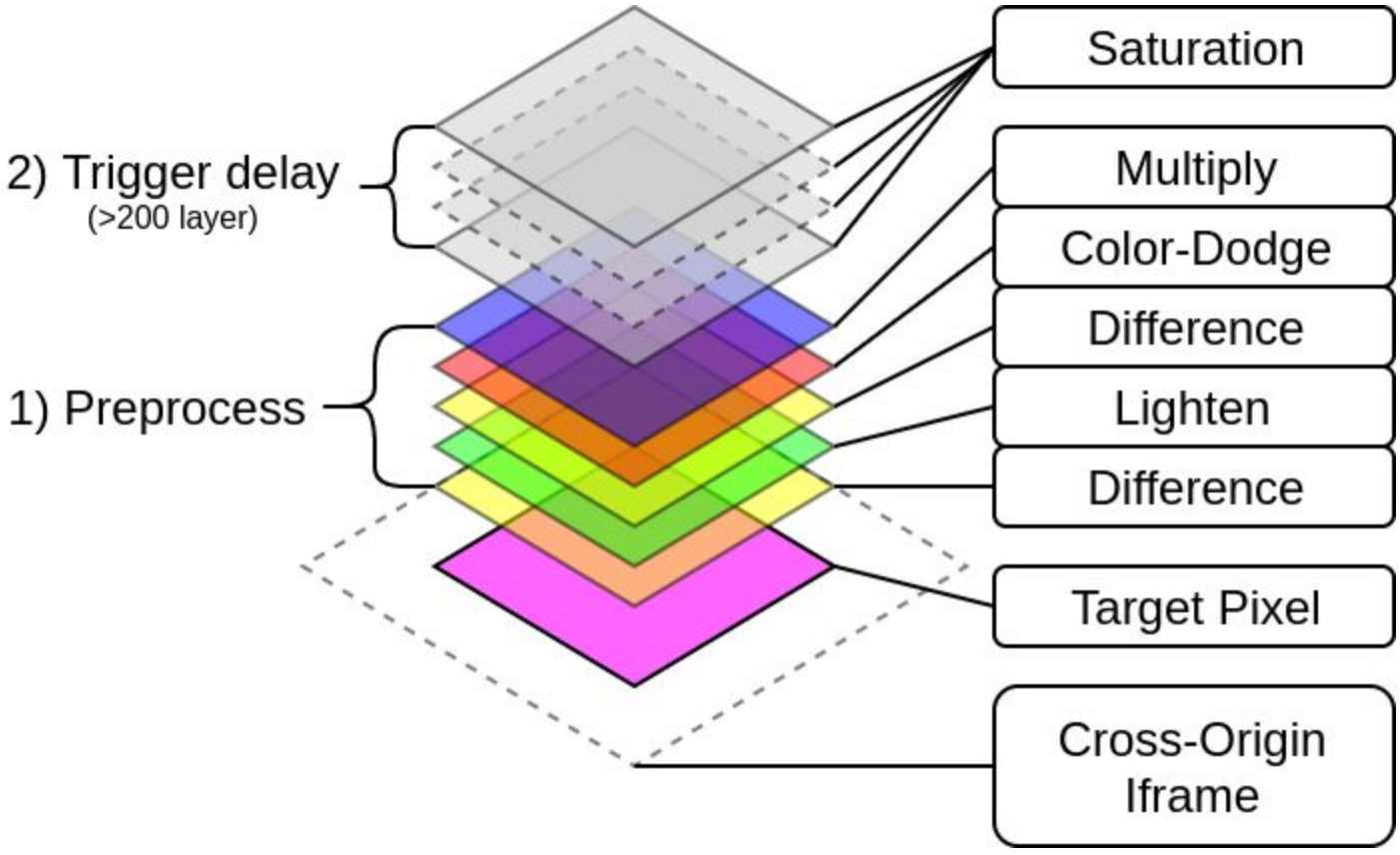
Security Now! #666 - 06-05-18

Certificate Transparency

This week on Security Now!

This week we discuss yesterday's further good privacy news from Apple, the continuation of VPNFilter, an extremely clever web browser cross-site information leakage side-channel attack, Microsoft Research's fork of OpenVPN for security in a post-quantum world, Microsoft drops the ball on a 0-day remote code execution vulnerability in JScript, Valve finally patches a longstanding and very potent RCE vulnerability, Redis caching servers continue to be in serious trouble, a previously patched IE 0-day continues to find victims, Google's latest Chrome browser has removed support for HTTP public key pinning (HPKP), and... what is "Certificate Transparency" and why do we need it?

Our Picture of the Week



Security News

Apple's Safari privacy improvements.
Apple's updated Safari browser in macOS Mojave and iOS 12 will include new default anti-tracking behavior.

ALL user tracking is a misuse of technologies which was designed for other purposes.

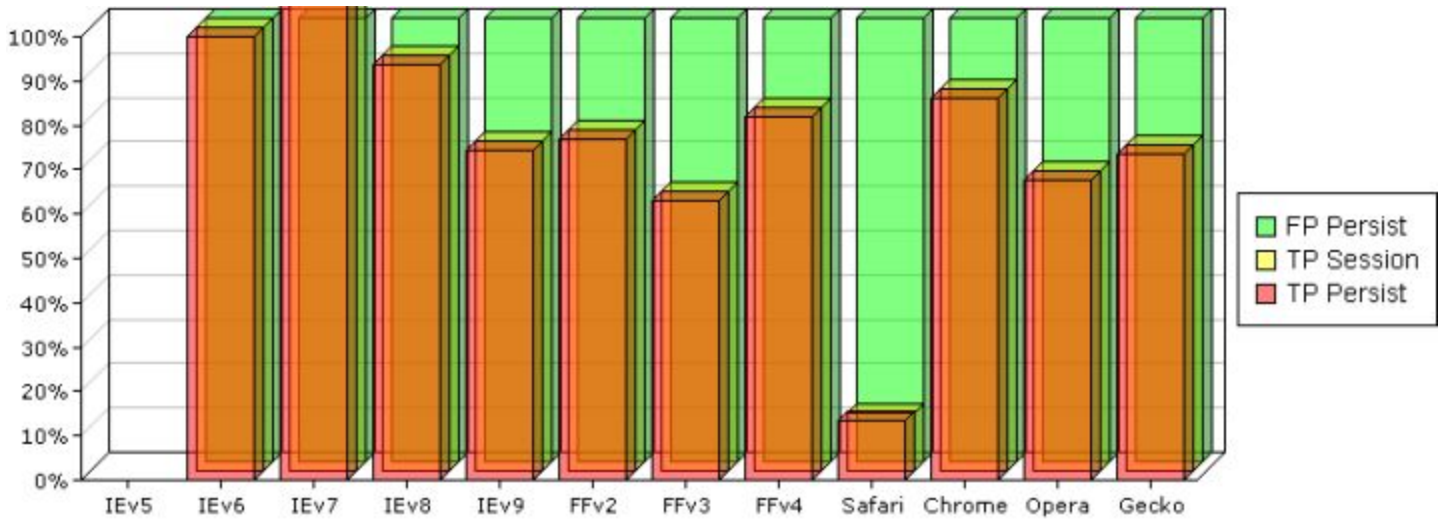
Web browser cookies were never designed to be used for tracking. They were designed to create stateful a relationship with individual websites..

Browser header fingerprinting was never designed to allow deanonymizing individuals who implicitly assume that unless they explicitly and deliberately share their information they are anonymous.

Remember that tracking and advertising are NOT the same. Annoying and abusive advertising is certainly a problem. But ad-blocking is related to, but separate from, tracking. Advertising networks WANT to track us, but they do not need to in order to exist.

And suggesting that blocking these technologies will damage a industry that depends upon them -- first of all is not true -- and that also has it exactly backwards: If blocking a technology from abuse, damages an industry that has grown to depend upon that abuse, then it is an industry built on the abuse of individuals' privacy and it deserves whatever happens to it as individuals take back their implicit privacy.

Apple's Safari has always been the lone browser which disables 3rd-party cookies by default:



Apple calls the new heightened pro-privacy measures "Intelligent Tracking Prevention 2.0." During yesterday's WWDC opening presentation, Apple's Craig Federighi showed a Safari page with a popup notification reading "Do you want to allow 'facebook.com' to use cookies and website data while browsing 'blabbermouth.net'? This will allow 'facebook.com' to track your activity."

(In response to this, Facebook's chief information security officer (CISO), Alex Stamos tweeted that it didn't appear as though the new Safari would be blocking tracking pixels or JavaScript components.)

But Will Strafach, an iOS security researcher and the president of Sudo Security Group was quoted by Wired Magazine as saying that "The consent popups will be a big deal to people. It's more visual, so you know that they are attempting to track you versus it just happening in the background silently."

As usual, I'm interested in the details, and we don't have those yet. We know that something will trigger these pop-up notifications, which will be interesting to experience. And we know from what Craig said, that Safari will be stripping and unifying the details provided by its browser headers to explicitly thwart browser fingerprinting by causing individual Safari browsers to all appear identical.

The attackers behind VPNFilter are continuing to exploit
https://jask.com/from-russia-with-love/

JASK and GreyNoise Intelligence (GNI) have been watching the 'Net in the wake of the FBI's takedown of the the VPNFilter botnet after acquiring the ToKnowAll.com domain and asking the world to reboot its routers.

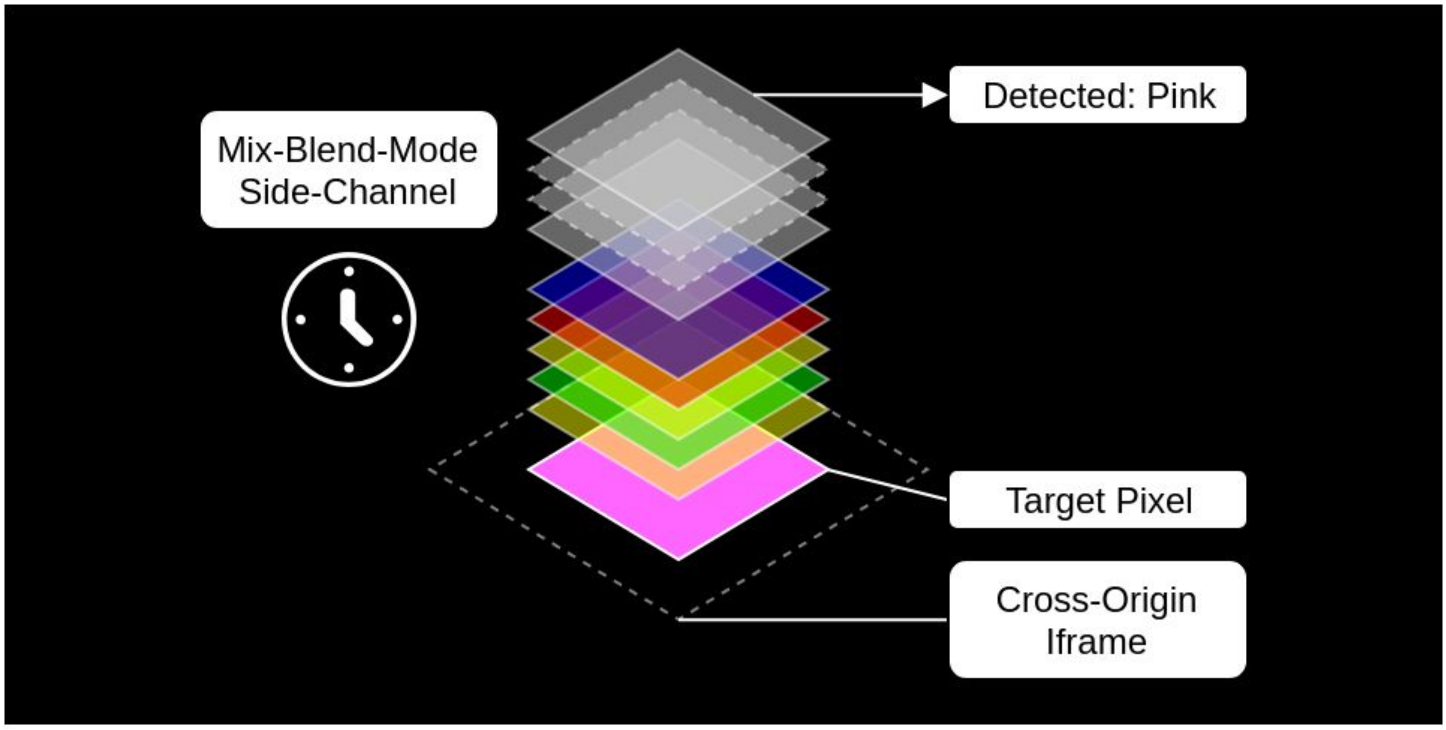
Since that time, a handful of IPs have been actively scanning port 2000 of Ukrainian IP space for MicroTik routers:

138.186.2.250	Brazil, Andradina	Noroestecom Telecomunicacoes Ltda
178.78.13.69,	Russia, Volgograd	LLC Columbia Telecom
178.78.6.224	Russia, Volgograd	LLC Columbia Telecom
187.85.58.107	Brazil, Magalhães	LENCO TECNOLOGIA LTDA
192.157.214.6	United States, Los Angeles	Enzu Inc
77.45.243.188	Russia, Lisk	Regional multiservice network access
88.213.189.253	Switzerland, Schaffhausen	sasag Kabelkommunikation AG
9.110.0.5	United States,	IBM

Various experts in the security industry have suggested that those behind VPNFilter have not been fazed by this setback. So it's going to be interesting to see how this develops.

An extremely clever hack uses CSS to determine site visitor Facebook names. Unfortunately, ArsTechnica run with the headline: "EXPOSED — Chrome and Firefox leaks let sites steal visitors' Facebook names, profile pics." We'll see in a moment how unfair that headline was.

https://www.evonide.com/side-channel-attacking-browsers-through-css3-features/



Same-Origin Policy -- crucial for maintaining security and isolation.

Several researchers independently discovered a side-channel vulnerability in browser implementations of the CSS3 feature "mix-blend-mode" which they were able to cleverly use to leak visual content from cross-origin iframes.

They have demonstrated the impact of this vulnerability by showing how a clever website could de-anonymize a Facebook user in 20 seconds. And, given time, their Facebook profile picture, username and likes could be determined.

This vulnerability affected Chrome and Firefox which fully supported the required CSS3 features.

CSS3 provides for multiple layers and inter-layer "blend modes" which arithmetically combine pixels among layers with operations such as multiply, screen, overlay, darken, lighten, color-dodge, and more.

https://css-tricks.com/basics-css-blend-modes/

But -- and here's the hook: These operations are not guaranteed to be accomplished in "constant time." There are some modes where different code paths are taken based upon the pixel content of the image.

So... if a malicious or naughty web page were to bring up someone's Facebook page in an iframe -- even though iframe's digital textual content is EXPLICITLY sandboxed and isolated from its containing page -- the iframe's IMAGE is not similarly protected... and its image can be overlayed with CSS image processing layers controlled by the hosting containing page.

Since the timing of layered operations depends upon the pixel values of the images, it is possible -- and has been demonstrated -- for the containing page to "scan" the image in the iframe with a probe pixel while observing how long the various operations take to complete.

Skia Graphics Library
Skia is an open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms. It serves as the graphics engine for Google Chrome and Chrome OS, Android, Mozilla Firefox and Firefox OS, and many other products.

Skia is sponsored and managed by Google, but is available for use by anyone under the BSD Free Software License. While engineering of the core components is done by the Skia development team, we consider contributions from any source.

Xfermode_Luminosity	80128.68 -> 27189.83
Xfermode_Luminosity_aa	18001.46 -> 14449.67
Xfermode_Color	183520.09 -> 27448.49
Xfermode_Color_aa	34257.75 -> 14694.85
Xfermode_Saturation	80323.96 -> 36449.01
Xfermode_Saturation_aa	19605.81 -> 18576.62
Xfermode_Hue	125866.82 -> 36611.03
Xfermode_Hue_aa	25318.00 -> 18489.33

Microsoft's research group have forked OpenVPN for Post Quantum Crypto
https://github.com/Microsoft/PQCrypto-VPN

"PQ" Post Quantum

- Frodo: a key exchange protocol based on the learning with errors problem

- SIKE: a key exchange protocol based on Supersingular Isogeny Diffie-Hellman

  (An isogeny is a transformation which maps a point on one elliptic curve to a point on another possibly-different elliptic curve and it is believed that this will remain an intractable problem even post-quantum.)

- Picnic: a signature algorithm using symmetric-key primitives and non-interactive zero-knowledge proofs

  “Picnic” is the code name for a post-quantum digital signature algorithm. Picnic is developed in collaboration with researchers and engineers from Aarhus University, AIT Austrian Institute of Technology GmbH, Graz University of Technology, Microsoft Research, Princeton University, and the Technical University of Denmark.

Unlike most other public-key cryptography, Picnic isn't based on hard problems from number theory. Instead, it uses what is called a zero-knowledge proof – where Alice can convince Bob that she knows a secret, without disclosing the secret itself. Picnic uses this concept together with symmetric cryptography, hash functions, and block ciphers, to create a unique signature scheme. The hard problems Picnic relies on for security relate only to hash functions and block ciphers, that are thought to be secure against quantum attacks.

"Shor's algorithm" from 1994, named after mathematician Peter Shor, is one of the things that has people worried since it is a quantum algorithm for a theoretical quantum computer which solves the "integer factorization" hard problem. Given any large integer N, it can determine its prime factors -- thus classic RSA asymmetric public key crypto, which relies upon the impracticality of factoring the product of large primes -- collapses in a world where that has become possible.

And Elliptic Curve crypto, in its current form, is also in trouble since Shor's algorithm can slice right through the discrete log problem that protects elliptic curve point multiplication.

Symmetric encryption is still believed to be fine -- because symmetric encryption does not depend upon a "hard problem" for its security. But it will require a doubling of its key length. So whereas AES with a 128-bit key is super strong today, post-quantum AES will need 256-bit keys for equivalent (super strong) security.

And, similarly, our existing larger hash functions SHA-2 at 256 and SHA-3 at 256 and larger are already fine.

Understand that we are not in danger TODAY, but that post-quantum research takes time and that research is going on now. So if anyone manages to create a quantum computer of more than toy-level complexity -- which is where we are today -- we'll have replacements ready to do.

For the truly paranoid, Microsoft's OpenVPN fork is working and running under Linux and Windows 10.

When 4 months is not enough time to fix a bad bug...

After discovering a bad RCE bug in Microsoft's JScript, Dmitri Kaslov of Telspace Systems reported his discovery to Trend Micro's Zero-Day Initiative (ZDI) group.

The ZDI group responsibly notified Microsoft on January 23rd and received a same-day acknowledgement.

Then a week ago, last Tuesday on May 29th -- more than four months later -- the ZDI group decided to hold Microsoft accountable for not patching this still-unpatched serious remote code execution bug in their JScript engine.

ZDI's post, dated May 29th, is titled: "(0Day) Microsoft Windows JScript Error Object Use-After-Free Remote Code Execution Vulnerability"
And begins: "This vulnerability allows remote attackers to execute arbitrary code on vulnerable

installations of Microsoft Windows. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file.

(In other words, just allowing Microsoft's JScript engine to interpret JavaScript code receive from a visit to any website.)

The specific flaw exists within the handling of Error objects in JScript. By performing actions in script, an attacker can cause a pointer to be reused after it has been freed. An attacker can leverage this vulnerability to execute code under the context of the current process.

This vulnerability is being disclosed publicly without a patch in accordance with the ZDI 120 day deadline.

- 01/23/18 - ZDI sent the vulnerability report to the vendor
- 01/23/18 - The vendor acknowledged and provided a case number
- 04/23/18 - The vendor replied that they were having difficulty reproducing the issue report without POC
- 04/24/18 - ZDI confirmed the POC was sent with the original and sent it again
- 05/01/18 - The vendor acknowledged receipt of the POC
- 05/08/18 - The vendor requested an extension
- 05/18/18 - ZDI replied "We have verified that we sent the POC with the original. The report will 0-day on May 29."

Mitigation:
Given the nature of the vulnerability the only salient mitigation strategy is to restrict interaction with the application to trusted files.

Since TODAY is the 1st Tuesday of June and next Tuesday the 12th will be the 2nd Tuesday, Microsoft will hopefully have this problem patched. However, we know a company releasing a patch is not equivalent to

Meanwhile, another IE 0-Day patched last month is being actively exploited
The recently patched vulnerability CVE-2018-8174 which affects Microsoft's VBScript in IE and Office has been added to the RIG exploit kit. After the patch's publication, both Kaspersky Labs and Malwarebytes disclosed details of the 0-day's exploitation chain...

https://blog.malwarebytes.com/threat-analysis/2018/05/internet-explorer-zero-day-browser-attack/
https://securelist.com/root-cause-analysis-of-cve-2018-8174/85486/

The infection chain is:

- A victim receives a malicious Microsoft Word document.
- After opening the malicious document, a second stage of the exploit is downloaded; an HTML page containing VBScript code.
- The VBScript code triggers a Use After Free (UAF) vulnerability and executes shellcode.

Valve Patches Security Bug That Existed in Steam Client for the Past Ten Years
https://www.bleepingcomputer.com/news/security/valve-patches-security-bug-that-existed-in-steam-client-for-the-past-ten-years/

Steam wrote: "Fixed a crash when packets in a UDP connection were malformed in a particular way. Thanks to Tom Court from Context Information Security for reporting this issue."

In Tom's blog posting titled "Frag Grenade! A Remote Code Execution Vulnerability in the Steam Client", he details the story behind a bug which had existed in the Steam client for at least the last ten years, and until last July would have resulted in remote code execution (RCE) in all 15 million active clients. (It was still possible after July, just much more difficult due to Steam's finally adding ASLR to their system.)

Steam has its own UDP-packet based, connection-oriented protocol.

UDP packets are designed to be fragmented in transit and reassembled upon arrival.

Normally a "datagram" is a single packet, but a single datagram might get broken up.

So UDP packets contain a packet length field and a total reassembled datagram length.

The bug was caused by the absence of a simple check to ensure that, for the first packet of a fragmented datagram sequence, the specified packet length was less than or equal to the total datagram length...as it always should be.

That check WAS present for all subsequent packets carrying successive fragments of the datagram. But it was missing from the handler for the first packet.

As Tom wrote: Without additional information-leaking bugs, heap corruptions on modern operating systems are notoriously difficult to control for remote code execution. In this case, however, thanks to Steam’s custom memory allocator and (until last July) no ASLR on the steamclient.dll binary, this bug COULD have been used as the basis for a highly reliable exploit.

~75% of Open Redis Servers Are Infected With Malware
https://www.bleepingcomputer.com/news/security/around-75-percent-of-open-redis-servers-are-infected-with-malware/

Redis servers are a big RAM-based indexed data store intended to be used within an organization and within the organization's internal intranet. As a consequence, Redis servers have assumed to be within an already-protected network environment and have NO network or authentication security enabled by default. They are wide open.

So, naturally, although it was never their designers' intent, many thousands of them -- somewhere around 72,000 of them, to give it a number -- are sitting happily out on the public Internet for anyone to horse around with. <sigh>

If all of this sounds vaguely familiar it's because we have touched upon these servers recently in connection with the Redis-based "WannaMine" botnet which has been merrily mining cryptocurrency on many tends of thousands of those open Redis servers.

Imperva Security has been running a network of Redis honeypot servers in order to observe and characterize the behavior of the attacks against them. And as a result, Imperva discovered that these Redis servers were having SSH (Secure Shell) authentication keys installed so that persistent future access could be obtained if the servers were ever secured in the future.

After developing a fingerprint of infection, they scanned the Internet and found that over 75% of the 72,000 publicly available Redis servers have have SSH keys known to be associated with malware botnets installed onto them.

Given today's target-rich environment, it's an interesting time to be a malicious blackhat hacker. There are certainly no lack of servers and routers and other exposed Internet-connected widgets to attack and screw around with.

I'm still struck by how bizarre today's climate feels. Just 10 years ago this =WAS= the stuff of science fiction and cyber thrillers. Now it's just episode #666 of Security Now! :(

“Deprecations and removals in Chrome 67”
https://developers.google.com/web/updates/2018/04/chrome-67-deps-rems

Deprecate HTTP-Based Public Key Pinning
HTTP-Based Public Key Pinning (HPKP) was intended to allow websites to send an HTTP header that pins one or more of the public keys present in the site's certificate chain.

But that never happened. Basically, no one ever used it.
Adoption was something like 0.04% of websites (1 in 2,500)

As Google wrote: "It has very low adoption, and although it provides security against certificate mis-issuance, it also creates risks of denial of service and hostile pinning."

HPKP's promises and pitfalls.

<<explain>>

To defend against certificate misissuance, web developers should use the Expect-CT header, including its reporting function. Expect-CT is safer than HPKP due to the flexibility it gives site operators to recover from configuration errors, and due to the built-in support offered by a number of certificate authorities.

Miscellany

Intel Crosspoint "Optane"

SpinRite

Gary Foard in England
Subject: Spinrite
Date: 01 Jun 2018 14:47:31

Hi Steve

I’m a big fan of your show and happy Spinrite owner. However as your next episode is 666 I think it is time to ask my awkward question. I recently stumbled upon an old screenshot of Norton’s Disk Doctor and I was surprised to see how similar one of the screen images look compared to the main Graphic Status Display of Spinrite.

Although I’ve followed your Security Now show for many years now I don’t really know what your personal background/time-line/progression/education is, all I know is I see a sincere, experienced I.T. chap who does G.A.S. (give a shit as you once said). I guess that is why we all keep listening.

So did you:

a, Work for or with Norton at some point?
b, Did DOS based graphics dictate the identical look?
c, Something else?

Regards, Gary Foard

Introduction to Certificate Transparency

CA's / Root certs / certs issued by CAs are digitally signed.

Misissuance / Revocation / Fraudulent issuance.

http://www.certificate-transparency.org/how-ct-works
Certificate Transparency adds three new functional components to the current SSL certificate system:

Certificate logs
Certificate monitors
Certificate auditors

Google writes: These functional components represent discrete software modules that provide supplemental monitoring and auditing services. They are not a replacement for, or an alternative to, the current SSL certificate system. Indeed, these components do not change the fundamental chain-of-trust model that lets clients validate a domain and establish a secure connection with a server. Instead, these components augment the chain-of-trust model by providing support for public oversight and scrutiny of the entire SSL certificate system.

Basic Log Features
At the center of the Certificate Transparency system are certificate logs. A certificate log is a network service that maintains a record of SSL certificates. Certificate logs have three important qualities:

- They're append-only. Certificates can only be added to a log; certificates cannot be deleted, modified, or retroactively inserted into a log.

- They're cryptographically assured. Logs use a cryptographic mechanism known as Merkle Tree Hashes to prevent tampering and misbehavior.

- They're publicly auditable. Anyone can query a log and verify that it's well behaved, or verify that an SSL certificate has been legitimately appended to the log.

Google writes: "The number of logs does not need to be large: there need to be enough logs so that log failures or temporary outages are not a problem, but not so many that they become difficult to monitor--say, more than 10 but much less than 1000. Each log operates independently of the other logs (that is, there's no automatic replication among the logs).

The append-only nature of a log allows it to use a special type of cryptographic hash to prove that it's not corrupt and that the log operator has not deleted or modified any certificates in the log. This special hash--known as a Merkle Tree Hash--also makes it possible for auditors to detect whether someone has forked a log or inserted back-dated certificates into a log.

Every certificate log must publicly advertise its URL and its public key (among other things). Anyone can interact with a log via HTTPS GET and POST messages.

https://www.digicert.com/blog/certificate-transparency-faqs/

Certificate Transparency: FAQs

On February 1, 2015, the DigiCert Certificate Transparency (CT) Log was the first independent CT log to be incorporated by Google in the Chrome browser. Certificate Transparency is Google's proposed solution to the, until now, inherent opaqueness of the CA ecosystem. CT provides a way for every certificate issued by any publicly trusted CA to be publicly logged, monitored, and audited. Certificate Transparency's main goal is to "remedy certificate-based threats by making the issuance and existence of SSL certificates open to scrutiny by domain owners, CAs, and domain users."

The surprisingly long list of CT log operators:

https://www.gstatic.com/ct/log_list/all_logs_list.json