Security Now! #699 - 01-29-19

Browser Extension Security

This week on Security Now!

This week we look at the expressive power of the social media friends we keep, the persistent DNS hijacking campaign which has the US Government quite concerned, last week's iOS and macOS updates (and doubtless another one very soon!), a valiant effort to take down malware distribution domains, Chrome catching up to IE and Firefox with drive-by file downloads, two particularly worrisome vulnerabilities in two Cisco router models publicly disclosed last Friday, some interesting miscellany, a particularly poignant SpinRite data recovery testimonial, and then some close looks at the state of the industry and the consequences of extensions to our web browsers.

The Weakest Link?

Security News

The acquaintances we keep...
The research is titled: "Information flow reveals prediction limits in online social activity"

I thought we'd start this week with an interesting tidbit to tuck away and bring out at the proper time, perhaps when standing around among a group of non-techie people:

ABSTRACT:
Modern society depends on the flow of information over online social networks, and users of popular platforms generate substantial behavioural data about themselves and their social ties. However, it remains unclear what fundamental limits exist when using these data to predict the activities and interests of individuals, and to what accuracy such predictions can be made using an individual's social ties. Here, we show that 95% of the potential predictive accuracy for an individual is achievable using their social ties only, without requiring that individual's data. We used information theoretic tools to estimate the predictive information in the writings of Twitter users, providing an upper bound on the available predictive information that holds for any predictive or machine learning methods. As few as 8–9 of an individual's contacts are sufficient to obtain predictability comparable to that of the individual alone. Distinct temporal and social effects are visible by measuring information flow along social ties, allowing us to better study the dynamics of online activity. Our results have distinct privacy implications: information is so strongly embedded in a social network that, in principle, one can profile an individual from their available social ties even when the individual themselves forgoes the platform completely.

Sophos reported in their coverage of this that:
"Jim Bagrow, a mathematician at the University of Vermont who led the research, said in a statement that he and his team used statistical models to analyze data from more than 30 million publicly available Twitter posts from 13,905 accounts. Using that data, they used machine learning to accurately predict what a person would post based on what their contacts have posted."

The researchers say that what's true for Twitter goes for Facebook:
"Even if you've never posted to either platform, it just takes between eight and nine of your friends to build a profile of your likes, interests and personality on social media.

So, in other words, what these researchers found was that if analyzing the postings of an individual provided a certain specificity of information about who this person is and what they believe, an equivalent level of information specificity can be obtained -- instead -- by examining 8-9 of their online friends.

It's not surprising, I suppose, that this study's finding strongly suggest that the most people select friends who are most like themselves.

The DNS Hijacking Campaign
Back on January 11th, US-CERT warned of an ongoing concerted DNS hijacking campaign: https://www.us-cert.gov/ncas/current-activity/2019/01/10/DNS-Infrastructure-Hijacking-Campaign

The National Cybersecurity and Communications Integration Center (NCCIC), part of the Cybersecurity and Infrastructure Security Agency (CISA), is aware of a global Domain Name System (DNS) infrastructure hijacking campaign. Using compromised credentials, an attacker can modify the location to which an organization's domain name resources resolve. This enables the attacker to redirect user traffic to attacker-controlled infrastructure and obtain valid encryption certificates for an organization's domain names, enabling man-in-the-middle attacks.

NCCIC encourages administrators to review the FireEye and Cisco Talos Intelligence blogs on global DNS infrastructure hijacking for more information. Additionally, NCCIC recommends the following best practices to help safeguard networks against this threat.

FireEye's report was titled: "Global DNS Hijacking Campaign: DNS Record Manipulation at Scale" https://www.fireeye.com/blog/threat-research/2019/01/global-dns-hijacking-campaign-dns-record-manipulation-at-scale.html

Introduction

FireEye's Mandiant Incident Response and Intelligence teams have identified a wave of DNS hijacking that has affected dozens of domains belonging to government, telecommunications and internet infrastructure entities across the Middle East and North Africa, Europe and North America. While we do not currently link this activity to any tracked group, initial research suggests the actor or actors responsible have a nexus to Iran. This campaign has targeted victims across the globe on an almost unprecedented scale, with a high degree of success. We have been tracking this activity for several months, mapping and understanding the innovative tactics, techniques and procedures (TTPs) deployed by the attacker. We have also worked closely with victims, security organizations, and law enforcement agencies where possible to reduce the impact of the attacks and/or prevent further compromises.

While this campaign employs some traditional tactics, it is differentiated from other Iranian activity we have seen by leveraging DNS hijacking at scale. The attacker uses this technique for their initial foothold, which can then be exploited in a variety of ways. In this blog post, we detail the three different ways we have seen DNS records be manipulated to enable victim compromises. Technique 1, involving the creation of a Let's Encrypt certificate and changing the A record, was previously documented by Cisco's TALOS team. The activity described in their blog post is a subset of the activity we have observed.

So let's think about what this means:
If a site's DNS record can be changed then, subject to DNS caching expiring and needing to be renewed, all traffic to the domains controlled by the altered DNS record will be redirected to an attacker-controlled IP address. And since that redirection includes the lookups being performed by the Let's Encrypt services, the attacker is able to immediately auto-obtain and auto-install a valid certificate for their fraudulent site at its fraudulent IP... to thus further defraud every visitor to that site who will see the fully-correct https:// with all security indications.

We've often talked about the need to have secure domain name lookup. And as we can see, this was made more important with the advent of Let's Encrypt... Because incorrect DNS is Let's Encrypt's greatest weakness. I've raved about Let's Encrypt in the past. As we know, it allows the equivalent of opportunistic encryption for any website that wants it. It's greatest power is that it fully automates the entire certificate issuance process. And it's greatest liability is that the entire certificate issuance process has been automated.

The predictable effect this has had is that the identity-assertion value of HTTPS has been reduced significantly. It's interesting that these bad guys have taken the time to obtain Let's Encrypt certs for their DNS-hacked domains since that suggests that HTTPS has moved from optional to required in the world. That's been all for the good. And I suppose that reducing certificate issuance cost to zero has been a necessary part of that, since it at least softens the complaints of the grumpy old bearded Unix gurus who think that HTTP should be just fine for their site which contains nothing of security importance to anyone.

Back in March of 2017 we covered the news that an analysis of Let's Encrypt certificates revealed that, even back then, 15,270 Let's Encrypt certificates containing the string “PayPal”, to be used in phishing campaigns, had been issued.

The benefits of automated certificate issuance mean that it's here to stay. Yet so, then, will its malicious exploitation. One interesting solution might be to require all automatically issued certs to contain an extra flag bit indicating that the cert was issued algorithmically and without any individual human oversight. Then web browsers could show some sort of different or additional indication whenever the user was visiting a site whose certificate was issued by a bot. Since it's unlikely that any major institution would be using Let's Encrypt -- or perhaps someday some other facility -- it would raise a useful warning.

The US Government is taking this VERY seriously. The threat that US-CERT highlighted then came to the attention of the US Department of Homeland Security:
https://cyber.dhs.gov/assets/report/ed-19-01.pdf

Background:
In coordination with government and industry partners, the Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA) is tracking a series of incidents involving Domain Name System (DNS) infrastructure tampering. CISA is aware of multiple executive branch agency domains that were impacted by the tampering campaign and has notified the agencies that maintain them. Using the following techniques, attackers have redirected and intercepted web and mail traffic, and could do so for other networked services:

1. The attacker begins by compromising user credentials, or obtaining them through alternate means, of an account that can make changes to DNS records.

2. Next, the attacker alters DNS records, like Address (A), Mail Exchanger (MX), or Name Server (NS) records, replacing the legitimate address of a service with an address the attacker controls. This enables them to direct user traffic to their own infrastructure for manipulation or inspection before passing it on to the legitimate service, should they choose. This creates a risk that persists beyond the period of traffic redirection.

3. Because the attacker can set DNS record values, they can also obtain valid encryption certificates for an organization's domain names. This allows the redirected traffic to be decrypted, exposing any user-submitted data. Since the certificate is valid for the domain, end users receive no error warnings.

To address the significant and imminent risks to agency information and information systems presented by this activity, this emergency directive requires the following near-term actions to mitigate risks from undiscovered tampering, enable agencies to prevent illegitimate DNS activity for their domains, and detect unauthorized certificates.

Required Actions:

Action One: Audit DNS Records

- Within IO business days, for all .gov or other agency-managed domains, audit public DNS records on all authoritative and secondary DNS servers to verify they resolve to the intended location. If any do not, report themto CISA. Cf SA recommends agencies prioritize NS records and those associated with key agency services offered to organizational users and the public (for example, websites that are central to the agency's mission, MX records, or other services with high utilization).

Action Two: Change DNS Account Passwords

- Within 10 business days, updatethe passwords for allaccountson systems that can make changes to your agency's DNS records. 2 CISA recommends the use of password managers to facilitate complex and unique passwords.

Action Three: Add Multi-Factor Authentication to DNS Accounts

- Within 10 business days, implement multi-factor authentication(MF A) for all accounts on systems that can make changes to your agency's DNS records.3 If MFA cannot be enabled, provide CISA with the names of systems, why it cannot be enabled within the required timeline, and when it could be enabled. CISA recommends using additional factors that are resilient to phishing. Consistent with NIST SP 800-63B, Short Message Service (SMS)-based MF A is not recommended.

Action Four: Monitor Certificate Transparency Logs

- Within 10 business days, CISA will begin regular delivery of newly added certificates to Certificate Transparency (CT) logs for agency domains, via the Cyber Hygiene service.
- Upon receipt, agencies shall immediately begin monitoring CT log data for certificates issued that they did not request. If an agency confirms that a certificate was unauthorized, it must report the certificate to the issuing certificate authority and to CISA.

CISA Actions:

- CISA will provide technical assistance to agencies that report anomalous DNS records.
- CISA will review submissions from agencies that cannot implement MF A on DNS accounts within the timeline and contact agencies, as needed.
- CISA will provide regular delivery of newly added certificates to CT logs for agency domains via the Cyber Hygiene service.
- CISA will provide additional guidance to agencies through an emergency directive coordination call following the issuance of this directive, as well as through individual engagements upon request (through CyberLiaison).

Beginning February 6, 2019, the CISA Director will engage Chief Information Officers (CIO) and/or Senior Agency Officials for Risk Management (SA ORM) of agencies that have not completed required actions, as appropriate, to ensure their most critical federal information systems are adequately protected.

By February 8, 2019, CISA will provide a report to the Secretary of Homeland Security and the Director of Office of Management and Budget (0MB) identifying agency status and outstanding issues.

Duration: This emergency directive remains in effect until replaced by a subsequent binding operational directive or terminated through other appropriate action.

Time to update iOS and macOS

The first updates of 2019 include some useful security fixes.
iOS moves to v12.1.3 (I believe I heard Rene say last week that he was surprised that this wasn't v12.2 by now. Presumably Apple has other plans for v12.2.

In their document titled "About the security content of iOS 12.1.3" Apple details:
https://support.apple.com/en-gb/HT209443

All of these apply to iPhone 5s and later, iPad Air and later, and iPod touch 6th generation

Because Apple uses the term "arbitrary code execution", and because those are definitely not good, I searched the security summary page for the word "arbitrary" and it lit up...

Bluetooth
Impact: An attacker in a privileged network position may be able to execute arbitrary code
Description: An out-of-bounds read was addressed with improved input validation.
CVE-2019-6200: an anonymous researcher

FaceTime
Impact: A remote attacker may be able to initiate a FaceTime call causing arbitrary code execution
Description: A buffer overflow issue was addressed with improved memory handling.
CVE-2019-6224: Natalie Silvanovich of Google Project Zero

Kernel
Impact: A malicious application may be able to execute arbitrary code with kernel privileges
Description: A memory corruption issue was addressed with improved input validation.
CVE-2019-6210: Ned Williamson of Google

Kernel
Impact: An application may be able to execute arbitrary code with kernel privileges
Description: A buffer overflow was addressed with improved bounds checking.
CVE-2019-6213: Ian Beer of Google Project Zero

libxpc (the Apple XPC system is part of iOS process management)
Impact: A malicious application may be able to execute arbitrary code with kernel privileges
Description: A memory corruption issue was addressed with improved input validation.
CVE-2019-6218: Ian Beer of Google Project Zero

SQLite
Impact: A maliciously crafted SQL query may lead to arbitrary code execution
Description: Multiple memory corruption issues were addressed with improved input validation.
CVE-2018-20346: Tencent Blade Team
CVE-2018-20505: Tencent Blade Team
CVE-2018-20506: Tencent Blade Team

WebKit
Impact: Processing maliciously crafted web content may lead to arbitrary code execution
Description: A memory corruption issue was addressed with improved memory handling.
CVE-2019-6227: Qixun Zhao of Qihoo 360 Vulcan Team
CVE-2019-6233: G. Geshev from MWR Labs working with Trend Micro's Zero Day Initiative
CVE-2019-6234: G. Geshev from MWR Labs working with Trend Micro's Zero Day Initiative

WebKit
Impact: Processing maliciously crafted web content may lead to arbitrary code execution
Description: A type confusion issue was addressed with improved memory handling.
CVE-2019-6215: Lokihardt of Google Project Zero

WebKit
Impact: Processing maliciously crafted web content may lead to arbitrary code execution
Description: Multiple memory corruption issues were addressed with improved memory handling.
CVE-2019-6212: an anonymous researcher, an anonymous researcher
CVE-2019-6216: Fluoroacetate working with Trend Micro's Zero Day Initiative
CVE-2019-6217: Fluoroacetate working with Trend Micro's Zero Day Initiative, Proteas, Shrek_wzw, and Zhuo Liang of Qihoo 360 Nirvan Team
CVE-2019-6226: Apple

WebRTC
Impact: Processing maliciously crafted web content may lead to arbitrary code execution
Description: A memory corruption issue was addressed with improved state management.
CVE-2019-6211: Georgi Geshev (@munmap), Fabi Beterke (@pwnfl4k3s), and Rob Miller (@trotmaster99) of MWR Labs (@mwrlabs) working with Trend Micro's Zero Day Initiative

Although iOS has historically been less prone to reverse engineering attacks than Windows, some of these are not good and might appear in the wild once the patches are seen. While the vulnerability Window may not be large, a phenomenal number of devices are initially susceptible.

On the macOS side the update is known under three names depending upon the macOS version, either: "The security content of macOS Mojave 10.14.3" / "Security Update 2019-001 High Sierra" / "Security Update 2019-001 Sierra"

Due to an increasing common code base, most of the CVEs mentioned in the iOS v12.1.3 update

are also present for macOS, including he one for BlueTooth, FaceTime, WebRTC, SQLite, IOKit, and those affecting the kernel. Specific to only macOS Sierra and High Sierra are a Remote Code Execution vulnerability affecting the Intel Graphics Driver and a privilege elevation issue affecting the OS's hypervisor. All macOS versions are also affected by an out-of-bounds flaw in QuartzCore that could allow an attacker to read restricted memory. So... Update now and update often!

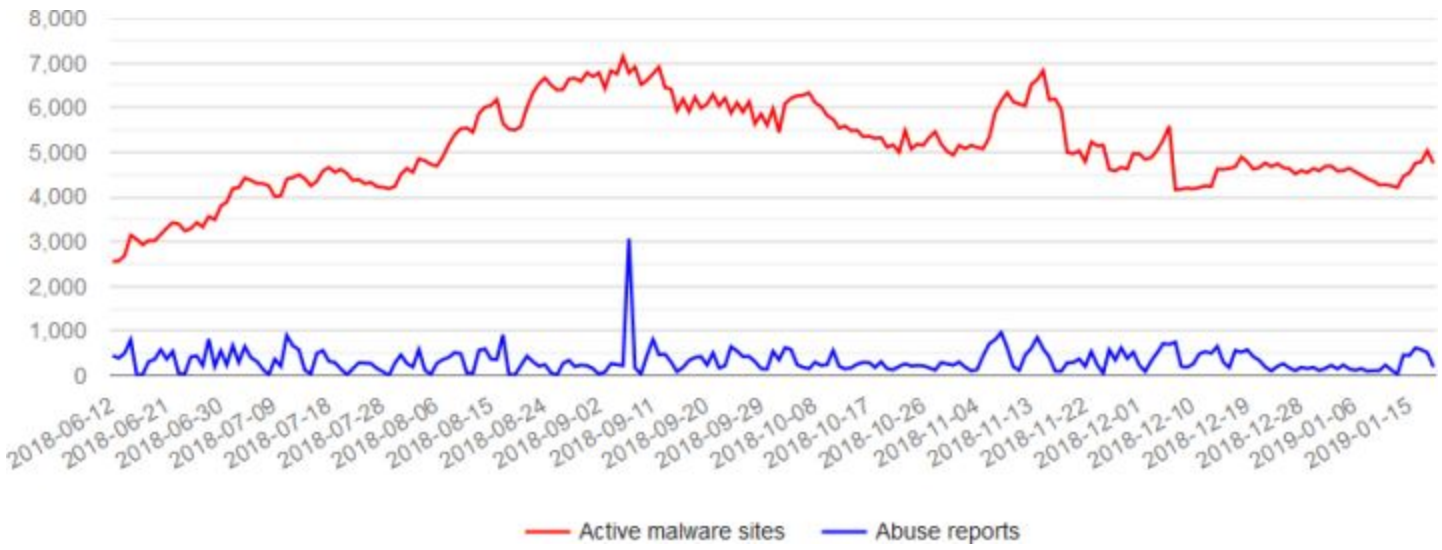
And we can also expect another update within the next few days! → The Facetime bug!

Taking down 100,000 malware sites
https://abuse.ch/blog/how-to-takedown-100000-malware-sites/

At the end of March 2018, the Swiss site "abuse.ch" initiated it's most recent project called "URLhaus". The goal of URLhaus was to collect and share URLs that are being used for distributing malware. The project was a huge success: with the help of 265 security researchers spread across the globe, URLhaus was able to coordinate the takedown almost 100,000 malware distribution sites in 10 months! During that time, these 265 researchers identified and submitted in average 300 malware sites to URLhaus each day in order to help others to protect their network and users from malware campaigns.

Working with the security community URLhaus managed to get the attention of many hosting providers, helping them to identify and re-mediate compromised websites hosted in their network. This was not a simple task, specially for large hosting providers who have tens of thousands of customers, and consequently a great many hijacked websites within their network that are being abused by cybercriminals to distribute malware.

Nevertheless, URLhaus in average counts between 4,000 and 5,000 active malware distribution sites every day, which is a way too much. The following chart shows the number of active malware distribution sites tracked since the launch of URLhaus. The blue line indicates the number of abuse reports sent out to the corresponding hosting providers and network owners.



Having a look at the average takedown time doesn't make the situation any better: In average,

malware distribution sites stay active for more than a week (8 days, 10 hours, 24 minutes). That's more than enough time to infect thousands of device every day.

The table below shows the top malware hosting networks, hosting active malware content (counting online malware distribution sites only as of Jan 20th, 2019). As you can easily spot, 2/3 of the top malware hosting networks are hosted either in the US or China.

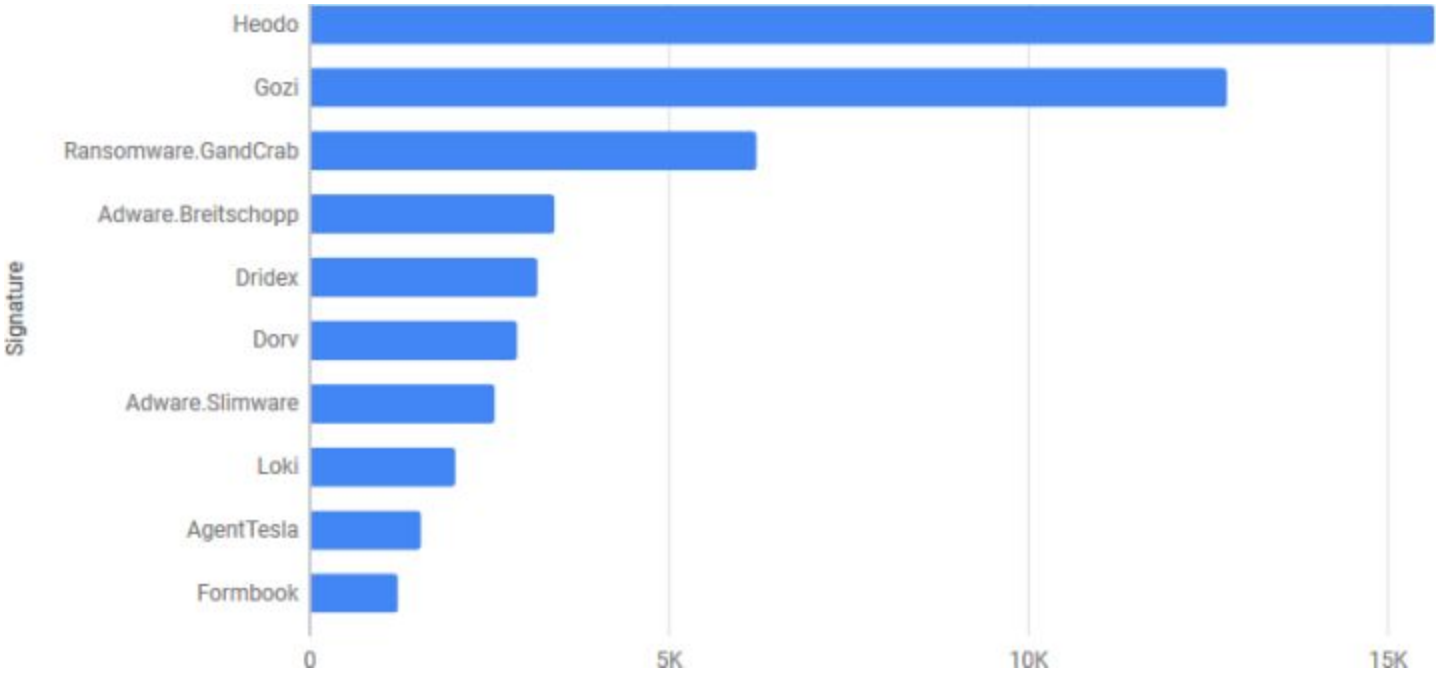
Rank	ASN	Country	Average Reaction Time	Malware URLs
1	AS14061 DIGITALOCEAN-ASN - DigitalOcean, LLC	US	6 days, 12 hours, 56 minutes	307
2	AS4134 CHINANET-BACKBONE No.31,Jin-rong Street	CN	1 month, 9 days, 19 hours, 22 minutes	256
3	AS4837 CHINA169-BACKBONE CHINA UNICOM China169	CN	1 month, 23 days, 8 hours, 41 minutes	163
4	AS48815 CRITICALCASE	IT	21 hours, 58 minutes	151
5	AS46606 UNIFIEDLAYER-AS-1 - Unified Layer	US	2 days, 11 hours, 54 minutes	127
6	AS53667 PONYNET - FranTech Solutions	US	13 days, 3 hours, 37 minutes	105
7	AS16276 OVH	FR	5 days, 22 hours, 6 minutes	104
8	AS60144 THREE-W-INFRA-AS -- TRANSIT --	NL	9 days, 10 hours, 37 minutes	83
9	AS13335 CLOUDFLARENET - Cloudflare, Inc.	US	13 days, 7 hours, 5 minutes	67
10	AS37963 CNNIC-ALIBABA-CN-NET-AP Hangzhou Alibaba	CN	1 month, 2 days, 0 hours, 1 minutes	66
11	AS8342 RTCOMM-AS	RU	10 days, 8 hours, 9 minutes	63
12	AS36352 AS-COLOCROSSING - ColoCrossing	US	16 days, 9 hours, 57 minutes	53
13	AS3462 HINET Data Communication Business Group	TW	17 days, 6 hours, 19 minutes	51
14	AS23650 CHINANET-JS-AS-AP CHINANET jiangsu province	CN	3 days, 11 hours, 50 minutes	51
15	AS3462 HINET Data Communication Business	TW	17 days, 6 hours, 19 minutes	51

What is also an eye-catcher is the takedown time of malware sites hosted in China: **The three top Chinese malware hosting networks have an average abuse desk reaction time of more than a month!**

A vast amount of the malware distribution sites tracked by URLhaus are related to Emotet (aka Heodo). Emotet gets propagated through spam that hits users inbox almost every day. These malspam campaigns usually contain a malicious office document with macros. Once the victim opens the document and enables macros, it will automatically download and execute Emotet from a compromised website. To bypass spam filters, these malspam campaigns sometimes point to a compromised website that hosts the malicious office document instead of attaching it to the email directly.

To dismantle these campaigns and prevent users from getting infected with Emotet, it is essential that the associated malware distribution sites get cleaned up quickly by responsible hosting providers.

The weight that Emotet has in the current threat landscape also becomes more clear when having a look at the identified malware families associated with the payloads URLhaus received from the tracked malware distribution sites. Across the **380,000 malware samples** (payloads) that URLhaus has collected over the past 10 months, Emotet/Heodo is the top malware as the following chart documents.



Chrome playing catch-up to IE and FireFox...
It will soon be blocking so-called "drive-by downloads" from iFrames.

Web browser iFrames have always been frightening from a security standpoint. They are another classic tradeoff between security and flexibility. iFrame is short for "inline frame". It allows the designer of a web page to set aside a rectangular region -- a frame -- whose contents will be filled in by the result of an iFrame URL fetch. The origin web page specifies the URL, the browser goes to fetch it and to render it as a mini-web page onto itself.

iFrames are what have enabled the entire web browser advertising industry, since they allow web pages to monetize themselves by agreeing to set aside space -- these frames --which will be available for fill-in by advertising aggregators, and for which the originating web sites are paid based upon the number of times those iFrame URLs are pulled and displayed.

The danger is that, unless controlled and restricted and restrained, those "mini web pages" are full browser citizens capable of loading anything they chose and running any JavaScript they wish.

On October 6th, 2015, the Chromium bugs list contained the observation:
https://bugs.chromium.org/p/chromium/issues/detail?id=539938#c3

IE and Firefox does not allow download from sandboxed iframe. The posting included this sample HTML code demonstrating "the bug"...

```
<!DOCTYPE>
<html>
<head>
</head>
<body>
<iframe sandbox src="https://osdn.jp/pastebin/2131?action=download"></iframe>
</body>
</html>
```

The point begin that, back then IE and FF had closed this unwanted hole whereas Chrome had not. Today, Chrome still has not. But, happily, that finally appears to be changing.

Various tech news outlets are reporting that Google's Developers have finally started working on adding drive-by download protection to Chromium. That new feature is already active in the current Chrome Canary edition and is scheduled to land in the stable version, Chrome 73 sometime in March or April.

Analysis has shown that when downloads are triggered in a web page's iframe element, hidden in its code, those downloads are almost always malicious. This is the #1 way "malvertising" is still crawling into people's machines.

The Chromium developers stated: "We plan to prevent downloads in sandboxed iframes that lack a user gesture, and this restriction could be lifted via an 'allow-downloads-without-user-activation' keyword, if present in the sandbox attribute list."

The solution of allowing an override makes sense since there might be instances where a web page might wish to allow content within an iFrame to have download privileges. So in those situations those specific iFrames can be given that right. But the default in Chrome will finally be switching to "no."

We should note that the presence of an override means that compromised sites in so-called "watering hole attacks" might still enable iFrame downloads since the hackers would be able to give an injected iFrame those privileges when the iFrame is injected. But this does seem like a useful and clean change to Chrome's default behavior.

Cisco RV320 and RV325 WAN VPN routers are under remote assault
Two days after Cisco released patches (last Wednesday), security researcher David Davidson published proof-of-concept exploit demonstration code on Github:

https://github.com/0x27/CiscoRV320Dump

Titled: "CVE-2019-1652 /CVE-2019-1653 Exploits For Dumping Cisco RV320 Configurations & Debugging Data AND Remote Root Exploit!"

Attacks against these routers started shortly after David's code went public.

So, first of all, the two vulnerabilities are horrendous:

- CVE-2019-1653 - allows a remote attacker to get sensitive device configuration details without a password.

- CVE-2019-1652 - allows a remote attacker to inject and run admin commands on the device without a password.

David's Github posting could not have been more seductive … and damaging:

CiscoRV320Dump

CVE-2019-1653/CVE-2019-1652 Exploits For Dumping Cisco RV320 Configurations and getting RCE

Implementations of the CVE-2019-1652 and CVE-2019-1653 exploits disclosed by Red Team Pentesting GmbH.

I only tested these on an RV320, but according to the Cisco advisory, the RV325 is also vulnerable.

The following Shodan queries appear to find them, if you are curious about how many are out there. There seems to be quite a few...
ssl:RV320
ssl:RV325
port:161 RV325
port:161 RV320

The vulnerabilities allow for the following:

- Dumping (Plaintext) Configuration File! (includes hashes for the webUI!)
- Dumping (Encrypted) Diagnostic/Debug Files! (including config, and the /etc and /var directories)
- Decrypting the encrypted Diagnostic/Debug Files! (yes, you get /etc/shadow!)
- Post-Authentication Remote Command Injection as root in the webUI!

As an aside, the default creds are cisco:cisco.

Troy Mursch at Bad Packets Report did some white-hat scanning and put up some more details last Saturday:
https://badpackets.net/over-9000-cisco-rv320-rv325-routers-vulnerable-to-cve-2019-1653/

Posted on January 26, 2019 by Troy Mursch

Over 9,000 Cisco RV320/RV325 routers are vulnerable to CVE-2019-1653

On Friday, January 25, 2019, our honeypots detected opportunistic scanning activity from multiple hosts targeting Cisco Small Business RV320 and RV325 routers. A vulnerability exists in these routers that allow remote unauthenticated information disclosure (CVE-2019-1653) leading to remote code execution (CVE-2019-1652).

Using data provided by BinaryEdge, we've scanned 15,309 unique IPv4 hosts and determined 9,657 Cisco RV320/RV325 routers are vulnerable to CVE-2019-1653.

- 6,247 out of 9,852 Cisco RV320 routers scanned are vulnerable (1,650 are not vulnerable and 1,955 did not respond to our scans)
- 3,410 out of 5,457 Cisco RV325 routers scanned are vulnerable (1,027 are not vulnerable and 1,020 did not respond to our scans)

These Cisco VPN routers are very popular among enterprise and ISP providers. If you have any of these routers within your purview, stop listening to this podcast right now, take them offline, update them to the latest firmware, and then carefully scrutinize your network traffic logs for any signs of related misconduct!

(And once you've done that, resume listening to this podcast because there's lots more to talk about this week!)

Miscellany:

Last Sunday's first presentation of the completed and finished SQRL system went well. I had previously given presentations at DigiCert's Security Summit and then to Stina and her crypto-techies at Yubico. Each of those points in time caught SQRL where it was then. Today, as I've been saying recently, it's finally finished. So the meetup of the LETHAL group (L.A. Ethical Hackers and Leets) was my first opportunity to present the entire system soup to nuts.

During that presentation I kept referring to GRC's online documentation, but also needing to continually apologize that those pages were more than five years old and that MUCH had changed since then. So my own next top priority -- is to go through, revise and synchronize the original online documentation to what we finally wound up doing.

In the meantime, our wonderful XenForo developer has sent me v1.0.0 of his integration SQRL into XenForo using the new SQRL Service Provider API. It's a formal XenForo add-on which could also be dropped into any XenForo forum site to enable SQRL authentication. That site would need to have an SSP API server, but as I've mentioned before, that's on the way now, too. Once I'm finished with today's podcast I'll be incorporating SQRL into the web forums for testing by our gang in the grc.sqrl newsgroup.

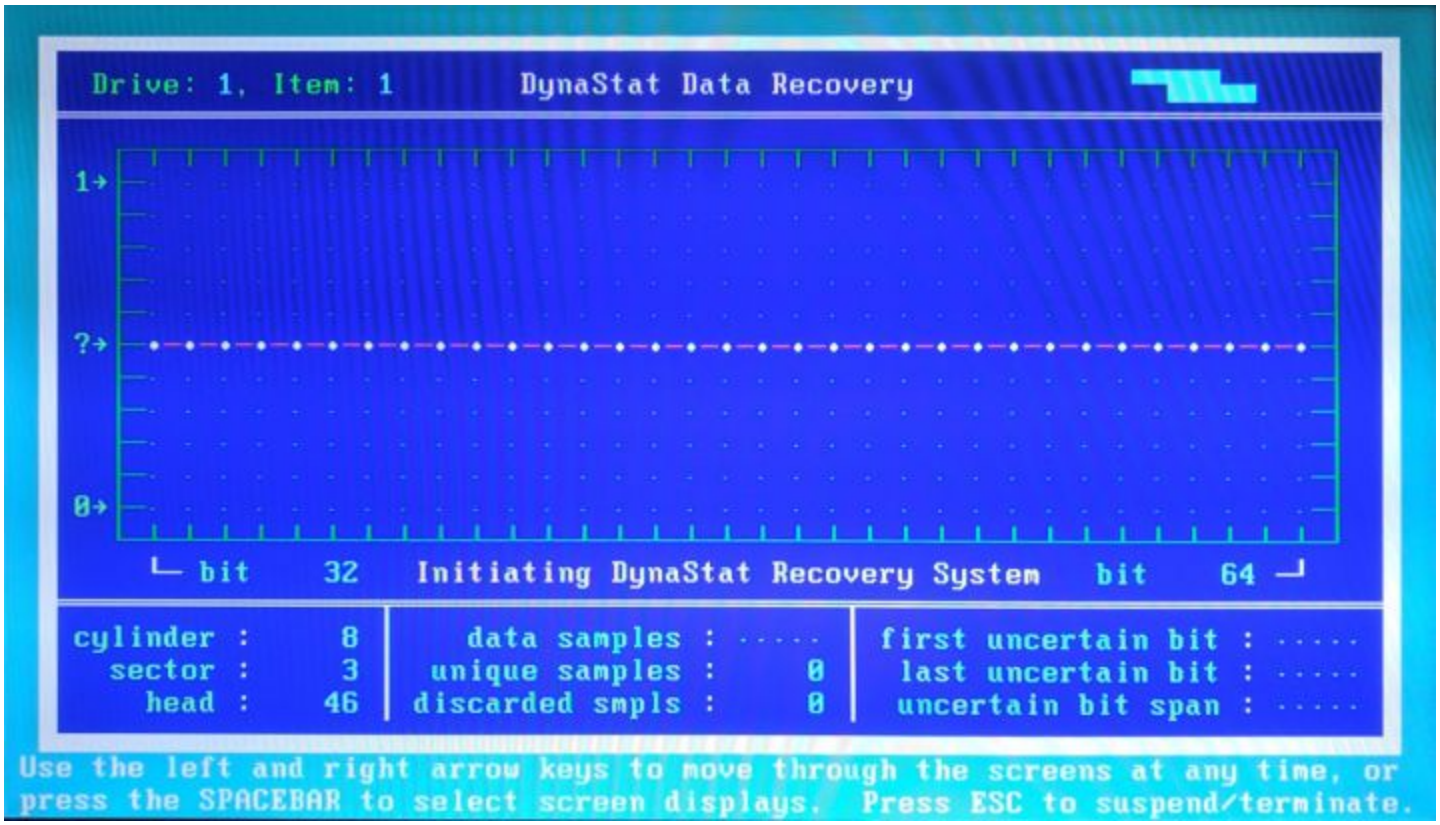
SpinRite

From: Joey Kelley
Subject: SpinRite Testimonial

Hello Steve and the rest of the GRC team,

I purchased SpinRite some time ago and have had few occasions (thankfully) to use it. Most of my customers (I own a small consulting firm on the side) have finally been convinced of the need to back up their data and even when a hard drive fails, it is usually an inconvenience, not total disaster.

Earlier today, I tweeted you a quick picture entitled 'DynaStat Engaged!' showing SpinRite going to work on a drive that I am glad to say was successfully helped by your product. Since the story behind this is interesting, I thought I would relay it to you.



In the past couple of years I have become quite good friends with a couple and their family that run a store and lunch counter near where my parents live. I swing in often and I've tweaked their computers here and there as asked. About two weeks ago they mentioned they had an old computer in the back that they would like to have the data from. That old story, repeated countless times between you and Leo on Security Now! began to play in my head - ending with that well remembered line '... and there is no backup'.

Knowing that these people have had a lot of knocks in their lives, I thought I'd help them out and wound up taking the old computer home. I set it up through a USB to IDE adapter and set SpinRite to going. It found the typical one unrecoverable sector in the early sectors of the drive - did some recovery and then finished. However, it would still not come up in either Windows or Linux as a valid drive. Figuring I had nothing to lose, I re-ran SpinRite on it with a level four scan, and after only four hours the scan completed. I was able to pull the drive up in both Linux and Windows, and was able to copy all of their data off the drive. Success!

Then something caught my eye - the name of one of the User folders: Almost 15 years ago, these folks lost their 14 year old son in an accident. This, was the family's computer at the time and it contains photos that they have nowhere else of their son, which they now have back thanks to SpinRite.

Thank you - for giving this family their memories back.

Joey, JoeyFixesComputers.com
PS - You have my permission to use this e-mail as you see fit!

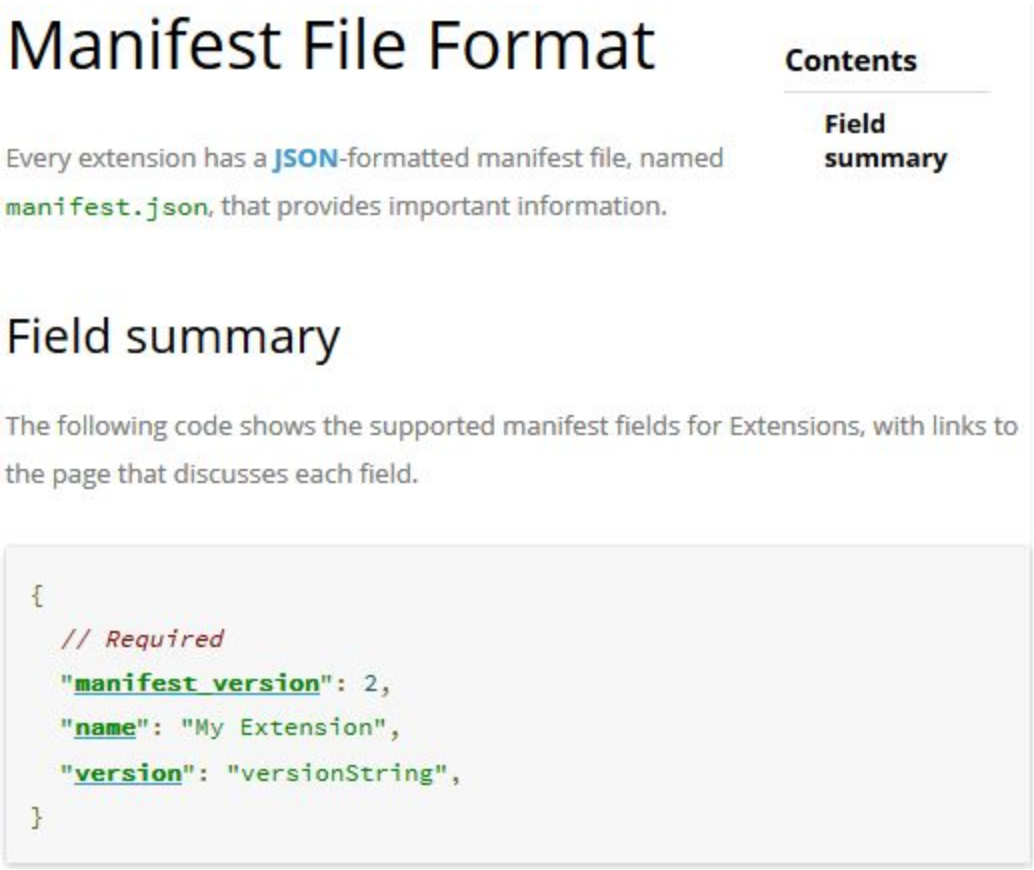
-Joey Kelley
JoeyKelley.com - My Life Online
JoeyKelleyPhoto.com - Photographing Today, For Tomorrow

Browser Extension Security

Chrome Extensions Permissions...
The persistent trade-off between capability, flexibility and security.

I call this "persistent" rather than inevitable because I think that using today's computer technology and models we're pretty much stuck with trading off one for the other, security versus freedom. But I think that's only due to the way we are currently solving these problems. It's beginning to feel as though inertia is holding us back from a radically different approach. We may see it in our lifetime.

But for now, here and today, we clearly do face a tradeoff. And this is a tradeoff which Google, with Chrome -- now the #1 web browser worldwide -- is continually struggling with. And to that end, they are working to evolve the interface offered to 3rd-party browser extensions. That interface determines what power browser extensions have. We've been at Version 2 and the Chromium development team is heading toward Version 3... but not without predictably ruffling some feathers.



Manifest File Format

Every extension has a **JSON**-formatted manifest file, named `manifest.json`, that provides important information.

Contents

Field summary

Field summary

The following code shows the supported manifest fields for Extensions, with links to the page that discusses each field.

```
{
  // Required
  "manifest_version": 2,
  "name": "My Extension",
  "version": "versionString",
}
```

ZDNet notes in their coverage of this: "Chrome API update will kill a bunch of other extensions, not just ad blockers. Chrome extensions for antivirus products, parental control enforcement, and various privacy-enhancing services also affected."

And in two separate postings, Bleeping Computer notes that our podcast favorite uBlock Origin may die and that "TamperMonkey" could also bite the dust.

Both of Raymond Hill's extremely popular Chrome extensions, uBlock Origin and uMatrix would, as he understands it, die if the changes happen as they are currently defined in the next version draft. In the forthcoming version 3, the Chrome developers have stated their intention to limit the blocking capabilities of the webRequest API, which Raymond's extensions require. The current proposal reads: "In Manifest V3, we will strive to limit the blocking version of webRequest, potentially removing blocking options from most events (making them observational only). Content blockers should instead use declarativeNetRequest." Raymond said that phasing out 'webRequest' API in favor of the 'declarativeNetRequest' API would mean the death of uBlock Origin which is used by over 10 million users on Chrome.

Raymond wrote to the bug tracking page the Manifest V3: "If this (quite limited) declarativeNetRequest API ends up being the only way content blockers can accomplish their duty, this essentially means that two content blockers I have maintained for years, uBlock Origin ("uBO") and uMatrix, can no longer exist."

In his posting, Raymond explained that his extensions are incompatible with the proposed declarativeNetRequest API because it allows for only one specific filtering engine, whereas uBlock Origin and uMatrix rely on various filtering designs to do their job properly.

The proposed modification is more oriented toward more limited fixed static filtering capabilities such as those provided by AdBlock Plus, but the redesign would also limit the number of filters to 30,000, which is insufficient even for AdBlock Plus.

Ray uses the example of the EasyList filters with rules for removing unwanted web content, which is larger than 30,000 entries and is not sufficient for a modern user's filtering needs. The EasyList filter rule set is used by both AdBlock Plus and uBlock Origin and it is much larger than the limit imposed by the 'declarativeNetRequest'.

And what about "TamperMonkey" ??

First: What IS a TamperMonkey? (It's nothing like a HoneyMonkey.) Although I've been blissfully unaware of it, it's apparently similar in popularity to uBlock Origin with over 10 million users. https://tampermonkey.net/

I've heard of GreaseMonkey, and a bit of research revealed that GreaseMonkey is the old timer here, having been available for Firefox long before Chrome existed.

"Userscripts" provide high-end power users the ability to inject their own scripts into sites they visit for specific purposes. Repositories of userscripts written by others exist, and there appear to be three primarily popular userscript managers in use:

Chrome: Tampermonkey or Violentmonkey
Firefox: Greasemonkey, Tampermonkey, or Violentmonkey
Safari: Tampermonkey
Microsoft Edge: Tampermonkey
Opera: Tampermonkey or Violentmonkey
Maxthon: Violentmonkey
Dolphin: Tampermonkey
UC: Tampermonkey
Qupzilla: (no additional software required)
AdGuard: (no additional software required)

Tampermonkey is the most popular userscript manager, with, as I said, over 10 million users. It's available for Chrome, Microsoft Edge, Safari, Opera Next, and Firefox.

Tampermonkey describes itself saying: Tampermonkey makes it very easy to manage your userscripts and provides features like a clear overview over the running scripts, a built-in editor, ZIP-based import and export, automatic update checks and browser and cloud storage based synchronization.

However, userscripts are very powerful and, of course, the script repositories have been overrun with malicious scripts hoping to get themselves injected into an unwitting user's browser. So it should not be surprising that Google's devs are wondering whether this whole thing is a good idea.

In his Google Groups posting, Jan Biniok, TamperMonkey's creator writes:

Hi Chromium developers, Hi Devlin

I'm the Tampermonkey developer and I have not studied all the planned changes in detail yet, but this is the one that worries me most.

> Beginning in Manifest V3, we will disallow extensions from using remotely-hosted code. This will require that all code executed by the extension be present in the extension's package uploaded to the webstore. Server communication (potentially changing extension behavior) will still be allowed. This will help us better review the extensions uploaded, and keep our users safe. We will leverage a minimum required CSP to help enforce this (though it will not be 100% unpreventable, and we will require policy and manual review enforcement as well).

While the text above might be interpreted in a way that an extension like Tampermonkey can continue to exist, I got the following explanation from Devlin in an email:

> Note that we will be limiting remotely-hosted/arbitrary code execution in all contexts. The goal is that we should be able to perform an in-depth security review of an extension and be confident in what it does and whether it poses a security or privacy risk to users (which is possible through web page contexts, as well). But let's move this conversation to another thread. :)

I understand the need for security, but this means that V3 P1, in the way it's currently planned, will stop Tampermonkey from working entirely, because arbitrary code execution is Tampermonkey's main functionality. Every little userscript would then have to become an own extension. Anyone who wants to do that has to pay $5 to be able to publish an extension. There are so many use cases for userscripts so I hope that this planned change is reconsidered.

One possibility would be e.g. a new permission that relaxes this constraint and allows remote code execution again. All extensions with this permission could then be provided with a special warning and be examined more intensively. What do you think?

I've been working on Tampermonkey since Chrome version 4 or 5 and I could not live without it anymore. :)

Thanks,
Jan

To that, someone named Jeff replies:

> Jan, this new remotely-hosted code restriction also affects my project, a process automation / RPA system. (wranggle.com) I asked the team about it at the Chrome Dev Summit last November, bringing up TamperMonkey as an example of a productive and popular use of remotely-hosted code, arguing that a permission showing a big scary warning on install should be adequate. Their response was that remote code is too big a threat vector, extensions can do too much harm, that even an extension that starts out benevolent might be later compromised. They seem pretty committed to the decision.

My solution is to output a custom browser extension for each customer, one containing all the automations/user scripts they use/want. If you're interested in using this kind of approach for TamperMonkey, I'd be happy to collaborate.

It avoids the new restriction but does complicate the script update process, requiring the user to rebuild periodically. I have some ideas for reducing rebuild frequency though, by making it easy for script authors to serialize css selectors and dom interactions. And luckily, user updates won't get delayed by the store review process--the team said unlisted and domain-locked extensions don't require review.

The Google Dev Jury is still out on how this will all fall out... but it seems foreseeable that as Chrome matures it may be forced to become less of a Swiss Army Knife container and more of a generic safe harbor browser. If that's the case, we can hope that FireFox might evolve into a browser with an equal feature set -- as it has been -- while also, by virtue of having a different user profile, retaining the Wild West extensibility that can have its uses. It feels as though that should exist somewhere.

Or, perhaps, Chrome might opt to have an advanced "you're on your own, here, partner" mode that would enable. My best buddy Mark, who is very funny, has been insisting, for decades, that I name something "Danger Bytes" -- I don't know why -- I suspect that it harkens from "Danger Will Robinson!" He just thinks it's funny. So perhaps Google could allow a "Danger Bytes" mode. That would be a fitting name for it!

EmPoWeb: Empowering Web Applications with Browser Extensions
A French security researcher, "Doli`ere Francis Som´e" took a good long and hard look at the security implications of the extreme powers which are (currently) given to web browser extensions:

Abstract—Browser extensions are third party programs, tightly integrated into browsers, where they execute with elevated privileges in order to provide users with additional functionalities. Unlike web applications, extensions are not subject to the Same Origin Policy (SOP) and therefore can read and write user data on any web application. They also have access to sensitive user information including browsing history, bookmarks, credentials (cookies) and list of installed extensions. They have access to a permanent storage in which they can store data as long as they are installed in the user's browser. They can trigger the download of arbitrary files and save them on the user's device.

For security reasons, browser extensions and web applications are executed in separate contexts. Nonetheless, in all major browsers, extensions and web applications can interact by exchanging messages. Through these communication channels, a web application can exploit extension privileged capabilities and thereby access and exfiltrate sensitive user information.

In this work, we analyzed the communication interfaces exposed to web applications by Chrome, Firefox and Opera browser extensions. As a result, we identified many extensions that web applications can exploit to access privileged capabilities. Through extensions' APIs, web applications can bypass Same Origin Policy (SOP) and access user data on any other web

application, access user credentials (cookies), browsing history, bookmarks, list of installed extensions, extensions storage, and download and save arbitrary files in the user's device.

Our results demonstrate that the communications between browser extensions and web applications pose serious security and privacy threats to browsers, web applications and more importantly… to users. We discuss countermeasures and proposals, and believe that our study and in particular the tool we used to detect and exploit these threats, can be used as part of extensions review process by browser vendors to help them identify and fix the aforementioned problems in extensions.

So begins a wonderfully detailed 19-page PDF paper whose link I have included in the show notes: http://www-sop.inria.fr/members/Doliere.Some/papers/empoweb.pdf

[[snippage]]

We built a static analyzer and applied it to the message passing interfaces exposed by Google Chrome, Firefox and Opera extensions to web applications. When the tool found that a privileged extension capability could potentially be exploited by web applications, the extension was flagged suspicious. By manually reviewing the code of suspicious extensions, we found that 197 of them (mostly on Chrome) can be exploited by web applications (attackers) to access elevated browser features and APIs and sensitive user information. The extensions we have found have vulnerabilities that can be exploited by web applications to (i) break the privilege separation between extensions and web applications and execute arbitrary code in extensions context, (ii) bypass the Same Origin Policy and access user data on other applications, (iii) read user cookies and use them to mount session hijacking attacks, (iv) access data such as user browsing history, bookmarks, list of installed extensions that besides violating user privacy can be used for tracking purposes, (v) store and retrieve data from extensions persistent storage for tracking purposes and (vi) trigger the download of malicious software on the user device which execution can damage user data.

[[snippage]]

A. Disclosure to vendors:
We have disclosed the list of extensions to Chrome, Firefox and Opera. All vendors acknowledged the issues. Firefox has removed all the reported extensions. Opera has also removed all the extensions but 2 which can be exploited to trigger downloads. The reason given by Opera is that the downloads can only be triggered from specific websites. However, we made them observe that those websites include third party scripts that can also trigger arbitrary downloads. So discussion still continues with Opera on the 2 remaining extensions, in particular to ensure that users are aware of the downloads. Chrome also acknowledged the problem in the reported extensions. We are still discussing with them on potential actions to take: either remove or fix the extensions.

~30~