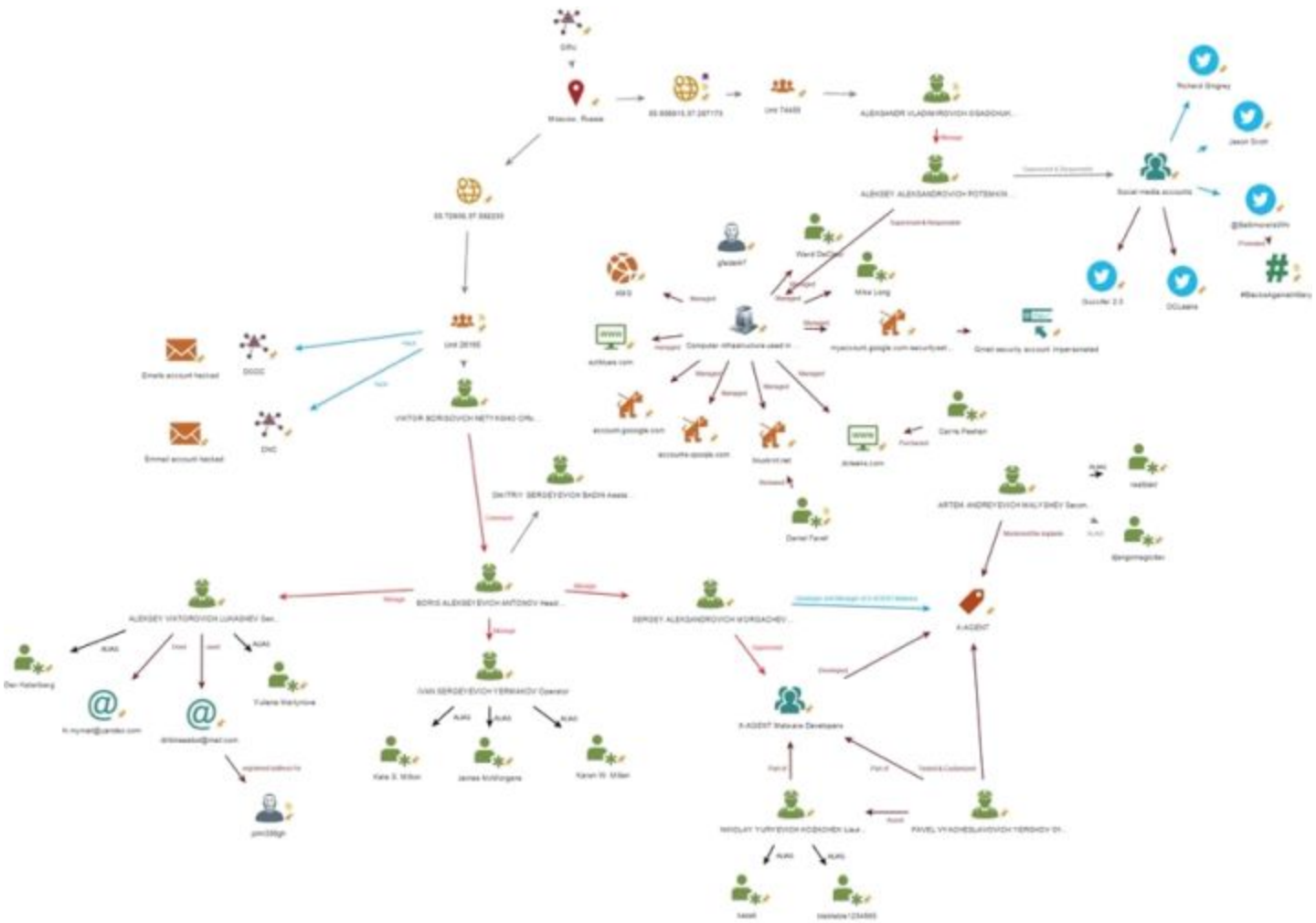
Security Now! #672 - 07-17-18

All up in their business

This week on Security Now!

This week we look at even MORE, new, Spectre-related attacks, highlights from last Tuesday's monthly patch event, advances in GPS spoofing technology, GitHub's welcome help with security dependencies, Chrome's new (or forthcoming) "Site Isolation" feature, when hackers DO look behind the routers they commandeer, the consequences of deliberate BGP routing misbehavior... and reading between the lines of last Friday's DOJ indictment of the US 2016 election hacking by 12 Russian operatives -- the US appears to really have been "all up in their business."

The connectivity and activity map of Russia's U.S. election hacking activity:



Security News

Spectre keeps on trucking
New Spectre 1.1 and Spectre 1.2 CPU Flaws Disclosed
Intel has paid a $100,000 bounty to a pair of MIT-affiliated researchers.
https://people.csail.mit.edu/vlk/spectre11.pdf

ABSTRACT:
Practical attacks that exploit speculative execution can leak confidential information via microarchitectural side channels. The recently-demonstrated Spectre attacks leverage speculative loads which circumvent access checks to read memory-resident secrets, transmitting them to an attacker using cache timing or other covert communication channels.

We introduce Spectre1.1, a new Spectre-v1 variant that leverages speculative stores to create speculative buffer overflows. Much like classic buffer overflows, speculative out-of-bounds stores can modify data and code pointers. Data-value attacks can bypass some Spectre-v1 mitigations, either directly or by redirecting control flow. Control-flow attacks enable arbitrary speculative code execution, which can bypass fence instructions and all other software mitigations for previous speculative-execution attacks. It is easy to construct return-oriented-programming (ROP) gadgets that can be used to build alternative attack payloads.

[And, as if that weren't enough] We also present Spectre1.2: on CPUs that do not enforce read/write protections, speculative stores can overwrite readonly data and code pointers to breach sandboxes.

Bleeping Computer produced a nice summary:

Variant	Description	CVE	Codename	Affected CPUs
Variant 1	Bounds check bypass	CVE-2017-5753	Spectre v1	Intel, AMD, ARM
Variant 1.1	Bounds check bypass on stores	CVE-2018-3693	Spectre 1.1	Intel, ARM
Variant 1.2	Read-only protection bypass	CVE unknown	Spectre 1.2	Intel, ARM
Variant 2	Branch target injection	CVE-2017-5715	Spectre v2	Intel, AMD, ARM
Variant 3	Rogue data cache load	CVE-2017-5754	Meltdown	Intel
Variant 3a	Rogue system register read	CVE-2018-3640	-	Intel, AMD, ARM, IBM
Variant 4	Speculative store bypass	CVE-2018-3639	SpectreNG	Intel, AMD, ARM, IBM

https://www.bleepingcomputer.com/news/security/new-spectre-11-and-spectre-12-cpu-flaws-disclosed/

Their paper is an amazing tour-de-force on the entire topic where I would argue that these guys (a) understand the subtleties of the possible abuse of speculative execution mechanisms as well as anyone I have ever seen and (b) earned and deserve their prize... since this work will be

extremely valuable to Intel and all other modern processor designers and manufacturers.

These attacks bypass the current software mitigations against Spectre 1. They conclude that the long term consequence of the realizations of the dangers of speculative execution will be new generations of designs which are hardened to these threats… but that until then, simpler fixes may be needed at a possibly significant cost to performance.

Patch Tuesday
Among the many fixes we received last week was a fix for the "Lazy FP State Restore" vulnerability affecting Intel CPUs.

Overall we received fixes for 15 Microsoft products and collectively 53 bugs.

Adobe's own update was more than twice as large, fixing 112 vulnerabilities, almost all in their biggest interpreter: Adobe Acrobat and Reader. They published 2 for Flash Player, 3 for their Experience Manager, 3 for Connect, and… 104 in Adobe Acrobat and Adobe Reader.

It would be very interesting to know how many of these are problems introduced in recent years. Unfortunately, it's not in Adobe's corporate interest to leave Acrobat Reader alone, though it was just fine a decade ago.

A we have often observed here, security vulnerabilities tend to increase with the square of the code.

"All Your GPS Are Belong To Us" -
Towards Stealthy Manipulation of Road Navigation Systems
http://people.cs.vt.edu/gangwang/sec18-gps.pdf

The GPS Spoofing state of the art leaps forward

Researchers from Virginia Tech, University of Electronic Science and Technology of China, Microsoft Research

Abstract
Mobile navigation services are used by billions of users around globe today. While GPS spoofing is a known threat, it is not yet clear if spoofing attacks can truly manipulate road navigation systems. Existing works primarily focus on simple attacks by randomly setting user locations, which can easily trigger a routing instruction that contradicts with the physical road condition (i.e., easily noticeable).

In this paper, we explore the feasibility of a stealthy manipulation attack against road navigation systems. The goal is to trigger the fake turn-by-turn navigation to guide the victim to a wrong destination without being noticed. Our key idea is to slightly shift the GPS location so that the fake navigation route matches the shape of the actual roads and trigger physically possible instructions. To demonstrate the feasibility, we first perform controlled measurements by implementing a portable GPS spoofer and testing on real cars. Then, we design a searching

algorithm to compute the GPS shift and the victim routes in real time. We perform extensive evaluations using a trace-driven simulation (600 taxi traces in Manhattan and Boston), and then validate the complete attack via real-world driving tests (attacking our own car). Finally, we conduct deceptive user studies in both the US and China. We show that 95% of the participants follow the navigation to the wrong destination without recognizing the attack. We use the results to discuss countermeasures moving forward.

We show that adversaries can build a portable spoofer with low costs (about $223), which can easily penetrate the car body to take control of the GPS navigation system. Our measurement shows that effective spoofing range is 40–50 meters and the target device can consistently latch onto the false signals without losing connections. The results suggest that adversaries can either place the spoofer inside/under the target car and remotely control the spoofer, or follow the target car in real time to perform spoofing.

Portable GPS Spoofer: We implemented a portable GPS spoofer to perform controlled experiments.

As shown in Figure 1. The spoofer consists of four components: a HackRF One-based frontend, a Raspberry Pi, a portable power source and an antenna. The whole spoofer can be placed in a small box and we use a pen as a reference to illustrate its small size. HackRF One is a Software Defined Radio (SDR). We connect it to an antenna with frequency range between 700 MHz to 2700 MHz that covers the civilian GPS band L1 (1575.42 MHz). A Raspberry Pi 3B (Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM) is used as a central server. It runs an SSH-enabled Raspbian Jessie operating system with a LAMP stack server. GPS satellite signals are generated by an open-source software called Wireless Attack Launch Box (WALB) [6] running on Raspberry Pi.

So... what we now have is the demonstrated ability to VERY inexpensively spoof GPS in such a fashion that humans following their directions will usually not detect that anything is amiss.

GitHub Security Alerts Now Support Python Projects
https://help.github.com/articles/about-security-alerts-for-vulnerable-dependencies/

Last November GitHub announced that for Ruby gems, JavaScript NPM and Python projects it would be tracking and reporting on the presence of known vulnerabilities in the packages upon which GitHub repositories are dependent.

Back then they wrote: "When GitHub receives a notification of a newly-announced vulnerability, we identify public repositories (and private repositories that have opted in to vulnerability detection) that use the affected version of the dependency. Then, we send security alerts to owners and people with admin access to affected repositories. You can also configure security alerts for additional people or teams working in organization-owned repositories."

We detect vulnerable dependencies in public repositories by default. Owners of and people with admin access to private repositories can also opt into vulnerability detection for the repository. For more information, see "Opting into or out of data use for your private repository."

Back then their plan was to support those three classes of projects. They recently announced that Python project repositories were finally added.

On the "Insights" tab under "Dependency graph":

We found potential security vulnerabilities in your dependencies. Some of the dependencies defined in these manifest files have known security vulnerabilities and should be updated.

Only users who have been granted access to vulnerability alerts for this repository can see this message.

Google Enables "Site Isolation" Feature for 99% of Chrome Desktop Users

In Chrome: "chrome://flags" then search "isolation"

- Strict site isolation

Security mode that enables site isolation for all sites. When enabled, each renderer process will contain pages from at most one site, using out-of-process iframes when needed. When enabled, this flag forces the strictest site isolation mode (SitePerProcess). When disabled, the site isolation mode will be determined by enterprise policy or field trial. – Mac, Windows, Linux, Chrome OS, Android
#enable-site-per-process

- Site isolation trial opt-out

Opts out of field trials that enable site isolation modes (SitePerProcess, IsolateOrigins, etc). Intended for diagnosing bugs that may be due to out-of-process iframes. Opt-out has no effect if site isolation is force-enabled via #enable-site-per-process or enterprise policy. Caution: this disables important mitigations for the Spectre CPU vulnerability affecting most computers. – Mac, Windows, Linux, Chrome OS, Android
#site-isolation-trial-opt-out

- Top document isolation

Highly experimental performance mode where cross-site iframes are kept in a separate process from the top document. In this mode, iframes from different third-party sites will be allowed to share a process. – Mac, Windows, Linux, Chrome OS, Android
#enable-top-document-isolation

- PDF Isolation

Render PDF files from different origins in different plugin processes. – Mac, Windows, Linux, Chrome OS
#pdf-isolation

- Use PDF compositor service for printing

When enabled, use PDF compositor service to composite and generate PDF files for printing. When site isolation is enabled, disabling this will not stop using PDF compositor service since the service is required for printing out-of-process iframes correctly. – Mac, Windows, Linux, Chrome OS
#use-pdf-compositor-service-for-print

For me, with 67.0.3396.99, ALL of these are still disabled by default.
https://support.google.com/chrome/answer/7623121?hl=en

Google Chrome Help

- On your computer, open Chrome.
- In the address bar at the top, enter chrome://flags/#enable-site-per-process and press Enter.
- Next to "Strict site isolation," click Enable.
- If you don't see "Strict site isolation," update Chrome.
- Click Relaunch now.

If you're an administrator, learn how to manage site isolation for your organization.
Known issues

- Memory: Site isolation will increase Chrome's memory use by approximately 10%.
- DevTools: Chrome Developer Tools don't fully support cross-site iframes with site isolation.

Threat Model
For a "one-site-per-process" security policy, we assume that an attacker can convince the user to visit a page that exploits a vulnerability in the renderer process, allowing the attacker to run arbitrary code within the sandbox. We also assume that attackers may use speculative side channel attacks (e.g., Spectre) to read data within a renderer process. We consider attackers that want to steal information or abuse privileges granted to other web sites.

Requirements
To support a site-per-process policy in a multi-process web browser, we need to identify the smallest unit that cannot be split into multiple processes. This is not actually a single page, but rather a group of documents from the same web site that have references to each other. Such documents have full script access to each other's content, and they must run on a single thread, not concurrently. This group may span multiple frames or tabs, and they may come from multiple sub-domains of the same site.

When hackers look BEHIND the routers they have commandeered:
Hacker Steals Military Docs Because Someone Didn't Change a Default FTP Password
https://www.bleepingcomputer.com/news/security/hacker-steals-military-docs-because-someone-didn-t-change-a-default-ftp-password/

The security firm "Recorded Future" has discovered sensitive military documents being offered for sale on hacker forums.

BleepingComputer reported that some of the sensitive documents put up for sale include maintenance course books for servicing MQ-9 Reaper drones, various training manuals describing comment deployment tactics for improvised explosive device (IED), an M1 ABRAMS tank operation manual, a crewman training and survival manual, and a document detailing tank platoon tactics.

And only asking between $150 and $200.

Recorded Future says it engaged the hacker online and discovered that he used Shodan to hunt down Netgear Nighthawk R7000 routers that are known to use a default FTP password. The hacker then used this FTP password to gain access to some of these routers whose owners had not thought to change the default... some of which were located in military facilities.

Based on the documents and details he shared online and with researchers in private conversations, one such location was the 432nd Aircraft Maintenance Squadron Reaper AMU OIC, stationed at the Creech AFB in Nevada.

Here, writes Bleeping Computer, he used his access to the router to pivot inside the base's network and gain access to a captain's computer, from where he stole the MQ-9 Reaper manual and a list of airmen assigned to Reaper AMU.

MQ-9 Reaper drones are some of the most advanced drones around and are used by the US Air Force, the Navy, the CIA, the Customs and Border Protection Agency, NASA, and the militaries of other countries.

And as for the routers.... The issue with Netgear routers using a set of default FTP credentials has been known since 2016 when a security researcher raised the alarm about it. Netgear responded by putting up a support page with information on how users could change their routers' default FTP password.

Recorded Future said that at the time of writing, there are more than 4,000 such routers (Netgear Nighthawk R7000) available online via "smart device" search engines like Shodan.

Our listeners may herd me say several times that those who have insecure routers should be counting their blessings that hackers have been, for the most part, using vulnerable routers as packet forwarding devices... but that eventually hackers were going to start looking inside the networks behind those routers to see what goodies might be available.

When BGP abuse becomes sufficiently blatant…
… the very slow to respond Internet managers finally DO respond.

((A refresher on BGP and Route Advertising))

There have been times when “mistakes were made”...

But it would also be possible for IP address space to be “borrowed” and provided to spammers and others to mask their identities.

A Portugal-based data center and Internet transit provider named Bitcanal has such a long record of abuse, dating back years, that the rest of the Internet finally banded together and removed Bitcanal from their networks.

NANOG (North American Network Operators' Group)

Ronald F. Guilmette started his NANOG message: "I mean seriously, WTF?" ...

> *As should be blatantly self-evident to pretty much everyone who has ever looked at any of the Internet's innumerable prior incidents of very deliberately engineered IP space hijackings, all of the routes currently being announced by AS3266 (Bitcanal, Portugal) except for the ones in 213/8 are bloody obvious hijacks.*
>
> *That's 39 deliberately hijacked routes, at least going by the data visible on bgp.he.net. But even that data from bgp.he.net dramatically understates the case, I'm sorry to say. According to the more complete and up-to-the-minute data that I just now fetched from RIPEstat, the real number of hijacked routes is more on the order of 130 separate hijacked routes for a total of 224,512 IPv4 addresses:*
>
> *https://pastebin.com/raw/Jw1my9Bb*
>
> *In simpler terms, Bitcanal has made off with the rough equivalent of an entire /14 block of IPv4 addresses that never belonged to them. (And of course, they haven't paid a dime to anyone for any of that space.)*

Guilmette alleges that Bitcanal is doing all of this —hijacking BGP routes— for the purpose of re-selling the hijacked IP addresses to spammer groups, which in turn use them to send out new spam campaigns from IPs not found in spam blacklists.

SpinRite

John Doe writing from "Moonbase 17"
Subject: MDADM Raid 6 Spinrite Comment
Date: 12 Jul 2018 08:10:02
:
Hopefully this gets to you in time, or you can forward it onto the person that asked the question.

On your last podcast (SN 671) there was a question about running Spinrite on a 5 disk raid 6 with mdadm on Linux. If the person turns on write-intent bitmaps then the rebuild time can often be reduced to a few minutes when a drive is put back into the raid (I see 5-10 minute resyncs).

https://raid.wiki.kernel.org/index.php/Write-intent_bitmap

When an array has a write-intent bitmap, a spindle (a device, often a hard drive) can be removed and re-added, then only blocks changes since the removal (as recorded in the bitmap) will be resynced.

Therefore a write-intent bitmap reduces rebuild/recovery (md sync) time if:

- the machine crashes (unclean shutdown)
- one spindle is disconnected, then reconnected

If one spindle fails and has to be replaced, a bitmap makes no difference.

Write-intent bitmap support is only available for RAID geometries causing data redundancy. For example: as RAID0 has no redundancy it cannot be inconsistent, so there is nothing to record in such a bitmap.

All Up In Their Business

https://www.justice.gov/file/1080281/download

Russia's intelligence agents:

- VIKTOR BORISOVICH NETYKSHO
- BORIS ALEKSEYEVICH ANTONOV
- DMITRIY SERGEYEVICH BADIN
- IVAN SERGEYEVICHY ERMAKOV
- ALEKSEY VIKTOROVICH LUKASHEV
- SERGEY ALEKSANDROVICH MORGACHEV
- NIKOLAY YURYEVICH KOZACHEK
- PAVEL VYACHESLAVOVICH YERSHOV
- ARTEM ANDREYEVICH MALYSHEV
- ALEKSANDR VLADIMIROVICH OSADCHUK
- ALEKSEY ALEKSANDROVICH POTEMKIN
- ANATOLIY SERGEYEVICH KOVALEV

COUNT ONE (Conspiracy to Commit an Offense Against the United States):

1. In or around 2016, the Russian Federation ("Russia") operated a military intelligence agency called the Main Intelligence Directorate of the General Staff ("GRU"). The GRU had multiple units, including Units 26165 and 74455, engaged in cyber operations that involved the staged releases of documents stolen through computer intrusions. These units conducted large-scale cyber operations to interfere with the 2016 U.S. presidential election.

2. Defendants [ABOVE] were GRU officers who knowingly and intentionally conspired with each other, and with persons known and unknown to the Grand Jury (collectively the "Conspirators"), to gain unauthorized access (to "hack") into the computers of U.S. persons and entities involved in the 2016 U.S. presidential election, steal documents from those computers, and stage releases of the stolen documents to interfere with the 2016 U.S. presidential election.

3. Starting in at least March 2016, the Conspirators used a variety of means to hack the email accounts of volunteers and employees of the U.S. presidential campaign of Hillary Clinton (the "Clinton Campaign"), including the email account of the Clinton Campaign's chairman.

4. By in or around April 2016, the Conspirators also hacked into the computer networks of the Democratic Congressional Campaign Committee ("DCCC") and the Democratic National Committee ("DNC"). The Conspirators covertly monitored the computers of dozens of DCCC and DNC employees, implanted hundreds of files containing malicious computer code ("malware"), and stole emails and other documents from the DCCC and DNC.

---

9. Defendant VIKTOR BORISOVICH NETYKSHO was the Russian military officer in command of Unit 26165, located at 20 Komsomolskiy Prospekt, Moscow, Russia. Unit 26165 had primary responsibility for hacking the DCCC and DNC, as well as the email accounts of individuals affiliated with the Clinton Campaign.

10. Defendant BORIS ALEKSEYEVICH ANTONOV was a Major in the Russian military assigned to Unit 26165. ANTONOV oversaw a department within Unit 26165 dedicated to targeting military, political, governmental, and non-governmental organizations with spearphishing emails and other computer intrusion activity. ANTONOV held the title "Head of Department." In or around 2016, ANTONOV supervised other co-conspirators who targeted the DCCC, DNC, and individuals affiliated with the Clinton Campaign.

11. Defendant DMITRIY SERGEYEVICH BADIN was a Russian military officer assigned to Unit 26165 who held the title "Assistant Head of Department." In or around 2016, BADIN, along with ANTONOV, supervised other co-conspirators who targeted the DCCC, DNC, and individuals affiliated with the Clinton Campaign.

(It continues like that, explaining who each of these people are, their job responsibilities and and the roles they played.)

ANTONOV, BADIN, YERMAKOV, LUKASHEV, and their co-conspirators targeted victims using a technique known as spearphishing to steal victims' passwords or otherwise gain access to their computers. Beginning by at least March 2016, the Conspirators targeted over 300 individuals affiliated with the Clinton Campaign, DCCC, and DNC.

For example, on or about March 19, 2016, LUKASHEV and his co-conspirators created and sent a spearphishing email to the chairman of the Clinton Campaign. LUKASHEV used the account "john356gh" at an online service that abbreviated lengthy website addresses (referred to as a "URL-shortening service"). LUKASHEV used the account to mask a link contained in the spearphishing email, which directed the recipient to a GRU-created website. LUKASHEV altered the appearance of the sender email address in order to make it look like the email was a security notification from Google (a technique known as "spoofing"), instructing the user to change his password by clicking the embedded link. Those instructions were followed. On or about March 21, 2016, LUKASHEV, YERMAKOV, and their co-conspirators stole the contents of the chairman's email account, which consisted of over 50,000 emails.

Starting on or about March 19, 2016, LUKASHEV and his co-conspirators sent spearphishing emails to the personal accounts of other individuals affiliated with the Clinton Campaign, including its campaign manager and a senior foreign policy advisor. On or about March 25, 2016, LUKASHEV used the same john356gh account to mask additional links included in spearphishing emails sent to numerous individuals affiliated with the Clinton Campaign, including Victims 1 and 2. LUKASHEV sent these emails from the Russia-based email account hi.mymail@yandex.com that he spoofed to appear to be from Google.

On or about March 28, 2016, YERMAKOV researched the names of Victims 1 and 2 and their association with Clinton on various social media sites. Through their spearphishing operations, LUKASHEV, YERMAKOV, and their co-conspirators successfully stole email credentials and thousands of emails from numerous individuals affiliated with the Clinton Campaign. Many of these stolen emails, including those from Victims 1 and 2, were later released by the Conspirators through DCLeaks.

On or about April 6, 2016, the Conspirators created an email account in the name (with a one-letter deviation from the actual spelling) of a known member of the Clinton Campaign. The Conspirators then used that account to send spearphishing emails to the work accounts of more than thirty different Clinton Campaign employees. In the spearphishing emails, LUKASHEV and his co-conspirators embedded a link purporting to direct the recipient to a document titled "hillary-clinton-favorable-rating.xlsx." In fact, this link directed the recipients' computers to a GRU-created website.

Beginning in or around March 2016, the Conspirators, in addition to their spearphishing efforts, researched the DCCC and DNC computer networks to identify technical specifications and vulnerabilities.

- a. For example, beginning on or about March 15, 2016, YERMAKOV ran a technical query for the DNC's internet protocol configurations to identify connected devices.

- b. On or about the same day, YERMAKOV searched for open-source information about the DNC network, the Democratic Party, and Hillary Clinton.

- c. On or about April 7, 2016, YERMAKOV ran a technical query for the DCCC's internet protocol configurations to identify connected devices.

By in or around April 2016, within days of YERMAKOV's searches regarding the DCCC, the Conspirators hacked into the DCCC computer network. Once they gained access, they installed and managed different types of malware to explore the DCCC network and steal data.

a. On or about April 12, 2016, the Conspirators used the stolen credentials of a DCCC Employee ("DCCC Employee 1") to access the DCCC network. DCCC Employee 1 had received a spearphishing email from the Conspirators on or about April 6, 2016, and entered her password after clicking on the link.

b. Between in or around April 2016 and June 2016, the Conspirators installed multiple versions of their X-Agent malware on at least ten DCCC computers, which allowed them to monitor individual employees' computer activity, steal passwords, and maintain access to the DCCC

network.

c. X-Agent malware implanted on the DCCC network transmitted information from the victims' computers to a GRU-leased server located in Arizona. The Conspirators referred to this server as their "AMS" panel. KOZACHEK, MALYSHEV, and their co-conspirators logged into the AMS panel to use X-Agent's keylog and screenshot functions in the course of monitoring and surveilling activity on the DCCC computers. The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees. The screenshot function allowed the Conspirators to take pictures of the DCCC employees' computer screens.

d. For example, on or about April 14, 2016, the Conspirators repeatedly activated X-Agent's keylog and screenshot functions to surveil DCCC Employee 1's computer activity over the course of eight hours. During that time, the Conspirators captured DCCC Employee 1's communications with co-workers and the passwords she entered while working on fundraising and voter outreach projects. Similarly, on or about April 22, 2016, the Conspirators activated X-Agent's keylog and screenshot functions to capture the discussions of another DCCC Employee ("DCCC Employee 2") about the DCCC's finances, as well as her individual banking information and other personal topics.

25. On or about April 19, 2016, KOZACHEK, YERSHOV, and their co-conspirators remotely configured an overseas computer to relay communications between X-Agent malware and the AMS panel and then tested X-Agent's ability to connect to this computer. The Conspirators referred to this computer as a "middle server." The middle server acted as a proxy to obscure the connection between malware at the DCCC and the Conspirators' AMS panel.

26. On or about April 18, 2016, the Conspirators hacked into the DNC's computers through their access to the DCCC network. The Conspirators then installed and managed different types of malware (as they did in the DCCC network) to explore the DNC network and steal documents.

a. On or about April 18, 2016, the Conspirators activated X-Agent's keylog and screenshot functions to steal credentials of a DCCC employee who was authorized to access the DNC network. The Conspirators hacked into the DNC network from the DCCC network using stolen credentials. By in or around June 2016, they gained access to approximately thirty-three DNC computers.

b. In or around April 2016, the Conspirators installed X-Agent malware on the DNC network, including the same versions installed on the DCCC network. MALYSHEV and his co-conspirators monitored the X-Agent malware from the AMS panel and captured data from the victim computers. The AMS panel collected thousands of keylog and screenshot results from the DCCC and DNC computers, such as a screenshot and keystroke capture of DCCC Employee 2 viewing the DCCC's online banking information.

28. To enable them to steal a large number of documents at once without detection, the Conspirators used a publicly available tool to gather and compress multiple documents on the DCCC and DNC networks. The Conspirators then used other GRU malware, known as "X-Tunnel," to move the stolen documents outside the DCCC and DNC networks through encrypted channels.

a. For example, on or about April 22, 2016, the Conspirators compressed gigabytes of data from

DNC computers, including opposition research. The Conspirators later moved the compressed DNC data using X-Tunnel to a GRU-leased computer located in Illinois.

b. On or about April 28, 2016, the Conspirators connected to and tested the same computer located in Illinois. Later that day, the Conspirators used X-Tunnel to connect to that computer to steal additional documents from the DCCC network.

29. Between on or about May 25, 2016 and June 1, 2016, the Conspirators hacked the DNC Microsoft Exchange Server and stole thousands of emails from the work accounts of DNC employees. During that time, YERMAKOV researched PowerShell commands related to accessing and managing the Microsoft Exchange Server.

31. During the hacking of the DCCC and DNC networks, the Conspirators covered their tracks by intentionally deleting logs and computer files. For example, on or about May 13, 2016, the Conspirators cleared the event logs from a DNC computer. On or about June 20, 2016, the Conspirators deleted logs from the AMS panel that documented their activities on the panel, including the login history.

32. Despite the Conspirators' efforts to hide their activity, beginning in or around May 2016, both the DCCC and DNC became aware that they had been hacked and hired a security company ("Company 1") to identify the extent of the intrusions. By in or around June 2016, Company 1 took steps to exclude intruders from the networks. Despite these efforts, a Linux-based version of X-Agent, programmed to communicate with the GRU-registered domain linuxkrnl.net, remained on the DNC network until in or around October 2016.

a. On or about May 31, 2016, YERMAKOV searched for open-source information about Company 1 and its reporting on X-Agent and X-Tunnel. On or about June 1, 2016, the Conspirators attempted to delete traces of their presence on the DCCC network using the computer program CCleaner.

34. In or around September 2016, the Conspirators also successfully gained access to DNC computers hosted on a third-party cloud-computing service. These computers contained test applications related to the DNC's analytics. After conducting reconnaissance, the Conspirators gathered data by creating backups, or "snapshots," of the DNC's cloud-based systems using the cloud provider's own technology. The Conspirators then moved the snapshots to cloud-based accounts they had registered with the same service, thereby stealing the data from the DNC.

35. More than a month before the release of any documents, the Conspirators constructed the online persona DCLeaks to release and publicize stolen election-related documents. On or about April 19, 2016, after attempting to register the domain electionleaks.com, the Conspirators registered the domain dcleaks.com through a service that anonymized the registrant. The funds used to pay for the dcleaks.com domain originated from an account at an online cryptocurrency service that the Conspirators also used to fund the lease of a virtual private server registered with the operational email account dirbinsaabol@mail.com. The dirbinsaabol email account was also used to register the john356gh URL-shortening account used by LUKASHEV to spearphish the Clinton Campaign chairman and other campaign-related individuals.

40. On or about June 14, 2016, the DNC—through Company 1—publicly announced that it had

been hacked by Russian government actors. In response, the Conspirators created the online persona Guccifer 2.0 and falsely claimed to be a lone Romanian hacker to undermine the allegations of Russian responsibility for the intrusion.

41. On or about June 15, 2016, the Conspirators logged into a Moscow-based server used and managed by Unit 74455 and, between 4:19 PM and 4:56 PM Moscow Standard Time, searched for certain words and phrases, including: "some hundreds of sheets" / dcleaks / illuminati / "worldwide known" / "think twice about" / "company's competence"

42. Later that day, at 7:02 PM Moscow Standard Time, the online persona Guccifer 2.0 published its first post on a blog site created through WordPress. Titled "DNC's servers hacked by a lone hacker," the post used numerous English words and phrases that the Conspirators had searched for earlier that day (bolded below):

45. The Conspirators conducted operations as Guccifer 2.0 and DCLeaks using overlapping computer infrastructure and financing.

a. For example, between on or about March 14, 2016 and April 28, 2016, the Conspirators used the same pool of bitcoin funds to purchase a virtual private network ("VPN") account and to lease a server in Malaysia. In or around June 2016, the Conspirators used the Malaysian server to host the dcleaks.com website. On or about July 6, 2016, the Conspirators used the VPN to log into the @Guccifer_2 Twitter account. The Conspirators opened that VPN account from the same server that was also used to register malicious domains for the hacking of the DCCC and DNC networks.

57. To facilitate the purchase of infrastructure used in their hacking activity—including hacking into the computers of U.S. persons and entities involved in the 2016 U.S. presidential election and releasing the stolen documents—the Defendants conspired to launder the equivalent of more than $95,000 through a web of transactions structured to capitalize on the perceived anonymity of cryptocurrencies such as bitcoin.

58. Although the Conspirators caused transactions to be conducted in a variety of currencies, including U.S. dollars, they principally used bitcoin when purchasing servers, registering domains, and otherwise making payments in furtherance of hacking activity. Many of these payments were processed by companies located in the United States that provided payment processing services to hosting companies, domain registrars, and other vendors both international and domestic. The use of bitcoin allowed the Conspirators to avoid direct relationships with traditional financial institutions, allowing them to evade greater scrutiny of their identities and sources of funds.

59. All bitcoin transactions are added to a public ledger called the Blockchain, but the Blockchain identifies the parties to each transaction only by alpha-numeric identifiers known as bitcoin addresses. To further avoid creating a centralized paper trail of all of their purchases, the Conspirators purchased infrastructure using hundreds of different email accounts, in some cases using a new account for each purchase. The Conspirators used fictitious names and addresses in order to obscure their identities and their links to Russia and the Russian government. For example, the dcleaks.com domain was registered and paid for using the fictitious name "Carrie Feehan" and an address in New York. In some cases, as part of the payment process, the

Conspirators provided vendors with nonsensical addresses such as "usa Denver AZ," "gfhgh ghfhgfh fdgfdg WA," and "1 2 dwd District of Columbia."

60. The Conspirators used several dedicated email accounts to track basic bitcoin transaction information and to facilitate bitcoin payments to vendors. One of these dedicated accounts, registered with the username "gfadel47," received hundreds of bitcoin payment requests from approximately 100 different email accounts. For example, on or about February 1, 2016, the gfadel47 account received the instruction to "[p]lease send exactly 0.026043 bitcoin to" a certain thirty-four character bitcoin address. Shortly thereafter, a transaction matching those exact instructions was added to the Blockchain.

61. On occasion, the Conspirators facilitated bitcoin payments using the same computers that they used to conduct their hacking activity, including to create and send test spearphishing emails. Additionally, one of these dedicated accounts was used by the Conspirators in or around 2015 to renew the registration of a domain (linuxkrnl.net) encoded in certain X-Agent malware installed on the DNC network.