Security Now! #632 - 10-10-17

The DNSSEC Challenge

This week on Security Now!

This week we take a look at a well-handled breach-response at Discus, a rather horrifying mistake Apple made in the implementation of their APFS encryption (and the difficulty to the user of fully cleaning up after it), the famous "robots.txt" file gets a brilliant new companion, somewhat shocking news about Windows XP... or is it?, Firefox EOL for Windows XP support coming next summer, the sage security thought for the day, an update on "The Orville", some closing the loop comments, including a recommendation of the best Security Now series we did in the past... and finally, a look at the challenge of DNSSEC.

Our Picture of the Week

Security News

A well-handled security breach at Discus.
Cory Doctorow for BoingBoing
https://boingboing.net/2017/10/09/salt-your-hashes.html

Five years ago, back in 2012, the Discus commenting service suffered an undetected breach of 17.5 million user accounts.

Troy Hunt, creator of the "Have I Been Pwned" site and service, discovered and disclosed the breach.

Troy was VERY HAPPY with their response... So much so that Troy's blog posting was titled: Disqus Demonstrates How to Do Breach Disclosure Right:
https://www.troyhunt.com/disqus-demonstrates-how-to-do-data-breach-disclosure-right/

QUOTE: "23 hours and 42 minutes from initial private disclosure to @disqus to public notification and impacted accounts proactively protected."

Think about everything that had to happen within this time frame:

- I had to get a response and establish communication
- I had to get the data to them securely (over Australian internet speeds...)
- They had to download and review the data
- They had to establish the legitimacy of the data
- They had to ensure there was no ongoing risk in their system
- They had to invalidate passwords that had been exposed
- They had to contact the impacted users in the previous point
- They had to prepare the communication in the aforementioned disclosure

When I look at how Disqus handled their incident, they ticked so many of the boxes:

- It was easy to report to them (admittedly, my having an existing contact there inevitably made it easier than if I was coming out of the blue)

- They applied urgency, more than I can honestly say I've seen any company do before under similar circumstances

- They disclosed early, earlier than anyone could have reasonably expected (I normally consider 72 hours the "Gold Standard")

- They protected impacted accounts very quickly by resetting the passwords of accounts that had them disclosed

- They were entirely transparent; there was never a moment where I thought they were attempting to spin this in their favour at the expense of the truth

- They provided details - the passwords were salted SHA1 hashes which is not a pretty story to tell in this day and age, but they told it truthfully regardless

- They apologised (it was one of the first things they said); they owned this incident from the outside and didn't attempt to divert blame elsewhere

On the topic of anyone can make a mistake: some are more embarrassing than others: Apple issued an update for its High Sierra desktop operating system last Thursday.

It was the "macOS High Sierra 10.13 Supplemental Update." It repairs two dangerous bugs in High Sierra, both of which exposed user passwords in some way.

In the first case, if you created a new APFS (Apple File System) encrypted volume on High Sierra, and set anything at all as the password hint, then your password was stored as the hint. In plain text. That means anyone could've gotten your password simply by clicking on the "Show Hint" button. If you didn't choose anything as your password hint, you were safe.

The second worrisome bug fixed by the 10.13 Supplement allowed a malicious attacker to extract all your keychain passwords with an unsigned app. Whoops!

If macOS High Sierra shows your password instead of the password hint for an encrypted APFS volume: https://support.apple.com/en-us/HT208168

Take steps to protect your data if you see your password instead of your password hint for an encrypted APFS volume.

Your password might be displayed instead of your password hint if you used the Add APFS Volume command in Disk Utility to create an encrypted APFS volume, and you supplied a password hint.

Changing the password on an affected volume clears the hint but doesn’t affect the underlying encryption keys that protect the data.

Apple recommends that you take these steps to guard the security of your data.

Follow these steps to update macOS High Sierra, and then back up, erase, and restore the encrypted APFS volume.

- Install the macOS High Sierra 10.13 Supplemental Update from the App Store updates page.
- Create an encrypted backup of the data in your affected encrypted APFS volume.
- Open Disk Utility and select the affected encrypted APFS volume in the sidebar.
- Click Unmount to unmount the volume.
- Click Erase.
- When asked, type a name for the volume in the Name field.
- Change Format to APFS.

- Then change Format again to APFS (Encrypted).
- Enter a new password in the dialog. Enter it again to verify the password, and if you'd like to, provide a hint for the encrypted APFS volume. Click Choose.
- Click Erase. You can see the progress of the Erase process.
- Click Done when the process is complete.
- Restore the data that you backed up in Step 2 to the new encrypted APFS volume that you just created.

Robots.txt --now--> Security.txt

- Long-time web developers are all familiar with the "robots.txt" file located in the root of a web site.
- <Whatszit?>
- Now: security.txt -- to provide a uniform discoverable means for reporting vulnerabilities to a site.
- Brilliant!
- https://github.com/securitytxt/security-txt

Contact:
Add an address that researchers MAY use for reporting security issues. The value can be an email address, a phone number and/or a security page with more information. The "Contact:" directive MUST always be present in a security.txt file.

Contact: security@example.com
Contact: +1-201-555-0123
Contact: https://example.com/security

Encryption:
This directive allows you to add your key for encrypted communication. You MUST NOT directly add your PGP key. The value MUST be a link to a page which contains your key. Keys SHOULD be loaded over HTTPS.

Encryption: https://example.com/pgp-key.txt

Disclosure:
Specify your disclosure policy. This directive MUST be a disclosure type. The "Full" value stands for full disclosure, "Partial" for partial disclosure and "None" means you do not want to disclose reports after the issue has been resolved. The presence of a disclosure field is NOT permission to disclose vulnerabilities and explicit permission MUST be saught where possible.

Disclosure: Full

Acknowledgement:
This directive allows you to link to a page where security researchers are recognized for their reports.

Acknowledgement: https://example.com/hall-of-fame.html

Example

```
# Our security address
Contact: security@example.com
Encryption: https://example.com/pgp-key.txt
Disclosure: Full
```

Old OSes are [oh so] very hard to kill!

What!?!: "Announcing support for TLS 1.1 and TLS 1.2 in XP POSReady 2009"

https://cloudblogs.microsoft.com/microsoftsecure/2017/10/05/announcing-support-for-tls-1-1-and-tls-1-2-in-xp-posready-2009/
Microsoft have backported TLS 1.1 and 1.2 support to XP. Will be appearing in Windows Update channels.

Firefox to End-Of-Life support for WinXP next June (2018)
https://blog.mozilla.org/futurereleases/2017/10/04/firefox-support-for-windows-xp-and-vista/

<quote> Last year we announced that Windows XP and Vista users would be automatically moved to the Firefox Extended Support Release (ESR), ensuring them continued updates until at least September, 2017.

Today we are announcing June 2018 as the final end of life date for Firefox support on Windows XP and Vista. As one of the few browsers that continues to support Windows XP and Vista, Firefox users on these platforms can expect security updates until that date. Users do not need to take additional action to receive those updates.

Sage Security Thought for the Day:

- Arthur C. Clarke: "Any sufficiently advanced technology is indistinguishable from magic."
- Matthew Green: "Sufficiently advanced incompetence is indistinguishable from malice."

Miscellany

The Orville:

- Dan Edwards (@dedwards66)
  First episode of The Orville was a train wreck!
  Episode 2 is much better and 3 and 4 are fantastic. Very Roddenberry-esk!

- Al Spaulding (@alibertarian)
  "The Orville" had eye-rolls at first. I pushed thru and episode 3 turned out to be good, with a Roddenberry worthy twist at the end.

- Adam van Kuik (@avankuik)
  You said The Orville wasn't your cup o tea after viewing few minutes of the 1st episode. You may start to enjoy it from ep2 on. I find The Orville more drama than comedy. I think episodes 3-5 would hold your interest, but I could be wrong.

Leo and iOS v11 -- buggy, yes??

An "Asphalt" battery?
No... but James Tour and other researchers at Rice University have discovered that a touch of asphalt may be the secret to high-capacity lithium metal batteries that charge 10 to 20 times faster than commercial lithium-ion batteries.

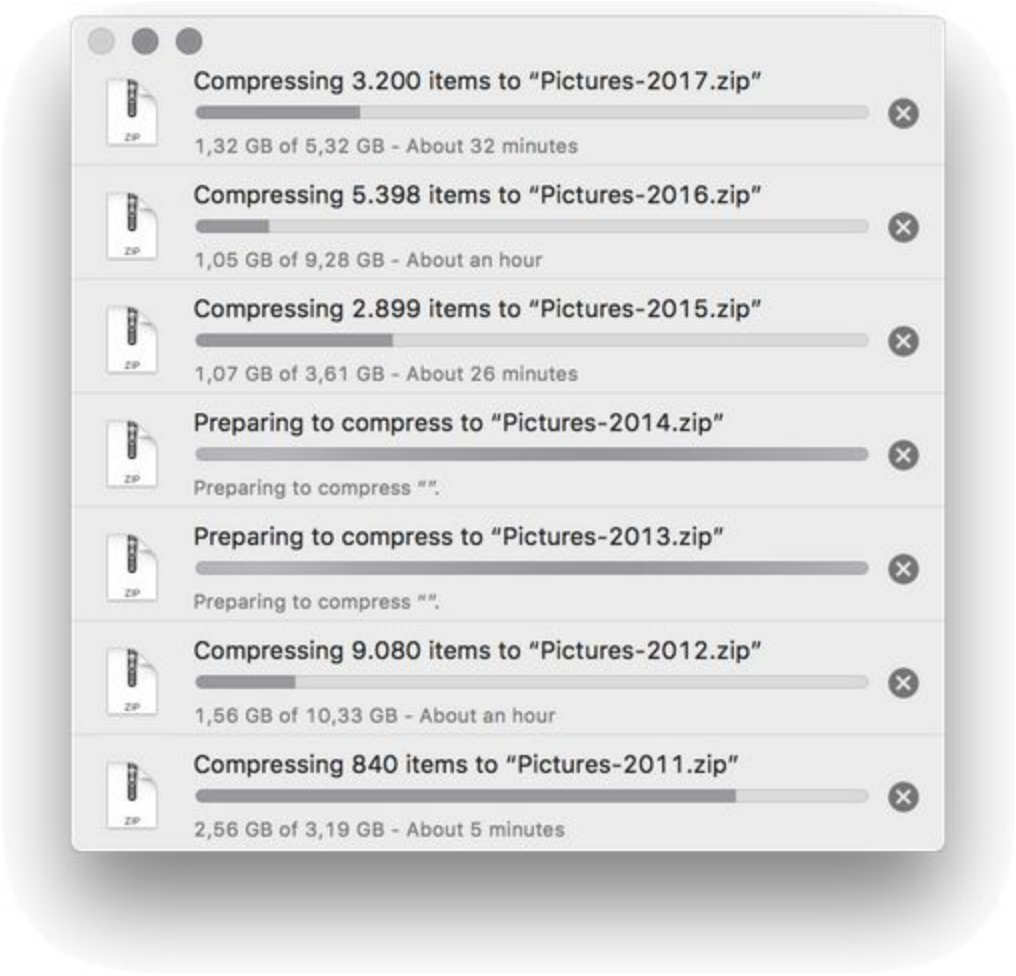
Tour was quoted in press coverage: “The capacity of these batteries is enormous, but what is equally remarkable is that we can bring them from zero charge to full charge in five minutes, rather than the typical two hours or more needed with other batteries.”

Testing revealed another significant benefit: The asphalt additive mitigated the formation of lithium dendrites which are those mossy deposits which invade a lithium ion battery’s electrolyte and... if they extend far enough, short-circuit the anode and cathode to cause the battery to fail, catch fire or explode. The asphalt-derived carbon prevents any dendrite formation.

But like all of the other power storage stuff we have covered here through the years... it's still stuck in the lab. What's worse, though, is that if asphalt batteries DO someday materialize we'll likely be needing to tolerate articles titled: "Paving the way to a better battery!"

SpinRite

Floris (@Floris) (edited a bit for clarity)
I dug out my very very old @SGgrc executable to try to fix a broken drive. Hooked it to an old WinXP machine. After ONLY 4 hours of constant rattling noises it made massive progress finding tens of gigs and tens of thousands of pix <3.



Tyson Clugy (@tyson0016)
PC locked up several times this week. Wouldn’t boot today. Less than an hour with #spinrite and it’s made a full recovery! Thanks!

NeutronJon (@NeutronJon)
Just bought SpinRite and wanted to give a shout out to a tutorial that helped me use it on a Mac: “Running SpinRite 6.0 on MacOS”
https://kevinstreet.co.uk/2017/09/11/running-spinrite-6-0-on-macos/
VERY nice walk through instructions!

Closing The Loop

Josh Freeman (@joshfreemanftw)
Been listening for years but what are some of the key/great episodes from the early years of SecurityNow that I should listen to?

#233: Let's Design a Computer (part 1)
“To understand the advances made during 50 years of computer evolution, we need to understand computers 50 years ago. In this first installment of a new Security Now series, we design a 50 year old computer. In future weeks, we will trace the factors that shaped their design during the four decades that followed.”

#235: Machine Language
#237: Indirection: The Power of Pointers
#239: Stacks, Registers & Recursion
#241: Hardware Interrupts
#247: The "Multi"-verse (multi-threading, multi-processing, multi-tasking, multi-core)
#250: Operating Systems
#252: RISCy Business
#254: What We'll Do for Speed

Paradigm Concepts (@pdxpc)
Steve, have you had a chance to review Bitdefender box? It describes itself as a total home protection device for unlimited iot devices and lets you roam using your home connection through VPN to you home) https://www.bitdefender.com/box/

Simon Zerafa (@SimonZerafa)
@SGgrc A lot of Netgear updates this week! ??
http://www.netgear.com/about/security/

Simon also forwarded this clear delineation of Identity and Authentication:
Wendy Nather (@wendynather)
Identifiers (telling people apart) are designed to be public; authenticators (proving who you are) should be secret. SSNs are used as both.

Kyle Hardin (@kylehardin_AK)
Excellent show. What do you think of RAID 10 as an alternative to RAID 6 regarding drive failure during rebuild?

VipX1?? (@vipx1)
What is best @Synology NAS for regular home, OS image backups using @Acronis?

--and--

SgtWilko (@sgtwilko)
Hiya, I know you've mentioned it on @SecurityNow, but I can't find it. Which NAS do you use/recommend?

I'm biased. I explored FreeNAS on UNIX and saw that it was only a thin gloss coating over the OS. And I would always deal with the raw underlying OS. So when I'm back to work on moving the SQRL forums over to their new server, I'm going to setup a big storage platform using ZFS. ZFS is *SO* impressive that's all that's needed. And the FreeBSD OS supports every connection protocol.

David Lemire (@dlemire60)
re: huge root cert trust stores: any practical way to clear out, then add back individually what's actually needed by user?

What someone COULD write (I could, but no one wants me to instead of finishing SQRL and then getting back to SpinRite) is an auditor that watches TLS connections and compiles a growing list of root certs that are actually used. After a year, any user will have likely used any that they are going to, and then those not used could be sequestered and only brought back when needed.

The DNSSEC Challenge

Domain **N**ame **S**ystem **SEC**urity Extensions

RFC 2065 - "Domain Name System Security Extensions" -- Dated: January 1997
(20 years ago)

Abstract

The Domain Name System (DNS) has become a critical operational part of the Internet infrastructure yet it has no strong security mechanisms to assure data integrity or authentication. Extensions to the DNS are described that provide these services to security aware resolvers or applications through the use of cryptographic digital signatures. These digital signatures are included in secured zones as resource records. Security can still be provided even through non-security aware DNS servers in many cases.

The extensions also provide for the storage of authenticated public keys in the DNS. This storage of keys can support general public key distribution service as well as DNS security. The stored keys enable security aware resolvers to learn the authenticating key of zones in addition to those for which they are initially configured. Keys associated with DNS names can be retrieved to support other protocols. Provision is made for a variety of key types and algorithms.

In addition, the security extensions provide for the optional authentication of DNS protocol transactions.

The latest RFCs are the set of 4033/4034/4035, which...
Obsoletes: 2535, 3008, 3090, 3445, 3655, 3658, 3755, 3757, 3845
Updates: 1034, 1035, 2136, 2181, 2308, 3225, 3007, 3597, 3226

The short version is that they got is badly wrong the first time. The original design required the use of a great deal of back and forth messaging and it quickly became clear that it would not and could not "scale" to the size of the entire Internet. So it was scrapped and the work began again.

An example of a valuable feature that DNSSEC makes possible is DANE - "**D**NS-based **A**uthentication of **N**amed **E**ntities"

"DANE" -- whenever it someday happens -- would allow domains to publish their own trusted TLS web server public keys rather than needing them to be signed by root CAs.

What's the problem with DNSSEC?
Whereas the original DNS was elegantly designed and quite lightweight, adding true security to DNS was -- and is -- not easy. DNSSEC is complicated. It requires a rather HUGE addition to the existing simple and lightweight DNS system.

And a perfect analogy is TCP: It's a beautiful, elegant, simple and robust system for INSECURELY interconnecting to remote Internet nodes. But if we want to secure that? OMG! After a long series of SSL's and now TLS's, we're STILL working to get it right. Security is difficult.

And, DNSSEC just suffered another setback...

TOMORROW, October 11th, had long been the planned ICANN KSK (Key Signing Key) rollover for DNSSEC… but...

> ICANN's announcement states the the KSK rollover is being delayed…
> *…because some recently obtained data shows that a significant number of resolvers used by Internet Service Providers (ISPs) and Network Operators are not yet ready for the Key Rollover. The availability of this new data is due to a very recent DNS protocol feature that adds the ability for a resolver to report back to the root servers which keys it has configured.*

Translation: After 20 years DNSSEC is still be changed, messed around with, and not settled.

DNS Operations, Analysis, and Research Center (*DNS-OARC*) is a DNS organizational body. And Friday before last, during the DNS-OARC Annual General Meeting, Verisign's Duane Wessels presented the result of their passive monitoring of DNS and DNSSEC in his talk, titled:

A Look at RFC 8145 Trust Anchor Signaling for the 2017 KSK Rollover
RFC 8145 ("Signaling Trust Anchor Knowledge") was published in April 2017. This RFC describes how recursive name servers can signal, to authoritative servers, the trust anchors that they have configured for Domain Name System Security Extensions (DNSSEC) validation. Shortly after its publication, both Unbound and BIND implemented the specification. As organizations begin to deploy the new software versions, some of this "key tag data" is now appearing in queries to the root name servers.

This is useful data for Key Signing Key (KSK) rollovers, and especially for the root. Since the feature is very new, the number of recursive name servers providing data is not as significant as one might like for the upcoming root KSK rollover. Even so, it will be interesting to look at the data. By examining this data we can understand whether or not the technique works and hopefully inspire further adoption in advance of future KSK rollovers.

The most obvious revelation from Duane's methodology was that published standards are one thing, but deployed and online implementations, where the bits hit the packets, is what ultimately matter. ICANN soberly decided to kill the long-planned rollover from the 2010 Key Signing Keys to the 2017 Key Signing Keys because the fragmentary data they were seeing revealed that the Internet was still not ready.
DNSSEC is finally beginning to happen, and the LAST thing we need at this point is to stumble and break an important system that is finally beginning to come to life.

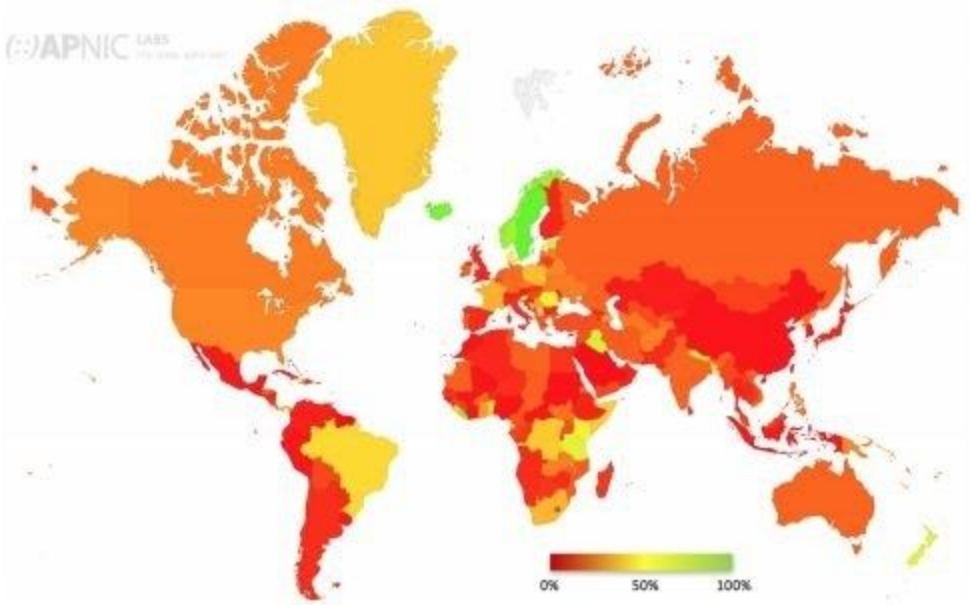
Signing of domains with DNSSEC:

o 89% of top-level domains (TLDs) zones signed.

- ~47% of country-code TLDs (ccTLDs) signed.

o Second-level domains (SLDs) vary widely:

- Over 2.5 million .nl domains signed (~45%) (Netherlands). [1]
- ~88% of measured zones in .gov are signed.
- Over 50% of .cz (Czech Republic) domains signed.
- ~24% of .br domains signed (Brazil). [2]
- While only about 0.5% of zones in .com are signed, that percentage represents ~600,000 zones.



o The major DNS authoritative server software and libraries support DNSSEC and have several years of deployment experience.

o Management tools have started to come online to assist deployment, e.g., key deployment and rollover.

o Encryption algorithms and key lengths

- The overwhelming majority of TLDs utilize RSA/SHA-256 with 2048 bit keys for the Key Signing Key (KSK) and 1024 bit keys for the Zone Signing Key (ZSK).

- A significant number of zones measured still utilize SHA-1.
- Utilization of ECDSA is at 5% and growing.

o In 2016, the Zone Signing Key (ZSK) for the Root Zone was successfully migrated to a 2048-bit RSA key.

Validation

o All major DNS recursive resolvers support DNSSEC validation.

o ~80% of clients request DNSSEC digital signature records in their DNS queries (per APNIC research).

o 26% of end user environments use DNSSEC-validating resolvers, but also pass queries to non-validating resolvers if validation results in a validation failure.

o Although only ~14% of clients globally exclusively use DNSSEC-validating DNS resolvers, the numbers vary greatly between regions and countries.

- Over 50% of clients in most Scandinavian countries exclusively use DNSSEC-validating DNS resolvers.

o A large ISP enabling DNSSEC validation on its recursive resolvers can have a big impact on a country's utilization numbers (e.g. Comcast in USA, Claro in Brazil).
o Google Public DNS (PDNS) service support for DNSSEC validation makes validation available globally (where allowed by law).

Applications/Services

o Libraries, APIs and tools are becoming available to enable DNSSEC use by application developers.

o DANE

- Utilization of DANE is relatively low, but growing.
- Libraries, APIs and tools are becoming available.
- Most prominent utilization of DANE is securing email transfers between email servers, led by German email providers.

"It's Complicated"

DNS is a global directory for retrieving tagged (named) resource records (RRs) given a hierarchical domain name.

The classic 'A' (address) record for www.grc.com returns: 4.79.142.202. Anyone in the world can look that up. The addition of IPv6 necessitated the creation of the AAAA record, which is the same thing for IPv6.

We also have MX (mail exchange), PTR (pointer), NS (nameserver), CNAME (canonical name) and many many others.

Under DNSSEC, these various resource records must now, themselves, be signed with a new RRSIG (resource record sig) record.
And, a new DNSKEY record contains the public key that a DNS resolver querying a DNSSEC-enabled server uses to verify the RRSIG signatures.

And if this wasn't already complicated enough… as we know, any practical system also requires us to manage -- meaning provide a means of smoothly retiring and replacing -- all of this cryptographic material.

So we wind up with Zone Signing Keys (ZSK), Key Signing Keys (KSK), Delegation Signer (DS) records.

In Conclusion…