Security Now! #456 - 05-20-14

Harvesting Entropy

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!

- Mozilla & Firefox support DRM video.
- China bans the use of Windows 8.
- New Swiss-based end-to-end encrypted eMail solution emerges from beta.
- Ladar Levinson sums up his nightmare with the US government.
- A surprisingly (and disturbingly) effective 2-factor authentication bypass,
- How adding the word "Quantum" to anything makes it exciting.
- Various Miscellany, SQRL & SpinRite... and
- Harvesting Entropy.

Security News:

Mozilla & Firefox to support DRM video.

- http://www.theguardian.com/technology/2014/may/14/firefox-closed-source-drm-video-browser-cory-doctorow

Reuters reports: No Win8 on Chinese Government Computers:

- China's Central Government Procurement Center issued a ban on installing Windows 8 on government computers in apparent protest over Microsoft's decision to stop supporting Windows XP which still has 50% share of Chinese desktops.
- Reuters reports: "The official Xinhua news agency said the ban was to ensure computer security after Microsoft ended support for its Windows XP operating system, which was widely used in China."
- And notes that: "Neither the government nor Xinhua elaborated on how the ban supported the use of energy-saving products, or how it ensured security."

ProtonMail.com

- End-to-end encryption
- <quote> We support sending encrypted communication to non-ProtonMail users via symmetric encryption. When you send an encrypted message to a non-ProtonMail user, they receive a link which loads the encrypted message onto their browser which they can decrypt using a decryption passphrase that you have shared with them.

- Paid accounts: $5/month for 1GB storage + Bitcoin or cash for anonymity.
- Switzerland-based servers.
- Android & iPhone apps expected by the end of the summer.
- Close to 20,000 users, new sign-ups closed while servers added.
- That's all wonderful… but:
    - Metadata concern.
    - Authentication concern.
    - Ease-of-use brings ease-of-compromise.

The Guardian (Today: Tuesday, May 20th): **"Secrets, lies and Snowden's email: why I was forced to shut down Lavabit"**

- "For the first time, the founder of an encrypted email startup that was supposed to insure privacy for all reveals how the FBI and the US legal system made sure we don't have the right to much privacy in the first place."
- http://www.theguardian.com/commentisfree/2014/may/20/why-did-lavabit-shut-down-snowden-email
- <quote> The problem here is technological: until any communication has been decrypted and the contents parsed, it is currently impossible for a surveillance device to determine which network connections belong to any given suspect. The government argued that, since the "inspection" of the data was to be carried out by a machine, they were exempt from the normal search-and-seizure protections of the Fourth Amendment.

  More importantly for my case, the prosecution also argued that my users had no expectation of privacy, even though the service I provided – encryption – is designed for users' privacy.

  If my experience serves any purpose, it is to illustrate what most already know: courts must not be allowed to consider matters of great importance under the shroud of secrecy, lest we find ourselves summarily deprived of meaningful due process. If we allow our government to continue operating in secret, it is only a matter of time before you or a loved one find yourself in a position like I did – standing in a secret courtroom, alone, and without any of the meaningful protections that were always supposed to be the people's defense against an abuse of the state's power.

"How I bypassed 2-Factor-Authentication on Google, Facebook, Yahoo, LinkedIn and others."

- http://shubh.am/how-i-bypassed-2-factor-authentication-on-google-yahoo-linkedin-and-many-others/
- The problem:
    - Second factor code can be delivered via voice to user's cellphone.
    - But, clever hacker noticed that busy cellphone diverts immediately to voicemail.
    - Voicemail was never really designed to be secure.
    - (We learned with celebrity voicemail hacking that many never have PINs enabled.)
    - Hacker gets the voice-mode 2nd-factor from victim's voicemail.
    - Logs onto their account.

- Mixed bag of responses:
    - **Facebook:**
        - We've temporarily disabled sending login approval codes via phone call while we investigate further. Our plan is to re-enable the system when we can prompt users for interaction as part of the phone call, which should prevent us from sending codes to voicemail boxes.
          Thanks,

    - **LinkedIn:**
        - Thank you for notifying us of this issue before publicly disclosing it.
        - While the potential impact for our members is limited, we have made the decision to temporality turn off the voice option in our Two-Step verification setting. We are working with the third-party vendor we use for this service to implement a fix. After the fix is in place, we will evaluate turning the voice option back on.

    - **Google:**
        - Hey,
        - Thanks for your bug report. We've taken a look at your submission and can confirm this is not a security vulnerability in a Google product. The attack presupposes a compromised password, and the actual vulnerability appears to lie in the fact that the Telcos provide inadequate protection of their voicemail system. Please report this to the telcos directly.
        - Regards,
          Jeremy

    - **Yahoo:**
        - <quote> Yahoo's main services which allowed for 2FA were also vulnerable to the exploit I document above. In fact, the exploit to get into Yahoo accounts with 2FA enabled is even more severe as the attacker does not fully risk the victim knowing about account access on login.
        - <quote> 14 days from disclosure, Yahoo still hasn't replied, and hence they are still vulnerable to the 2FA bypass.

Headlines: "How to make a quantum random-number generator from a mobile phone."

- "Quantum" gets everyone excited, as if some exotic physics are in use.
- ---> Hum and Hiss.
    - Hum is 60 hertz from power lines.
    - Hiss ... is **quantum** noise.
- The real news here, though, is the BANDWIDTH (1.25 Gbit/sec.) of the "quantum" system. Since a high-resolution camera is a massive array of individual photon-sensing elements, this has massive bandwidth... massive data rate.

Academic research paper published a Eurocrypt this year

- "Science Daily: New algorithm shakes up cryptography"
- Researchers have solved one aspect of the discrete logarithm problem. This is considered to be one of the 'holy grails' of algorithmic number theory, on which the security of many cryptographic systems used today is based. They have devised a new algorithm that calls into question the security of one variant of this problem, which has been closely studied since 1976.
- They broke the DLP (discrete log problem) in any field with a small characteristic.
- Schneier: "It's nice work, and builds on a bunch of advances in this direction over the last several years. Despite headlines to the contrary, this does not have any cryptanalytic application -- unless they can generalize the result, which seems unlikely to me."

Syncthing.net

- <quote> Syncthing replaces Dropbox and BitTorrent Sync with something open, trustworthy and decentralized. Your data is your data alone and you deserve to choose where it is stored, if it is shared with some third party and how it's transmitted over the Internet.
  Using syncthing, that control is returned to you.

- Features:
  - Private. None of your data is ever stored anywhere else than on your computers. There is no central server that might be compromised, legally or illegally.
  - Encrypted. All communication is secured using TLS. The encryption used includes perfect forward secrecy to prevent any eavesdropper from ever gaining access to your data.
  - Authenticated. Every node is identified by a strong cryptographic certificate. Only nodes you have explicitly allowed can connect to your cluster.
  - Open Discourse. Development and usage is always open for discussion.
  - Open Source. All source code is available on GitHub — what you see is what you get, there is no hidden funny business.
  - Open Protocol. The protocol is a documented standard — no hidden magic.
  - Open Development. Any bugs found are immediately visible for anyone to browse — no hidden flaws.
  - Web GUI. Configure and monitor Syncthing via a responsive and powerful interface accessible via your browser.
  - Portable. Works on Mac OS X, Windows, Linux, FreeBSD and Solaris. Run it on your desktop computers and synchronize them with your server for backup.
  - Simple. Syncthing doesn't need IP addresses or advanced configuration: it just works, over LAN and over the Internet. Every machine is identified by an ID. Just give your ID to you friends, share a folder and watch: uPnP will do if you don't want to port forward or you don't know how.
  - Powerful. Synchronize as many folders as you need with different people.

- Private network VS Cloud?

- I would be FAR more comfortable with this than undocumented BitTorrent Sync.

Miscellany:

Mark Russinovich's "Rogue Code"

- Kindle, Hardcover, Audible (also in the UK, via Simon Zerafa)

Harry's

- Harrison Ward (@HBomb341)
  @SGgrc @harrys 1st shave today and WOW that's a good shave. Comparable to my 7-month old's baby butt... darn close to the same. Agree… slick handle.

Halt and Catch Fire Full 1st episode preview:
http://www.amctv.com/full-episodes/halt-and-catch-fire/3571290828001/i-o-sneak-preview

Godzilla:
~$200 million over 1st weekend.

SQRL:

- Revocation behind me.
- SQRL's revocation harvester is written and finished.
- Moving forward to implement the test of the client.
- "Identity Creation" wizard is next up...

SpinRite: Remote Control SpinRite Story:

Dear Steve,

I've always been my parents tech support guy. When I joined the US Air Force, they stationed me pretty far away from home. Thankfully, I've always been able to VNC into their computers and get things straightened out. This week I was presented with a rather unique challenge. My mother called to tell me that their computer was throwing a bunch of disk errors in Windows. Fixing this problem was particularly difficult for two reasons. First, VNC wouldn't help them if their hard disk suddenly crashed. Second, I'm currently deployed to Afghanistan, where getting a good enough connection to VNC into their computer can be difficult. I knew that SpinRite might be able to fix the disk errors but would I be able to walk my parents (who aren't the most tech savvy people) though it over the phone?

Believe it or not, I able to get a good enough connection to VNC into their computer and make a SpinRite bootable image. I was also able to instruct my mother, over the phone, how to boot into SpinRite and start the repair process. She called me the next day and told me that SpinRite had fixed 12 errors and that their computer was back to normal.

I cannot tell you how much we appreciate your product. It really saved the day.

Jeremy Webb

Harvesting Entropy

What are we trying to achieve with cryptographic entropy?

- With **NO** regard for an attacker:
  - Every bit has a totally random 50/50 chance of being a 1 or a 0.
  - With 256 bits, the risk of collision is ridiculously small:
    If we have a perfectly random source of 'p' sets of 'n' bits, the probability of collision is about $p^2/2^{n+1}$ (this is an approximation which is valid for "small" p, i.e. substantially smaller than $2^{n/2}$).
  - So, with 256 bits (n=256) and one billion messages ($p=10^9$) then the probability is about $4.3*10^{-60}$.

    An ELE (extinction level event) occurs about once every 30 million years on average. This leads to a probability of such an event occurring in the next second to about 10^{-15}. So that's 45 orders of magnitude ***more*** probable than **any** collision occurring between <u>any two sets of 256 bits</u> given one billion sets of bits.

  - Thus, anyone worrying about 256-bit collisions is worrying about the wrong thing.

- **WITH** regard for attackers:
  - Resistant to them calculating the past or the future from the present.
  - Algorithmic solutions have this vulnerability if the state can ever be known or determined.
  - Resistant to them acquiring internal state.
  - Resistant to them interfering/modifying with our entropy generation/collection.

The AMOUNT of entropy needed is another crucial factor:

- A TLS web server needs (and consumes) huge amounts.
- A TLS client needs a modest amount.
- A SQRL server needs some, but
- SQRL clients needs almost none.

How MUCH "entropy" is needed?

- "Quantum" hardware always has some upper rate limit
- Software can run at any speed... but...
  - The size of it's "state" limits its output.
    - (I needed the UHEPRNG for the Off the Grid project.)
    - Linear Congruential RNG
    - Counter AES
    - Hash based
  - Perfect Forward & Backward Secrecy
- Hardware seeded software PRNG
- The best possible solution is entirely random.

The Amazing Hash Function

- "Digest" "Compression"
- Avalanche effect: change *any* single bit and every output bit is affected 50% of the time.
- Encryption is NOT lossy:
    - AES is 128 bits in, a *different* 128 bits out. A one-to-one mapping.
    - Since no information is lost, encryption CAN be reversed.
- Hashing is deliberately lossy:
    - Not reversible since information is lost in the output... there's not enough information present to determine what was input.

SQRL's Threat Model

- External attacker:
- Passively observes the result.
- Passively affects the OS/Application boundary.
    - Where does the Application obtain its entropy?
- Actively interferes with the entropy gathering:
    - Blocking the OS's return data.

SQRL's solution: Many diverse streams of noise, each having different characteristics, all continuously pouring into a hash function, mixed and mashed and hashed all together.

SQRL's Entropy Harvester Source Code:
https://www.grc.com/miscfiles/SQRL-Entropy-Harvester.png

- Use many system-level sources of uncertainty:
    - OS high-speed processor clock count.
    - With 100 nanosecond resolution:
        - Time all threads spent in user mode
        - Time all threads spent in the kernel
        - Time the scheduler spend idle
        - Instantaneous snapshot of global memory usage statistics
        - Time this thread spent in: user/kernel/idle and instant created.
        - Time this process spent in: user/kernel/idle and instant created.
    - Slower changing and locally unpredictable data:
        - Current Process ID
        - Current Thread ID
        - Various system handle values
        - The instantaneous X and Y position of the mouse.

- Use the platform's Cryptographic RNG, but don't rely solely upon it.
    - Since the data crosses the process boundary, it could be interfered with.

- Use the chip's new built-in TRNG, but rely solely upon it.
    - RDRAND instruction
    - May not be present, may not be without NSA influence.

- Use the RDTSC / clock / counter:
    - Least significant bits cannot be controlled and cannot be known.
    - Always present.
    - There is SO MUCH GOING ON inside today's micros -- with branch prediction, speculative execution, deep instruction pipelining with out-of-order instruction execution, register renaming, two or three levels of caching with cache hits and misses, etc. The processor's entire instantaneous

- Capture ALL OF THIS 50 times per second on a low-priority thread, thus lots of jitter
- Capture ALL OF THIS for every event the application experiences…
    - Every movement of the mouse over the UI, every button press...