is another of the things SpinRite can deal with. But this write fatigue means that SpinRite should only be used at Level 2, not Level 4 on solid state drives. Level 2 places SpinRite into a "read mostly" mode where it will rapidly scan the solid state drive for any trouble and will only switch into recovery and testing mode (Level 4) in the spots where trouble actually exists. Though solid state mass storage can develop new detects over time, the built-in controller, working in concert with SpinRite, is able to take bad spots out of service and keep the drives running well.

So... SpinRite is definitely useful for solid state mass storage data recovery and repair, and it should be used at Level 2 on those drives.

DROWN

"**D**ecrypting **R**SA with **O**bsolete and **W**eakened e**N**cryption"

CVE-2016-0800
A novel cross-protocol attack that uses SSLv2 handshakes to decrypt TLS sessions.
https://drownattack.com/

Quoting from the introduction:
DROWN is a serious vulnerability that affects HTTPS and other services that rely on SSL and TLS, some of the essential cryptographic protocols for Internet security. These protocols allow everyone on the Internet to browse the web, use email, shop online, and send instant messages without third-parties being able to read the communication.

DROWN allows attackers to break the encryption and read or steal sensitive communications, including passwords, credit card numbers, trade secrets, or financial data. Our measurements indicate 33% of all HTTPS servers are vulnerable to the attack.

Modern servers and clients use the TLS encryption protocol. However, due to misconfigurations, many servers also still support SSLv2 (the predecessor to TLS). This support did not matter in practice, since no up-to-date clients actually use SSLv2. Therefore, even though SSLv2 is known to be badly insecure, until now, merely supporting SSLv2 was not considered a security problem, because clients never used it.

- 17% of current HTTPS servers still allow SSLv2 connections.

DROWN shows that merely supporting SSLv2 is a threat to modern servers and clients. It allows an attacker to decrypt TLS connections between up-to-date clients and servers by sending probes to a server that supports SSLv2 and uses the same private key.

Since certificates have historically been expensive and difficult to obtain, REUSING CERTIFICATES where possible made sense and was commonly done.

- An ADDITIONAL 16% of domains have "reused" certs, raising total vulnerable HTTPS servers to 33%.

- https://test.drownattack.com/?site=vmware.com
- https://test.drownattack.com/?site=twit.tv

To protect against DROWN, server operators need to ensure that their private keys are not used anywhere with server software that allows SSLv2 connections. This includes web servers, SMTP servers, IMAP and POP servers, and any other software that supports SSL/TLS.

OpenSSL has just been updated so that it is impossible to configure a TLS server in such a way that it would be vulnerable to DROWN.

Great additional information:
https://www.openssl.org/blog/blog/2016/03/01/an-openssl-users-guide-to-drown/

- Private key is NOT exposed. ONLY a session might be decrypted.
- NO EVIDENCE that DROWN has ever been employed in the wild.
- DROWN attackers *CAN* decrypt sessions RECORDED in the past.
- An OpenSSL patch one year ago (March 2015) coincidentally fixed a bug that allows attacks to succeed much more quickly -- in minutes on a laptop.
- DROWN scans found 4 million HTTPS server running EARLIER OpenSSL!

Profile of two attacks:

Slow Attack:
The more general attack exploits a combination of thus-far unnoticed protocol flaws in SSLv2 to develop a new and stronger variant of the Bleichenbacher attack. A typical scenario requires the attacker to observe 1,000 TLS handshakes, then initiate 40,000 SSLv2 connections and perform 2^50 offline work to decrypt a 2048-bit RSA TLS ciphertext.

An implementation of the attack and can decrypt a TLS 1.2 handshake using 2048-bit RSA in under 8 hours using Amazon EC2, at a cost of $440. Internet-wide scans found 33% of all HTTPS servers are vulnerable to this protocol-level attack, due to widespread key and certificate reuse.

Fast Attack:
An even faster attack can apply a newly discovered vulnerability in OpenSSL that was present in releases from 1998 to early 2015 (coincidentally fixed one year ago). *Given an unpatched SSLv2 server to use as an oracle, we can decrypt a TLS ciphertext in one minute on a single CPU*—fast enough to enable man-in-the-middle attacks against modern browsers.

26% of HTTPS servers are still vulnerable to this attack -- one year after OpenSSL was fixed!

Mostly... another lesson in how difficult security actually is.