Security Now! #929 - 06-27-23

Operation Triangulation

This week on Security Now!

Today's podcast is chock full of news. What has DuckDuckGo just announced? What about the Tor Project? Has Opera just made a big mistake? What is the KasperskyOS? What's happening to non-Russian web hosting for Russians? Are SolarWinds executives finally going to be held to account? We now have the US Space Force, what's next? What's the latest large site to support Passkeys? Who would like permission to spy on their own citizens? Which facial recognition smartphone unlocking can you trust and which should not be? And what was the inevitable shoe to drop following last week's coverage of the Massive MOVEit Transfer mess? Then, after sharing a bit of listener feedback, we're going to take a much closer look into Kaspersky's discovery of a pervasive 4-year iPhone spyware campaign.

When Apple creates a "Walled Garden" they don't mess around!

Security Catch Up

Leo Does Disneyland

Before we plow into the past week's news, we need to bring Leo up to speed on the four things he missed last week while Jason was co-hosting. First of all, last week's picture of the week was just too fun to miss:

If it's not tied down...

Second, the team of brilliant researchers at Israel's Ben-Gurion University of the Negev (who turn out to be Security Now! listeners) showed us that unless explicit measures were taken to isolate a device's power LED from the power consumption fluctuations created during the processing of cryptographic secrets, those secrets could be extracted locally or at a distance, by carefully examining the power LED, or even after-the-fact in a video recording of the power LED.

Next, the 25-year-old persistent SQL injection vulnerability struck once again in the widely used MOVEit file management system. This resulted in thousands of organizations' internal private data being exfiltrated to Russia where extortion demands by the CLOP gang were then made.

And, finally, it appears that the development of the DOS side of SpinRite v6.1 is wrapping up. Sunday before last I posted the 29th Alpha release with the assertion that we were done. That resulted in the discovery of four small, mostly cosmetic, bugs which were resolved in this past Sunday's 30th release. So I'm at work on the SpinRite FAQ document which will be built into the final code while I am actively encouraging any final pre-release testing.

Security News

DuckDuckBrowse

Joining the MacOS browser they launched last year, DuckDuckGo now has their Windows browser in public beta. As we would expect from the privacy-first search folks, the DuckDuckBrowser — I sure hope that's not what they name the thing — is privacy first.

- It sports *"Duck Player,"* a YouTube player that allows viewing YouTube videos without privacy-invading tracking ads and prevents videos viewed from impacting future recommendations.

- They claim that the browser's tracker blocking goes above and beyond what's available from Chrome and other browsers, writing: *"Our 3rd-party Tracker Loading Protection, for example, blocks the hidden trackers from companies like Google and Facebook lurking on other websites before they get a chance to load."*

- It's unclear what this means. They wrote: *"Smarter Encryption to ensure that more of the websites you visit and the links you click are encrypted, relative to other browsers."* — I guess they say they're being more clever about choosing HTTPS alternatives when they are available.

- Here's something that might be worth the price of admission (which, being zero, admittedly sets the bar rather low): Cookie Pop-up Management, a tool that automatically selects the most private options available and hides cookie consent pop-ups. Okay, I want that.

- The *"Fire"* Button (as in lighting a fire), burns recent browsing data in one click. There's also a *"Fireproof"* option for any sites you want to stay logged into. I suppose if you name your privacy-centric search service *"DuckDuckGo"* you've already lowered expectations about the names you're going to use for things. But somehow the idea of a web browser having a *"burn bag"* into which websites are tossed by pressing the *"Fire"* button – to light them on fire and reduce them to ashes – unless you have *"Fireproofed"* them ahead of time... I don't know.

- The browser also offers built-in email protection to hide user email addresses behind uniquely generated @duck.com addresses when signing up online. That sounds handy, though that would also create some quite powerful lock-in effects.

The beta of the browser, which apparently goes by the catchy name *"DuckDuckGo for Windows"* is available from https://duckduckgo.com/windows. And they note that switching is easy since the browser can import bookmarks and passwords from other browsers and password managers. Their announcement had a couple of additional interesting things to say. They wrote:

> The browser doesn't have extension support yet, but we plan to add it in the future. In the meantime, we've built the browser to include features that meet the same needs as the most popular extensions: ad-blocking and secure password management.

Secure password management: Our browser includes our own secure and easy-to-use password manager that can automatically remember and fill in login credentials. DuckDuckGo for Windows can now also suggest secure passwords for new logins. This will get even more convenient soon when we roll out private syncing across devices, so you'll be able to sync your bookmarks and saved passwords between different devices, whether you're using a DuckDuckGo browser on Windows, iOS, Android, or Mac.

Ad blocking: DuckDuckGo for Windows is equipped with our privacy-protecting alternative to ad blockers: the browser blocks invasive trackers before they load, effectively eliminating ads that rely on creepy tracking. (Because so many ads work that way, you'll see way fewer ads – if any at all.) We also remove the whitespace left behind by those ads for a clean, distraction-free look without the need for an outside ad blocker. That sounds good.

Duck Player, our browser's more-private way to watch YouTube: This built-in video player protects you from tracking cookies and personalized ads with a distraction-free interface that incorporates YouTube's strictest privacy settings for embedded video. (In our testing, by blocking the trackers behind personalized ads, Duck Player prevented ads from loading on most videos altogether.) YouTube still logs video views, so it's not completely anonymous, but none of the videos you watch in Duck Player contribute to your personalized recommendations or your YouTube advertising profile. You can leave the feature always-on, or opt in on individual videos.

And I thought what was most interesting was that this recently created browser was not simply window dressing surrounding the Chromium engine – as are pretty much everyone else's web browser, including Microsoft's own Edge. They explained:

DuckDuckGo for Windows was built with your privacy, security, and ease of use in mind. It's not a "fork" of any other browser code; all the code, from tab and bookmark management to our new tab page to our password manager, is written by our own engineers. For web page rendering, the browser uses the underlying operating system rendering API. (In this case, it's a Windows WebView2 call that utilizes the Blink rendering engine underneath.)

Our default privacy protections are stronger than what Chrome and most other browsers offer, and our engineers have spent lots of time addressing any privacy issues specific to WebView2, such as ensuring that crash reports are not sent to Microsoft.

Leo: Since Paul Thurrott appears to have an interest in exploring the experiences and features offered by various web browsers, perhaps, when the subject of web browsing next comes up, you could mention "DuckDuckGo for Windows" and see whether he's ready to switch again! :)

And an updated Tor Browser:
While we're on the subject of web browsers, I'll note for the benefit of any of our Tor browser users that version 12.5 has just been released. It sports a bunch of UI improvements, including a redesigned visualization of the Tor circuit which shows the Tor onion router hops between you and whatever site you're visiting.

Opera, now enhanced with "AI"

Okay. One more browser update: Not long ago everything was *"blockchain this"* and *"blockchain that"*. Blockchain was the magic pixie dust that was being sprinkled on everything to make it *"more better"*. Today, that role has been taken up by AI. So I suppose it shouldn't surprise anyone that every other word in Opera's announcement of their *"totally rebuilt from the ground up"* all new web browser is "AI". Last Tuesday they posted:

> *Hey Opera fans! Today we're excited to drop the big news that **Opera One**, the latest incarnation of the Opera Browser, is here and ready for you to download!*
>
> *Here's the scoop – **Opera One** is your familiar Opera Browser, but it's been given a major makeover. And we're not just talking about a new coat of paint – we've reimagined and rebuilt Opera from the ground up, paving the way for a new era in which AI isn't just an add-on, but a core part of your browsing experience. **So, what's actually new?** Well, for starters, Opera One is introducing **Aria**, the first-ever native browser AI. There's also a totally fresh Modular Design and a bunch of game-changing features like Tab Islands, ingrained within the browser.*

I'm not going to spend any more time on this. And from the comments in the announcement's posting, the totally new look, feel, and AI are not going over very well with existing Opera users. Big changes always risk that, especially in any product which is so much about the user interface experience as any web browser.

The KasperskyOS Phone

As we've reported, the Kremlin in Russia is now moving away as quickly as possible from Western made smartphones. So it only makes sense that they would turn to their own well regarded Kaspersky for a solution. To that end, Kaspersky has previewed the first version of their KasperskyOS, a "hack-resistant" mobile-targeted operating system they've been developing for the past several years. It was demonstrated at a business conference held in Saint Petersburg earlier this month, with the initial version equipped with a bare bones set of basic applications for calling, SMS messaging, an address book, and a settings panel. Kaspersky says it's currently working on adding a Chromium-based web browser and support for a camera, Wi-Fi, and NFC features. They're seeking a partnership with a hardware smartphone vendor to produce a finished product which will eventually be made available on Russia's internal market.

The cost of doing business in Russia

And while we're on the subject of Russia, the cost of doing web hosting business in Russia just increased. So I suppose that means that the cost of web hosting to Russian citizens located within Russia will also be increasing as those costs are passed along.

Last Thursday, our favorite Russian Internet watchdog, **Roskomnadzor** named the 12 largest and most popular Internet hosting companies who must participate in some new legislation. I had Google translate Roskomnadzor's announcement from Russian. According to the legislation, foreign hosting providers, whose users are located, among other things, on the territory of the

Russian Federation, are subject to Federal Law No. 236-FZ which is titled *"On the activities of foreign persons on the Internet in the territory of the Russian Federation"*.

Inclusion in the list of entities imposes obligations on foreign hosting providers to open a branch, a representative office, or some legal Russian entity in Russia, post an electronic feedback form for Russian users on their website, and register an account on the Roskomnadzor website for interaction with local Russian authorities. Failure to comply with the legislation risks the imposition of fines or even access blocked to their infrastructure. The list is pretty much the Who's Who of Internet hosting: AWS, DigitalOcean, GoDaddy, HostGator, DreamHost, Bluehost, Hetzner, WPEngine, Network Solutions, Ionos, FastComet, and Kamatera.

Slowly turn the wheels of justice

SolarWinds has said that some of its current and former executives have received what's known as a Wells notice from the US Securities and Exchange Commission in their roll of overseer of publicly traded companies. The notice in this case, of course, is in connection with the company's quite devastating 2020 security incident — which is why we all know the name "SolarWinds." A Wells notice is a letter the SEC sends to companies when the agency is planning to bring an enforcement action against them. SolarWinds says the SEC may fine or bar some executives from serving as officers or directors of public companies.

The US to create a new "Cyber Force"

Last Friday, the Senate Armed Services Committee announced that it will be formally exploring the idea of creating a new dedicated Cyber Force branch of the US military, similar to and having equal standing alongside our existing Army, Navy, Air Force, Marine Corps, Coast Guard, National Guard, and Space Force. To further this, a provision has been added to the 2024 National Defense Authorization Act calling for an assessment of creating such a dedicated Cyber Force branch.

Leo: What I want to know is why do these photos of US Cyber Defense always show guys with shaved heads, sitting in front of their screens and keyboards, dressed up in full cammo?? Is this an attempt to avoid being seen by the webcam? Because if so, I don't think it works the way they think, and this doesn't bode well for the judgment of the US Cyber Force. A yellow PostIt note can be far more effective for that!

Apple.com now supports Passkeys

Just a quick note that Apple has added Passkeys support for logging onto Apple.com. You'll need to either wait for the formal release of iOS 17, iPadOS 17 or MacOS Sonoma, or be using the beta of those. But that support is now there. I suppose that other Passkeys clients should work now, too.

Selective GDPR enforcement?

Get a load of this tidbit: Several European governments, specifically French, German, and Dutch officials, are pushing the EU to add an exemption in its upcoming European Media Freedom Act (EMFA) which would explicitly allow EU member states to continue spying on the electronic communications of journalists under the guise of "national security." The push follows the results of the EU's own PEGA commission which advised the EU to head in the opposite direction by adding additional safeguards to protect democracy and the rule of law in the EU against the abuse of spyware tools. In their report published last year, the PEGA commission said several EU countries were abusing surveillance technologies to illegally spy on their own citizens, including journalists, under murky and vague "national security" justifications. More than 60 journalistic organizations and civil society groups have signed a joint letter to the EU Council advising against weakening the upcoming law and giving governments an explicit spying carte-blanche. So, yeah... everyone else gets constrained by the GDPR, but the governments behind the GDPR are seeking to legislate a loophole to allow themselves to use spyware, which is itself, let's not forget, illegal malicious software. Nice.

Facial Recognition is Photo Recognition

This may be obvious to everyone, but I think it's still worth reminding everyone that because Apple did a beautiful job and got the whole facial recognition challenge correct, that fact should in no way confer **any** presumption that anyone else did the same.

A recent study, updating an earlier study four years ago, concluded that with the sole exceptions of Apple and Samsung, the phrase "smartphone facial recognition security" is an oxymoron. The updated research conducted by a Dutch consumer protection association found that facial recognition systems on most of today's mid to upper-tier smartphones can be bypassed using a simple photo. Researchers bypassed facial recognition on 26 different smartphone models by showing the phones a photo of its owner. Only Apple and Samsung devices were found to be secure. Researchers were unable to bypass facial recognition on any of Apple's iPhones, and only one out of 12 Samsung models failed the test. Fourteen of the 26 smartphones that failed the test are Xiaomi models. Among the failures were Motorola Moto's, Nokia's, a OnePlus, two OPPO's, that one Samsung Galaxy A04s, and a ton of Xiaomi's.

When Apple first unveiled their facial recognition, the first thing that naturally occurred to all of us was to wonder how easily their technology could be spoofed. What we learned was that the phone projects a scanning dotted grid out which is viewed by offset cameras to determine whether what's being presented to it matches the model of the 3D face that was created and mapped when the phone user's face was originally presented to it and deliberately moved around to create that map. While this system can still be spoofed by creating 3D replicas of the user's face, no simple-to-create flat photo will do the job.

So, I wanted to remind everyone that just because Apple went to the extreme measures to create a highly spoof-resistant facial recognition and unlocking technology, no one should assume that anyone else took the time to get it right. Since getting it wrong is far far easier to do, that's what is typically done. The danger is that other phone manufacturers will ride on Apple's coattails after Apple showed the world that facial recognition can be sufficiently secure. Yes, it can be. But it isn't necessarily or automatically so.

Google cybersecurity clinics

Google has committed more than $20 million to the creation of cybersecurity clinics at 20 higher education institutions across the US. The clinics will provide free cybersecurity training and hands-on experience for thousands of students. Some Google employees will serve as mentors and trainers at some of the clinics. Google will also provide free scholarships to allow some students to attend its Cybersecurity Certificate program. Google said: "These clinics provide free security services in the same way law or medical schools offer free clinics in their communities. They give students the opportunity to learn and improve their skills, while helping to protect critical infrastructure such as hospitals, schools and energy grids."

This sounds like a great idea, though I'll admit that the cynic in me wonders whether this might not also be a terrific means for recruiting talent from those institutions. Not that there's anything at all wrong with doing so. After all, the reason those students are there is to acquire the knowledge and skill necessary to find gainful employment.

Progress/MOVEit sued

I suppose it was inevitable that the subject of last week's *"Massive MOVEit Malestrom"* podcast, Progress Software, would soon be facing lawsuits. And, sure enough, at least two federal class action lawsuits have been filed so far in connection with the devastating SQL injection vulnerability which was discovered and widely exploited, as we covered in detail last week, in the MOVEit Transfer software. The lawsuits allege that it was the company's negligence which led to the breach, thus putting their personal financial data at risk.

The first suit filed on June 15 in U.S. District Court for the Eastern District of Louisiana alleges that the vulnerability led to the breach of the state Office of Motor Vehicles. Louisiana State officials announced the breach the same day, warning all Louisiana motor vehicle drivers that their names, addresses, dates of birth, driver's license numbers, social security numbers, vehicle registrations and other information was likely stolen. Pretty much the whole enchilada. About 6 million records were exposed and likely stolen.

The plaintiff, Orleans Parish resident Jason Berry, alleges his personal data was put at risk by the breach. He alleges that the company also failed to promptly notify potential victims of the risk of exposing their personal information. The suit seeks class action status for others impacted by the breach.

As we were recording last week's podcast on this topic, the second case was being filed in the U.S. District Court for the District of Massachusetts on behalf of three Louisiana residents: Shavonne Diggs, and Brady and Christina Bradberry. The class exceeds 100 people and the plaintiffs are seeking upwards of $5 million, according to the complaint. This second Massachusetts case alleges that Progress Software failed to adhere to Federal Trade Commission guidelines for data security, failed to protect customer data and failed to properly monitor its own internal systems.

I don't have any opinion about this one way or the other. One issue may be that the plaintiffs need to be more than just upset over the news of this happening. At this point they may just be chasing ambulances. I suspect that they need to demonstrate that they have been individually and collectively harmed by the breach. And that may not be easy. The CLOP gang extortionists did say that they wanted nothing to do with government, educational or police agencies and that any data obtained from them would be immediately deleted. So I hope that Progress Software's attorneys are up to speed on that.

Everyone knows quite well that I have no sympathy whatsoever for anyone who designs web server software in such a way that it feeds any user-provided text to a back-end SQL database which stupidly mixes commands and query text in the same text stream. Anyone who is still doing that 25 years after it was first observed to be a really bad idea, and with it being consistently the top vulnerability on OWASP's top 10 list of really bad ideas, is probably going to get what they deserve.

But we don't know, in sufficient detail, how this happened. Back in November of 2015, when Marriott International acquired Starwood Hotels & Resorts, the Marriott execs didn't know that Starwood's network was hosting some serious security vulnerabilities. And three years later, in September of 2018, that oversight came back to bite them hard. Should Marriott have done an in-depth security verification? Yes. And perhaps they did. If vulnerabilities were not extremely difficult to find they would all be eliminated before software was shipped and the entire bug bounty industry and Pwn2Own competitions would not exist.

So in this case of MOVEit and Progress Software, I don't feel any sense of schadenfreude. This is a tragedy all around where everyone has lost. Our listeners know that I always completely separate mistakes from policies. So my only argument here is that the use of SQL in this way, in any way that opens the door for injection, is a POLICY decision. It was a mistake that this policy was not implemented perfectly. But if this database architecture policy had not been used at all in the first place, then there would have been no reliance upon the filtering code needing to be perfect.

Closing the Loop

David Scholten / @Palanthas86

@SGgrc I have loved listening to SN over the last 10+ years and believe it has helped me greatly in my IT career, from technician to IT admin. Now, I have a non-IT question. Is it just me, or have I been hearing a fire alarm low battery beep in the background in several podcasts?

I dearly wish it was your imagination, David. But it's not. Something in my environment started beeping occasionally many weeks ago and I have no idea what it is or where it is. It's not any of my smoke detectors and the room it's in is full of equipment so there are a great many places it might be. Since it began, I've embarked on several missions to locate and find it. But the chirp is so short that I don't get enough of a sample to obtain a bearing. And when I try to incrementally zero-in on it by moving near to where it seems to be, then waiting for the next beep, it appears to move elsewhere. I'm pretty much able to tune it out. But I've been conscious of the podcast. So I've been hoping that the directionality of the Heil microphone, which is pointed away from the room, would keep it from interfering. But several of our listeners have previously mentioned it and asked. So, yes... for the record, something in my environment is periodically beeping and I have no idea what it is or where it is. I'll find it eventually.

Fabian Santiago / @fabiansantiago

I'm still sore for and with you about sqrl (vs passkeys, etc). it does warm my heart to see the sqrl ios testflight client app still receiving updates though. (just today for me)

I wanted to take this opportunity to give Jeff Arthur, SQRL's iOS client author, a shout out and thanks for his continuing work on SQRL. I know that it's been a labor or love for him, and it would be terrific if something were to ever come of it. If FIDO2/WebAuthn & Passkeys evolves to require elliptic curve crypto as one of its available crypto suite options, **that** would immediately enable the use of SQRL-style deterministic – rather than random – private keys. And that would mean that **all** of the other work that's been done on SQRL to solve all of the other problems that today's Passkeys clients still have, would be immediately available, too.
So... we'll see how this evolves. All may not be lost.

David R Bunting / @dabunting

Jungle Disk- Do you still recommend it? Thanks, Steve!

No. JungleDisk was a very early and very good TNO (Trust No One) client-side encrypted cloud storage solution. They were purchased by something called "CyberFortress" and it appears that they've "gone corporate." My #1 favorite choice and recommendation is the Canadian firm and service SYNC.COM. You can get 5GB for free to see how you like it, or can you my referral code to start off with 6GB: https://grc.sc/sync. And you just create a username and password with no credit card or anything else required. I've been using them since August 7th of 2019, so approaching 4 years, and they are a total win. The only downside is that they don't support

Linux and although they know there's a demand for it, especially from our listeners, that's dwarfed by the interest in Windows and Mac... so I wouldn't hold my breath.

What I like most about SYNC is that it's probably the right solution for most people because it just works. When it's installed it creates a SYNC folder in the system's directory tree, and anything that's placed under there is kept fully and immediately backed up to the cloud. And, if you have multiple machines, all of their SYNC directories are kept fully cross-synchronized through the cloud. And all of this is done with full versioning. Using their web interface you're able to browse back 6 months to a year to obtain any previous version of anything that's been synchronized – even things you've deleted. There's also zero configuration about how often you want to sync. Everything is always synced. Using the Windows tray utility it's possible to select things you may not want to sync for some reason. So there is some optional flexibility there.

When I was deep into SpinRite work, all of my code and management scripts assumed that the ASM directory was at the C: drive's root. But to have it all backed up to the cloud and synchronized between machines, I needed it to be physically under the SYNC directory. So I moved the \ASM directory under the SYNC directory, then created a Windows NTFS junction link so that an apparent \ASM directory on the root would be aliased to the ASM directory under the SYNC directory. So nothing needed to change. Everything still referenced all of my code as if it was \ASM, even though it was actually under the SYNC directory.

Anyway, it all worked perfectly. And there have been several times when the full automatic retention of previous file versions has come in very handy!

So once again, if you're interested in their free trial, you can use https://grc.sc/sync which will bounce you over to them with my affiliate code appended to start you off with an extra 1GB for a total of 6GB. Then, after experiencing it – I just checked – their basic personal plan is $8 per month for 2TB of storage with more available as and if needed.

So that's my current and well-proven recommendation for a simple to use foolproof cloud storage solution if you don't need Linux clients. I should note that while I'm still using Sync for many things, I've switched to using a pair of cross-synchronized Synology NAS boxes and I am **so** impressed by Synology. I use a very nice free Windows utility called **@MAX SyncUp** to synchronize my machine's local directories to my local Synology NAS. **@Max SyncUp** is also a terrific solution if you want to synchronize to Google Drive.

And before we leave this discussion we should remind everyone of SyncThing, which is a terrific peer-to-peer cross-platform solution that's quite happy with Linux. I still have SyncThing running on a surviving Drobo, which is Linux based. That SyncThing instance is keeping my wife's fleet of remote Windows laptops synchronized in the field. And SyncThing really is a terrific peer-to-peer solution.

SpinRite

I thought I'd said all I had to say about SpinRite, until I caught up with my Twitter feed and found a very heart warming pair of Tweets from

☣crzy8ers☣ / @crzy8ers

Had a catastrophic hard drive failure… All my finished photos were ready for print… Thought my memories were lost forever... Until Someone on @Twitter recommended this software... SpinRite by: Mr. Steve Gibson @SGgrc Here is his website to find SpinRite Data Recovery.....

☣crzy8ers☣ / @crzy8ers

Don't know how else I can thank you for your amazing Hard Drive Data Recover Software... When I print my next photo book, I'm gonna send you a copy... Thank You…

I'm hoping that SpinRite 6.1's ability to once again run on drives of truly any size, with any format file system, and in a reasonable amount of time, will help to dispel the lingering misperception that SpinRite's day has come and gone. Here's fresh Proof that SpinRite is still alive and well. And THAT was done with SpinRite v6.0.

During the testing we've all been doing, many of us are watching SpinRite recovering sectors of data just as well today as it ever has – if not perhaps a bit moreso, since modern drives have pushed the data storage envelope even further.

And I have some surprises up my sleeve for v7.

Operation Triangulation

Three weeks ago, while covering the week's news for episode #926 which was our *"Windows Platform Binary Table"* topic, we touched on Kaspersky's discovery earlier in the week of something unknown, which was apparently generating unexpected network traffic, which they had just found crawling around in their network. And the unknown traffic appeared to be originating from some of their iPhones. At the time I quoted them saying:

> *The malicious toolset does not support persistence, most likely due to the limitations of the OS. The timelines of multiple devices indicate that they may be reinfected after rebooting. The oldest traces of infection that we discovered happened in 2019. As of the time of writing in June 2023, the attack is ongoing, and the most recent version of the devices successfully targeted is iOS 15.7.*

Recall, that they were examining iPhone backups to detect traces of this infection. And they had named this still-unknown malware campaign *"Operation Triangulation."* That being the title of today's podcast, you might expect that we're returning to this because they now know a lot more. And their knowing a lot more coincides with the need all iOS, iPadOS, macOS, and watchOS users had to restart their devices last Wednesday after Apple pushed out a raft of emergency updates in response to what Kaspersky discovered.

So... what DID Kaspersky discover? They used mobile device backups to look at partial snapshots of those device's file systems. And from that they determined this sequence of events:

- *The target iOS device receives a message via the iMessage service, with an attachment containing an exploit.*
- *Without any user interaction, the message triggers a vulnerability that leads to code execution.*
- *The code within the exploit downloads several subsequent stages from the C&C server, that include additional exploits for privilege escalation.*
- *After successful exploitation, a final payload is downloaded from the C&C server, that is a fully-featured APT platform.*
- *The initial message and the exploit in the attachment is deleted*

They explained that at the network level, a successful exploitation attempt can be identified by a sequence of several HTTPS connection events:

- *Legitimate network interaction with the iMessage service, usually using the domain names *.ess.apple.com*
- *Download of the iMessage attachment, using the domain names .icloud-content.com, content.icloud.com*

Multiple connections to the C&C domains, usually 2 different domains (the list of known domains follows). Typical netflow data for the C&C sessions will show network sessions with significant amount of outgoing traffic.

The iMessage attachment is encrypted and downloaded over HTTPS, the only implicit indicator that can be used is the amount of downloaded data that is about 242 Kb.

Using the forensic artifacts, it was possible to identify the set of domain names used by the exploits and further malicious stages. They can be used to check the DNS logs for historical information, and to identify the devices currently running the malware:

Addatamarket[.]net, backuprabbit[.]com, businessvideonews[.]com, cloudsponcer[.]com Datamarketplace[.]net, mobilegamerstats[.]com, snoweeanalytics[.]com, tagclick-cdn[.]com Topographyupdates[.]com, unlimitedteacup[.]com, virtuallaughing[.]com, web-trackers[.]com Growthtransport[.]com, anstv[.]net, ans7tv[.]net

So, essentially, they are unable to see into their iOS devices, and they are unable to see into the communications traffic of their iOS devices. So they're limited to inspecting the metadata traces that are available. They get metadata from examining iPhone backups and from the inevitable DNS lookups which they're able to intercept. As I noted before, this is the double-edged sword of Apple's strong security. It attempts to prevent malware from gaining a foothold in the device. But it just as strongly prevents legitimate researchers from gaining a foothold to understand any malware that does manage to get into a device.

And one of the distressing and growing trends we're witnessing is that these incursions are not arising from some blackhat bad guys wanting to sneak into our devices. The driving forces here appear to be legitimate democracies such as those in France, Germany and the Netherlands. And those are only the ones who have raised their hands to ask whether this could please be made less illegal and unofficially sanctioned. We know that more traditionally repressive regimes are doing the same without asking for anyone's permission. My point is, the more we learn about the increasing pressure to subvert the privacy of our personal communications devices – predominantly coming from the world's governing bodies – the more happy I'm becoming that Apple has been steadfastly working in this direction. There was a time, maybe ten years ago, when the effort Apple was putting into this seemed like a bit of overkill. I no longer think that.

Unfortunately, we're still talking about this today because they haven't yet succeeded in getting it 100% buttoned down. And it's not even clear that it's going to be possible. While we're still using our current hardware architectures and software models, all evidence suggests that new critical bugs are being introduced at about the same pace as old bugs are being found and eliminated. Windows is certainly showing no signs of running out of bugs to patch, and nor, unfortunately, is iOS. While it's true that iOS may have many fewer of them per month, it only ever takes one.

Okay, back to Kaspersky. In their pursuit of this malware over the past three weeks, Kaspersky has posted a series of updates, their most recent one being last Wednesday, coinciding with Apple's release of patches for the 0-day, 0-click problems Kaspersky uncovered.

Over the years, there have been multiple cases when iOS devices were infected with targeted spyware such as Pegasus, Predator, Reign and others. Often, the process of infecting a device involves launching a chain of different exploits, e.g. for escaping the iMessage sandbox while processing a malicious attachment, and for getting root privileges through a vulnerability in the kernel. Due to this granularity, discovering one exploit in the chain often does not result in retrieving the rest of the chain and obtaining the final spyware payload.

For example, in 2021, analysis of iTunes backups helped to discover an attachment containing the FORCEDENTRY exploit. However, during post-exploitation, the malicious code downloaded a payload from a remote server that was not accessible at the time of analysis. Consequently, the analysts lost "the ability to follow the exploit."

In researching Operation Triangulation, we set ourselves the goal to retrieve as many parts of the exploitation chain as possible. It took about half a year to accomplish that goal, and, after the collection of the chain had been completed, we started an in-depth analysis of the discovered stages. As of now, we have finished analyzing the spyware implant and are ready to share the details.

Their comment about this taking them half a year took me by surprise. I had assumed that when said they had caught this malware in their network, they meant a week or two before. But they apparently meant half a year ago and that they have only recently been making the results of this ongoing research public. And now, in retrospect, that does make more sense, since what they are revealing is far more than three week's worth of reverse engineering.

The implant, which we dubbed TriangleDB, is deployed after the attackers obtain root privileges on the target iOS device by exploiting a kernel vulnerability. It is deployed in memory, meaning that all traces of the implant are lost when the device gets rebooted. Therefore, if the victim reboots their device, the attackers have to reinfect it by sending an iMessage with a malicious attachment, thus launching the whole exploitation chain again. In case no reboot occurs, the implant uninstalls itself after 30 days, unless this period is extended by the attackers.

The TriangleDB implant is coded using Objective-C, a programming language that preserves names of members and methods assigned by the developer. In the implant's binary, method names are not obfuscated; however, names of class members are uninformative acronyms, which makes it difficult to guess their meaning:

In other words, as a HUGE aid to anyone wishing to reverse-engineer Objective-C code, the names, and thus the purpose and intentions, of the code routines remain visible, but in this case the names of the variable parameters they are exchanging are not useful. Examples of method names are: populateWithFieldsMacOSOnly, populateWithSysInfo, getCInfoForDump, unmungeHexString and getBuildArchitecture.

Having those names is far more useful than unnamed hexadecimal address offsets which is all that's generally available from any language that compiles all the way down to native machine code after any space-wasting symbols have been removed. Although the variable names that were contained in the exploit code are far less useful, they noted that in many cases it's possible to guess what their acronym names mean from context. For example, osV is the iOS version, and iME contains the device's IMEI. They continue to explain...

Once the implant launches, it starts communicating with the C2 server, using the Protobuf library for exchanging data. The configuration of the implant contains two servers: the primary and the fallback. Normally, the implant uses the primary server, and, in case of an error, it switches to the fallback server by invoking the "swapLpServerType" method.

Additionally, the sent and received messages are encrypted with symmetric (3DES) and asymmetric (RSA) cryptography. All messages are exchanged via the HTTPS protocol in POST requests, with the cookie having the key g and a value that is a digit string from the pubKI configuration parameter.

The implant periodically sends heartbeat beacons that contain system information, including the implant version, device identifiers (IMEI, MEID, serial number, etc.) and the configuration of the update daemon (whether automatic downloads and installations of updates are enabled).

My first thought was that it was interesting that heartbeat data was being periodically sent, since that makes this thing more noisy and thus more prone to discovery. But then I realized that an iPhone is probably already extremely noisy with all of the legitimate traffic it has going in and out. So any heartbeat data is likely able to hide in plain sight without fear of discovery.

The C2 server responds to heartbeat messages with commands. Commands are transferred as Protobuf messages that have type names starting with CRX. The meaning of these names is obscure: for example, the command listing directories is called CRXShowTables, and changing C2 server addresses is handled by the command CRXConfigureDBServer. In total, the implant we analyzed has 24 commands designed for:

- *Interacting with the filesystem (creation, modification, exfiltration and removal of files);*
- *Interacting with processes (listing and terminating them);*
- *Dumping the victim's keychain items, which can be useful for harvesting victim credentials;*
- *Monitoring the victim's geolocation;*
- *Running additional modules, which are Mach-O executables loaded by the implant. These executables are reflectively loaded, with their binaries stored only in memory.*

Their documentation lists each of the individual commands and explains each one's purpose. I won't enumerate them here, but it should be abundantly clear that this represents a full and deep remote takeover of any exploited iPhone. And get a load of this:

One of the interesting commands we discovered is called CRXPollRecords. It monitors changes in folders, looking for modified files that have names matching specified regular expressions. Change monitoring is handled by obtaining a Unix file descriptor of the directory and assigning a vnode event handler to it. Whenever the implant gets notified of a change, the event handler searches for modified files that match the regex provided by the attacker. Such files are then scheduled for uploading to the C2 server.

In other words, it's possible for the command and control server to prime a device to

autonomously notify the server when something happens of specific interest. When a change in the contents of a directory occurs, a check is done for relevancy and if that comes back affirmative, the files in question are queued for transmission. In a very real sense, it's no longer your iPhone in your pocket, it's theirs. Talk about being Pwned!

> *While analyzing TriangleDB, we found that the class CRConfig (used to store the implant's configuration) has a method named populateWithFieldsMacOSOnly. This method is not called anywhere in the iOS implant; however, its existence means that macOS devices can also be targeted with a similar implant;*
>
> *The implant requests multiple entitlements (permissions) from the operating system. Some of them are not used in the code, such as access to camera, microphone and address book, or interaction with devices via Bluetooth. Thus, functionalities granted by these entitlements may be implemented in modules.*

At the end of the work of assembling all of this I found an earlier note written by Eugene Kaspersky himself at the beginning of the month. He wrote:

> *We believe that the main reason for this incident is the proprietary nature of iOS. This operating system is a "black box", in which spyware like Triangulation can hide for years. Detecting and analyzing such threats is made all the more difficult by Apple's monopoly of research tools – making it a perfect haven for spyware. In other words, as I've often said, users are given the illusion of security associated with the complete opacity of the system. What actually happens in iOS is unknown to cybersecurity experts, and the absence of news about attacks in no way indicates their being impossible – as we've just seen.*

I thought that was very interesting. He's clearly quite annoyed by their inability to obtain any visibility into what's going on inside an iPhone. They are limited to monitoring encrypted traffic for metadata and making iPhone backups and sifting through that detritus for clues. I can understand his frustration when they are some of the targets of these attacks. And what he just said echoes that thought that occurred to me a few weeks ago when I realized that Apple's security has the unintended effect of protecting malware from discovery.

So, this is everything that Kaspersky has publicly shared so far. And the glaring piece of information that's lacking is any commentary about how this thing crawls into iPhones by escaping Apple's security controls. We have one clue thanks to the CVE associated with one of Apple's updates last week:

> *CVE-2023-32434:* ***Integer overflow in kernel.*** *An app may be able to execute arbitrary code with kernel privileges. Apple is aware of a report that this issue may have been actively exploited against versions of iOS released before iOS 15.7. [Credit given to Georgy Kucherin (@kucher1n), Leonid Bezvershenko (@bzvr_), and Boris Larin (@oct0xor) of Kaspersky.]*

That iOS 15.7 exactly matches what Kaspersky said about this attack on them, and the three Russians credited with this discovery and report are employed by Kaspersky. So it appears that

Kaspersky knows more than they are saying for the time being. Given that this vulnerability apparently enables a powerful 0-click iPhone takeover, we may never learn more. And that would be just as well.

One last piece of information that came from Eugene Kaspersky was an explanation for their choice of the name "Triangulation" – which I'd been wondering about. He wrote:

> ***P.S. Why the name "Triangulation"?***
>
> *To recognize the software and hardware specifications of the attacked system, Triangulation uses Canvas Fingerprinting technology, drawing a yellow triangle in the device's memory.*

What he means there, is that it's possible and often useful, to ask graphic rendering software to draw into an off-screen buffer. And the precise details of one graphic renderer compared to another may differ ever so slightly. The difference might be invisible to the naked eye. But, for example, when a diagonal line is drawn, as when rendering a triangle, the exact values chosen by the line smoothing anti-aliasing algorithm might differ from one generation or model of device to another. The practice known as "Canvas Fingerprinting" uses those invisible yet significant details.

So... thanks to Kaspersky's intrepid work, with their forensic analysis being actively impeded every step of the way by the very security they were trying to strengthen, last Wednesday's Apple updates foreclosed upon a kernel vulnerability that had apparently been in active use for at least four years. We'll never know who or why or what or where... but at least now, we do know how.

Do the bad guys have another way in? Unfortunately, that seems more than likely. What's most annoying and galling, though, is the idea that our own governments may be the customers for whatever comes next.