Security Now! #582 - 10-18-16

Q&A #241

This week on Security Now!

- Some serious concerns raised over compelled biometric authentication,
- A detailed dive into the recently completed audit of VeraCrypt, the successor to TrueCrypt,
- More on web browsers fatiguing system main SSD storage,
- A bunch of interesting miscellany (including... are we living within a simulated reality?),
- And eleven questions and observations from our terrific listeners.

"You're Doing it Wrong" in a simple picture:

Security News

"Feds Walk Into A Building, Demand Everyone's Fingerprints To Open Phones"

- http://www.forbes.com/sites/thomasbrewster/2016/10/16/doj-demands-mass-fingerprint-seizure-to-open-iphones/

FORBES: In what's believed to be an unprecedented attempt to bypass the security of Apple iPhones, or any smartphone that uses fingerprints to unlock, California's top cops asked to enter a residence and force anyone inside to use their biometric information to open their mobile devices.

FORBES found a court filing, dated May 9 2016, in which the Department of Justice sought to search a Lancaster, California, property. But there was a more remarkable aspect of the search, as pointed out in the memorandum: "authorization to depress the fingerprints and thumbprints of every person who is located at the SUBJECT PREMISES during the execution of the search and who is reasonably believed by law enforcement to be the user of a fingerprint sensor-enabled device that is located at the SUBJECT PREMISES and falls within the scope of the warrant." The warrant was not available to the public, nor were other documents related to the case. According to the memorandum, signed off by U.S. attorney for the Central District of California Eileen Decker, the government asked for even more than just fingerprints: "While the

government does not know ahead of time the identity of every digital device or fingerprint (or indeed, every other piece of evidence) that it will find in the search, it has demonstrated probable cause that evidence may exist at the search location, and needs the ability to gain access to those devices and maintain that access to search them. For that reason, the warrant authorizes the seizure of 'passwords, encryption keys, and other access devices that may be necessary to access the device."

Legal experts were shocked at the government's request. Marina Medvin of Medvin Law said: "They want the ability to get a warrant on the assumption that they will learn more after they have a warrant. Essentially, they are seeking to have the ability to convince people to comply by providing their fingerprints to law enforcement under the color of law – because of the fact that they already have a warrant. They want to leverage this warrant to induce compliance by people they decide are suspects later on. This would be an unbelievably audacious abuse of power if it were permitted."

Jennifer Lynch, senior staff attorney at the Electronic Frontier Foundation (EFF), added: "It's not enough for a government to just say we have a warrant to search this house and therefore this person should unlock their phone. The government needs to say specifically what information they expect to find on the phone, how that relates to criminal activity and I would argue they need to set up a way to access only the information that is relevant to the investigation. The warrant has to be particular in how it describes the place to be searched and the thing to be seized and limited in scope. That's why, if a government suspects criminal activity to be happening on a property, and there are 50 apartments in that property, they have to specify which apartment and why and what they expect to find there."

Jennifer Lynch added... "We've never seen anything like this."

VeraCrypt has been audited by QuarksLab courtesy of the OSTIF

OSTIF: The Open Source Technology Improvement Fund

- Bug Bounties
- Direct Code Improvement Through Grants
- Professional Audits

TrueCrypt --> VeraCrypt

About ten weeks ago, on August 1st (2016), the OSTIF announced:

OSTIF is proud to announce that we have come to an agreement to fully fund an audit of VeraCrypt. Using funds that were donated by DuckDuckGo and VikingVPN, we plan to hire QuarksLab to go over the code and search for vulnerabilities and backdoors.

VeraCrypt is a crucial piece of open-source software that can encrypt any storage medium with powerful and highly tamper-resistant encryption that greatly enhances the personal security of anyone that uses it. An audit of the code brings fresh professional perspectives and a deep analysis of the code to search for vulnerabilities.

As of our agreement today, QuarksLab will be executing their auditing work in mid-August with completion and results before mid-September. The team has been instructed to give any results of this audit directly to the lead developer of VeraCrypt using heavily encrypted communications. This is to prevent their research from leaking zero-day vulnerabilities to the public, and so that the OSTIF does not have access to the results ahead of the public.

Once all parties are satisfied that any major vulnerabilities have been patched, we plan to post the results of this audit to all three of our websites (OSTIF.org, QuarksLab, and VeraCrypt) simultaneously.

This audit is very exciting for us, as it is the first time that we are converting promises and goals into results! We look forward to bringing the open-source community and the public closer together, and we are committed to fighting hard for the technical solution to the privacy problem.

32 man-days later...
https://ostif.org/the-veracrypt-audit-results/
https://ostif.org/wp-content/uploads/2016/10/VeraCrypt-Audit-Final-for-Public-Release.pdf

What was found?

- No huge showstoppers, but a handful of things that needed to be cleaned up.
- They were mostly weaknesses that COULD be used in highly targeted attacks to compromise the target's data privacy and security.
- Two broad categories:
    - Troubles inherited from TryeCrypt
    - New troubles introduced by new VeraCrypt features.

From TrueCrypt:

- The PBKDF2 algorithm was not acceleration resistant and did not employ a sufficiently large iteration count.
    - VeraCrypt has increased the iteration count from 1000 and 2000 (depending upon the hash algorithm and its usage) to 200,000 and 655,331. And VeraCrypt also allows the count to be manually specified through the use of a PIM: Personal Iterations Multiplier. (So if you don't mind waiting longer at boot or mount time, you can make brute-forcing much more difficult.)

- Sensitive information might be paged out from kernel stacks.
    - VeraCrypt doesn't change this. They simply state that this will be a vulnerability unless system partition is encrypted and the paging file is therefore written to an encrypted partition.

- Concerns over exploitation of known problems with the older version of decompression code used by TrueCrypt.
    - Everyone agrees that while this would be nice to fix, it's not really exploitable since an attacker would need to first modify the compressed code to induce an error, and that requires an administrator or physical access to the system... both which are outside of the protection guarantees of TrueCrypt.

- Windows kernel driver uses the "memset" function to zero memory before its release.
    - (Remember that some clever compilers notice that memory is being cleared then freed... and "optimize out" the memory zeroing.)
    - This was fixed by using RtlSecureZeroMemory instead of memset.

- Several other subtle TrueCrypt kernel driver problems, FIXED by VeraCrypt.

- There is an unfixed information leak in the kernel driver, where the kernel driver can be used to check for the existence of files without access permission. This was deliberately done so that TrueCrypt could check for the presence of the boot loader on disk without requiring administrator privileges. But this could be exploited by any unprivileged process. VeraCrypt had not addressed this issue.

- The non-instruction accelerated (non AES-NI) software implementations of AES are susceptible to cache timing attacks which could be leveraged to obtain the operating AES symmetric key.
    - VeraCrypt disclaims "responsibility" for this by stating: "VeraCrypt does not: Secure any data on a computer if the computer contains any malware (e.g. a virus, Trojan horse, spyware) or any other piece of software (including VeraCrypt or an operating system component) that has been altered, created, or can be controlled, by an attacker.
    - However... all recent Intel chips provide support for the AES-NI instruction which completely eliminates AES cache timing attacks.

- Keyfile mixing is horrific -- a home-grown hash using CRC-32.
    - Remember that the Cyclic Redundancy Check was never designed to be a cryptographic integrity mechanism. It is meant for efficient error detection. So it is intended to prevent accidents, not thwart attacks.
        - This has a number of theoretical consequences, including:
        - From a set of valid keyfiles, it is possible to create another distinct one (i.e. allowing to mount a volume),
        - It is possible to create keyfiles that do not modify the keyfile pool,
        - From a set of keyfiles, it is possible to create a new keyfile which removes the security brought by the former set of keyfiles (i.e. which zeroes the keyfile pool),
        - From a known passphrase, it is possible to create a keyfile which removes the security brought by this passphrase.
    - This problem has long been known (since 2008), but the TrueCrypt authors denied this being any concern since the attacks require the prior knowledge of a secret or the prior manipulation of a machine.
    - This SHOULD be fixed.
- Volume headers are also "authenticated" (checksummed) with CRC-32.
    - This should also be fixed, but has not been, since it would break compatibility with existing installations. However, though sloppy, it's unclear how this could be leveraged for an attacker's benefit.

- James Forshaw with Google's Project Zero discovered and reported a pair of problems in the TrueCrypt kernel driver.
    - The "Drive Letter Symbolic Link Creation" flaw allowed for a high vulnerability elevation of privilege.
    - Both have been fixed.

From VeraCrypt:

- When booting from a BIOS, the user's password length can be obtained:
    - The BIOS saves keyboard keystrokes into a 32-byte, 16-character ring buffer. Once the system is booted, VeraCrypt prevents this potential leak by reaching down and zeroing the 32-bytes in the BIOS Data Area. HOWEVER, the "Head" and "Tail" pointers, located just before it, are not cleared. This allows an attacker to gain information about the length of the boot-time password... and this would significantly accelerate password brute-forcing.

- VeraCrypt continues using known-vulnerable compression and decompression algorithms, though exploiting the known vulnerabilities is believed to be difficult.
    - To decompress the bootloader when the hard drive is encrypted.
    - To create and check the recovery disks if the system is encrypted and uses UEFI.
    - During the installation to extract programs.
    - An example of such manipulation would be that a VeraCrypt recovery disk could theoretically be manipulated to induce a buffer overflow during recovery that could compromise the user's security by, for example, writing the user's password or decrypted master key to some attacker-accessible location.

- VeraCrypt added a PIM (the Personal Iterations Multiplier), but the math used for large secure values can overflow and wrap, resulting in a tiny effective value and much reduced security.

- Command-Line vulnerabilities:
    - TrueCrypt allowed the volume password to be provided as a command-line argument. VeraCrypt added support for SmartCards, and allows the SmartCard PIN to also be included on the command-line. VeraCrypt previously documented the insecurity of the password, saying: "Warning: This method of entering a volume password may be insecure, for example, when an unencrypted command prompt history log is being saved to unencrypted disk."
    - The auditors strongly object to the provision of secure information on command lines.

- VeraCrypt added four non-Western cryptographic algorithms to the project.
    - One of them GOST89 is a Russian symmetric cipher designed in the 1970's. While it uses a 256-bit key, its block size is only 64-bits... and we know that 64-bit block sizes no longer provide sufficient security margins.
    - Moreover, an attempt was made to create a 128-bit cipher from two 64-bit GOST iterations using CBC (cipher block chaining)... but doing this has multiple security implications.
    - The auditors strongly recommended that GOST89 be removed from VeraCrypt.

- A handful of other things were spotted and rated by the auditors as "informational".
    - What this mostly shows us, is that the auditors really did study the code from an adversarial perspective, which is exactly what we want and need.

- The UEFI boot loader still leaves the concern over the user's password remaining in a low memory buffer:
    - Each UEFI driver maintains its own keystroke buffers and, unlike the BIOS, the buffer's internal address and location are unknown. Examination of Intel developer kit sample code shows they pull the keystroke from the underlying BIOS. The ConsoleIn API does have a "Reset" method, but there is no guarantee of what that does on any specific implementation.

- A problem was found with recent support for password changing:
    - The code endeavors to wipe the user's boot time password from RAM, but it fails to similarly clear the new password after a password change.

The audit concludes:
This audit, funded by OSTIF, required 32 man-days of study. It shows that this follow-up of TrueCrypt is very much alive and evolves with new functionalities like the support of UEFI.
The results shows that evaluations at regular intervals of such difficult security projects are not an option. When well received by the project's developers, they provide useful feedbacks to help the project mature. The openess of the evaluation results help build confidence in the product for the final users.

VeraCrypt has issued a number of fixes for many of these problems:
- GOST89 was removed.
- The compression & decompression were replaced with secure zip libraries (libzip).
- Password length leakage and password buffers are being cleared (as much as possible).
- Other bootloader vulnerabilities were resolved.

VeraCrypt is now at v1.19.

Web browsers continually writing to "disk" -- or SSD!
http://www.ghacks.net/2016/09/26/display-the-disk-activity-in-bytes-of-any-process-in-windows/
- FF: about:config / browser.sessionstore.interval
    - Defaults to "15000" - every 15 seconds.

- Firefox:
    - Terminate Firefox.
    - Edit the %appdata%\Mozilla\Firefox\profiles.ini
    - Set IsRelative=x value to zero, and edit the path.
    - Move the existing profile folder to the new location.
    - The profiles.ini file:
        - [General]
          StartWithLastProfile=0
          [Profile0]
          Name=default
          IsRelative=1
          Path=Profiles/{nonce}.default
          Default=1

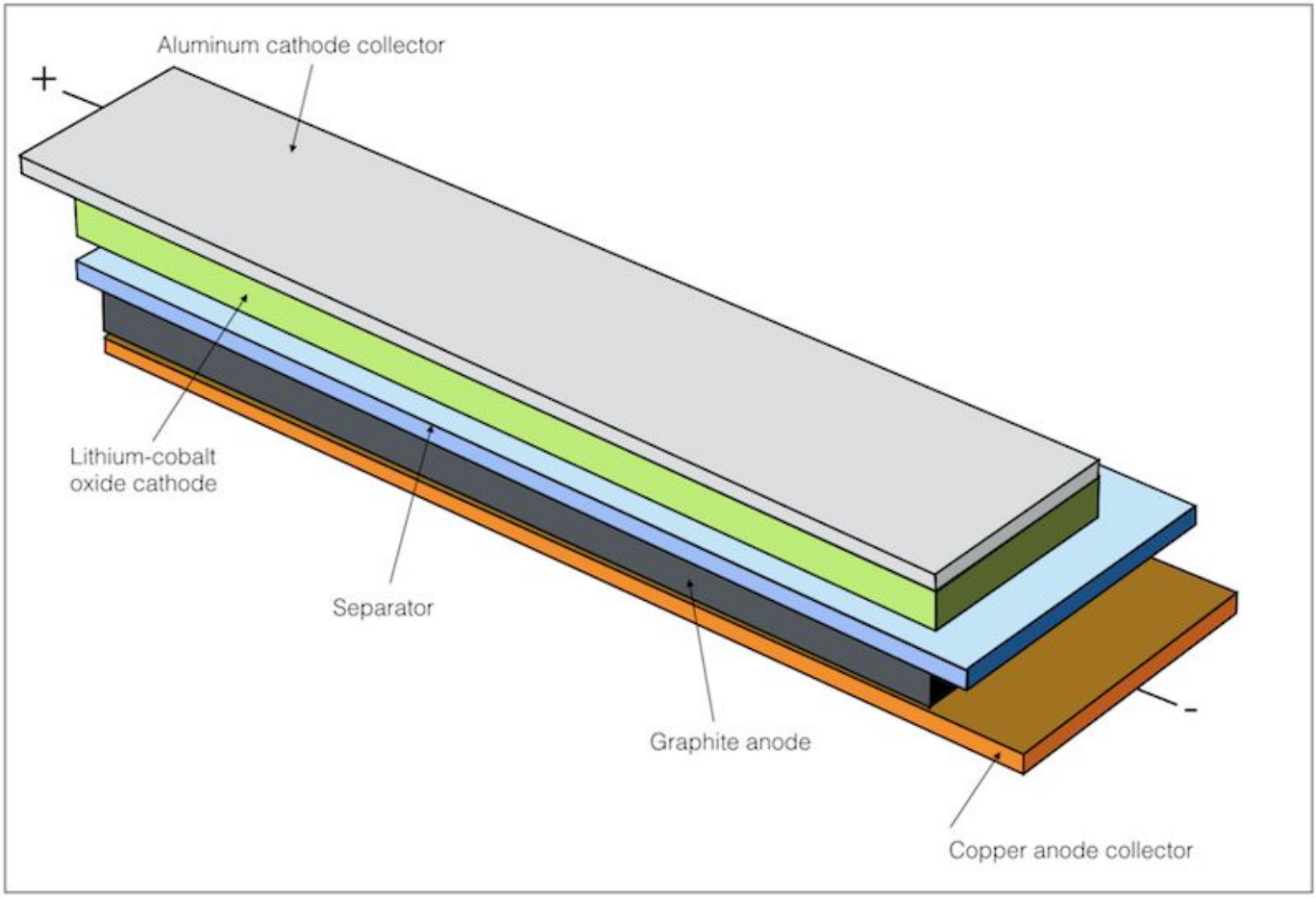
Miscellany

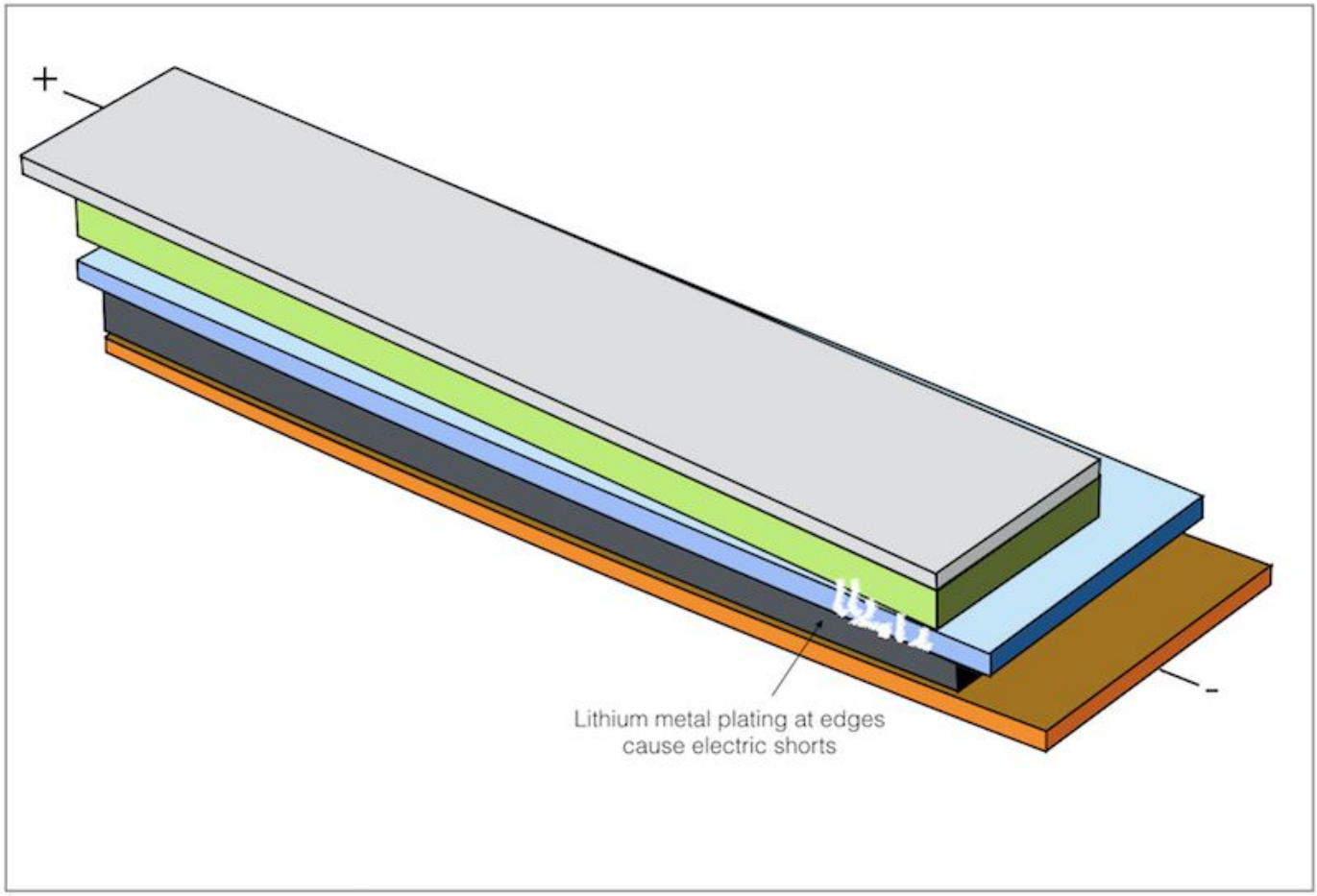
Are we living in a simulation?

- The setting is the Recode Code Conference 2016. Elon Musk is on stage with interviewers Walt Mossberg and Kara Swisher... and taking questions from the audience...
- Josh Topolsky, a co-founder of The Verge in the audience, asks Elon whether he thought we might be living in a simulation.
- We learned that Elon has ruminated about and hashed through this topic --so thoroughly already-- that he and his brother have formally banned its further discussion anytime they are in hot tubs. Its discussion is banned from the hot tub.
- Now, this is not the sort of existential question that can ever be answered definitely... so it actually IS the perfect subject for hot tub debate.
- However, in answering the questioner, Elon put a number to it, stating several times (because people were so surprised) that he believed we WERE, indeed, living within a simulation and, in fact, that there was only a one in a billion chance that we were NOT simulated and were living in what he referred to as a "base reality."
- And this is an ongoing theme for Elon. Back in 2014, at Vanity Fair's 2014 New Establishment Summit, Elon made the case that our lives are not at all what we think they are. Musk concluded: “There’s a one in a billion chance that this is reality.”
- During the Recode Code Conference, he broke it down: “The strongest argument for us being in a simulation is the following: 40 years ago, we had Pong. Two rectangles and a dot. Now, 40 years later, we have photo-realistic 3D with millions playing simultaneously. If you assume any rate of improvement at all, then the games will become indistinguishable from reality. It would seem to follow that the odds that we’re in base reality is one in millions.”
- HOWEVER... the comparisons to "The Matrix" is not at all what Elon means or believes.
- STTNG: Watson's arch villain an Professor Moriarty.

The Li-Ion battery challenge

http://www.powerelectronicsnews.com/technology/pushing-to-the-very-edge-of-safe-li-ion-charging


Aluminum cathode collector
+
Lithium-cobalt
oxide cathode
Separator
Graphite anode
-
Copper anode collector


+
Li
Lithium metal plating at edges
cause electric shorts
-

Qnovo:
http://qnovo.com/qnovation-blog/

- 104. THE PERILS OF 4.4 VOLTS
- 103. THE REAL SCIENCE BEHIND BATTERY SAFETY
- 102. IT'S EASY, BUT NOT RIGHT, TO PICK ON SAMSUNG
- 101. MAKING SENSE OF 100 KWH
    - Tesla Motors announced today upgraded versions of the Model S and X boasting 100 kWh battery packs, up from 90 kWh used in their earlier top-of-the-line electric vehicles. One hundred kilowatt-hours sounds like a lot, and it is, but I bet that many readers don't have an intuitive sense of this amount of energy. This is what this post is for.
- 100. THE STATE OF THE LITHIUM ION BATTERY

HSF - Dosage Determination Protocol now published.

http://treksit.com/

- A lovely, simple, bit of online JavaScript

The two-book consensus appears to be:

- "du·ol·o·gy" noun a pair of related novels, plays, or movies.
- "bilogy" (trilogy, bilogy)

SpinRite

Bob Port
Location: Albany, NY
Subject: How SpinRite kicks butt
Date: 09 Oct 2016 16:24:03
:
Steve,

It finally happened. I knew it eventually would. A 5-year-old HD on a perfectly servicable Dell Windows PC started acting up. It was just getting old, like all of us always are. I downloaded your utility, SpinRite, and, viola, my HD was back in business, virtually good as new. How simple can life be?

My eternal thanks,

Bob Port
Albany, NY

(((Note: I have long resisted many requests for adding this or that feature to SpinRite. But it makes so much sense for it to simply do what it does... just fix the disk. Recently, I offered to build the "Beyond Recall" feature into SpinRite, but the community pushed back hard and fast on that. They want it separate.)))