Security Now! #928 - 06-20-23
The Massive MOVEit Maelstrom

This week on Security Now!

This week, two big stories dominate our podcast. We start by taking a quick look back at last week's Microsoft Patch Tuesday. Then we examine the latest surprising research to emerge from the Ben-Gurion University of the Negev. What these guys have found this time is startling. Then, after sharing some feedback from our listeners and a long-awaited big SpinRite milestone announcement, we're going to spend the rest of our available time examining the story behind this month's massive cyber-extortion attack which is making all of the recent headlines and causing our listeners to tweet: "I'll bet I can guess what you're going to be talking about this week." Yes, indeed.

If it's not tied down…

Security News

Patch Tuesday

This being the 3rd Tuesday of the month, we're able to look back on last week's Patch Tuesday. It's still somewhat astonishing that just last week Microsoft patched 26 Remote Code Execution vulnerabilities, four being critical, with three of those four spotted and fixed in a single Windows component known as PGM, the Pragmatic General Multicast queue. Because, you know, the thing you really want from your general multicast queues is some pragmatism. What you don't want from them is remote code execution flaws... especially when they're bearing a CVSS score of 9.8. So, after applying last Tuesday's patches, for the time being there are three fewer RCE's... which probably makes the decision to use the queuing API at least somewhat more pragmatic.

There were also 17 Elevation-of-Privilege problems fixed, only one of which made it to the critical list with a CVSS score of a whopping 9.8. If exploited, an attacker could gain administrative privileges. Microsoft wrote: "An attacker who has gained access to spoofed JSON Web Token authentication tokens could use them to execute a network attack which bypasses authentication and allows them to gain access to the privileges of an authenticated user. The attacker needs no privileges nor does the user need to perform any action." Since we're talking about SharePoint, which is the same sort of database sharing that has landed the MOVEit Transfer software in such hot water this month, getting that one patched is something that should probably not be put off... especially given its critical rating.

Overall, of the 73 flaws fixed, 6 are rated Critical, 63 are Important, 2 are rated Moderate, and one is rated Low in severity. Included within that group are 3 issues fixed in the Edge browser. It's worth noting that during the intervening month between last month's patches and this month's, Microsoft also eliminated 26 other flaws in Edge. Included among those was a 0-day bug that Google last week disclosed as being actively exploited in the wild.

But this month is the first time in several months when none of the known and patched problems were either publicly known or under active attack at the time of their fixes. So that's good.

The final note is that two of the remaining remote code execution vulnerabilities were found and fixed in Exchange Server. Being that Exchange Server is all about exchanging content with the outside world, it's generally a significant portion of an enterprise's attack surface.

Does EVERYTHING leak??

This next bit of jarring news leads me to pose the rhetorical question: *"Does everything leak?"* As in leaking information? A couple of days ago, I received a Twitter DM from Ben Nassi:

Ben Nassi / @ben_nassi

> *Hi Steve, my name is Ben Nassi, a postdoctoral researcher at Cornell Tech and a long time listener of Security Now. I just published a new research that I think you should see. We recovered cryptographic keys from devices by obtaining video footage of their power LED. The devices were not compromised. The research will be presented at BlackHat and DEFCON this year.* https://eprint.iacr.org/2023/923.pdf

The summer is approaching with BlackHat and DEFCON being held back-to-back every summer in Las Vegas, Nevada. Ben also sent a link to the research paper which is titled: *“Video-Based Cryptanalysis: Extracting Cryptographic Keys from Video Footage of a Device’s Power LED”*.

Ben is one of the prolific researchers with the Ben-Gurion University of the Negev, who have for years brought us often entertaining, but also often-sobering, extremely clever examples of data exfiltration. Our longtime listeners will recall, for example, their work with extracting audible conversation from a room at a distance by visually detecting the sympathetic vibrations induced in a birthday party balloon, the leaf of a plant, or a lightbulb, all of which allowed them to recover the audio filling the rooms occupied by those objects.

I think that we could safely label these guys the masters of the detection and recovery of side-channel information leakage. And so we now have another. And whereas some of their schemes have required malware to first be installed in the victim device -- like to deliberately change the sound being emitted by a machine's power supply or changing the system's fan speed in order to signal an air-gapped microphone, this most recent attack, as Ben indicated in his Tweet, requires no prior compromise.

In order to put myself in the same place as our listeners as I share this, I have not yet even glanced at Ben's paper. We'll do that together in a moment when I share the paper's Abstract which I have not yet seen. But what we immediately and intriguingly ascertain from Ben's Tweet is that, astonishingly, variations in the work being done inside devices, where that work is dependent upon cryptographic secrets, must be sufficient to produce tiny variations in the power being supplied to such a device's LED power indicator. That's surprising enough. But even if we were to stipulate that this was true, then one would hope – and actually expect – that any such fluctuations would be so tiny as to be undetectable. Or that if they were theoretically detectable that they could not be detected at a distance by a standard video system due to the depth of digitizing resolution, the effects of video compression, or the frame rate of the video. But to do what they've done, Ben's group apparently overcame all of these practical barriers. And as I've observed in the past with their work, what really distinguishes their accomplishments is that they wrestle these things all the way to the ground.

So now let's discover together what Ben's group explains at the start of their paper. They write:

> *In this paper, we present video-based cryptanalysis, a new method used to recover secret keys from a device by analyzing video footage of a device’s power LED. We show that cryptographic computations performed by the CPU change the power consumption of the device which affects the brightness of the device’s power LED. Based on this observation, we show how attackers can exploit commercial video cameras (e.g., an iPhone 13’s camera or Internet-connected security camera) to recover secret keys from devices. This is done by obtaining video footage of a device’s power LED (in which the frame is filled with the power LED) and exploiting the video camera’s rolling shutter to increase the sampling rate by three orders of magnitude from the FPS rate (60 measurements per second) to the rolling shutter speed (60K measurements per second in the iPhone 13 Pro Max).*

Okay. That's brilliant. With an LED, even if the illumination across the surface of the LED is not perfectly uniform, as in fact it won't be, being a solid state illuminator, any CHANGE in illumination **will** be uniform and effectively instantaneous. So this is the key that allows them to obtain a sufficiently high **EFFECTIVE** sampling rate from an otherwise grossly insufficient 60 frames per second video recording. Continuing...

> *The frames of the video footage of the device's power LED are analyzed in the RGB space, and the associated RGB values are used to recover the secret key by inducing* [I'm sure they meant deducing] *the power consumption of the device from the RGB values.*

Pausing for a moment, again... we long ago talked about variations in power consumption during cryptographic operations being a well understood side-channel that could theoretically be used to reverse engineer the work being done by a device when that work is a function of secret data. But the presumption has been that theory runs smack up against reality when there's no way to practically obtain instantaneous power consumption measures without hooking deeply into a target device's electronics. These guys have quite cleverly solved the problem of doing that. They realized that minute variations in the device's power draw would induce tiny changes in the system power supply voltage. The instantaneous brightness of an LED is determined by the instantaneous current flowing through it. LEDs will invariably have a resistor in series with them to set their operating current. But that current is not otherwise regulated. And that means that any variation in the system's total supply voltage will create a variation in the LED's current and therefore in its illumination. I'm still surprised that this works, but apparently it does, and you have to imagine that the world's intelligence services just perked up in response to this news.

> *We demonstrate the application of video-based cryptanalysis by performing two side-channel crypt-analytic timing attacks and recover: (1) a 256-bit ECDSA key from a smart card by analyzing video footage of the power LED of a smart card reader via a hijacked Internet-connected security camera located 16 meters away from the smart card reader, and (2) a 378-bit SIKE key from a Samsung Galaxy S8 by analyzing video footage of the power LED of Logitech Z120 USB speakers that were connected to the same USB hub (that was used to charge the Galaxy S8) via an iPhone 13 Pro Max. Finally, we discuss countermeasures, limitations, and the future of video-based cryptanalysis in light of the expected improvements in video cameras' specifications.*

Before we go any further I'll just mention that the Light Pen I developed back in 1983 for the Apple II had a response time of 140 nanoseconds because that was the pixel clock rate of the Apple II's video. 140 nanoseconds is a rate of 7.14 megahertz. That happens to be twice the NTSC color burst frequency, which was part of Woz's design brilliance for the Apple II.

My point is, since monitoring the power LED of devices that are performing secret computations has now been proven to work, it would be trivial to take the technology of a high performance light pen, place its photodiode at the user-end of a telescope, and aim that scope at any power LED of any device containing secrets to begin collecting data. The data gathering and secret gathering power of such a system with seven megahertz bandwidth would be somewhat terrifying.

While it would be possible to create a highly sophisticated spying scope, the brilliance of their discovery and invention is the ability to use existing camera technologies thanks to their observation that the cameras in our devices do not actually snap an entire scene at once. Instead, they actually scan the image from top or bottom or left to right in much the way the images of our original cathode ray tubes did. This brilliantly allows them to sample the illumination of a device's LED with far greater temporal resolution than the camera's overall frame rate. Here's how Ben's paper describes it under the title "Increasing a Video Camera's Sampling Rate Using a Rolling Shutter"...

> *We note that the FPS rate supported by the vast majority of commercial smartphones and security/IP video cameras is limited to 60-120 FPS which is insufficient for performing cryptanalysis. In order to increase the number of measurements per second (sampling rate) to a level sufficient for cryptanalysis, the attacker can exploit the video camera's rolling shutter. The rolling shutter is an image-capturing method in which a frame of a video (in video footage) is captured by scanning the scene vertically/horizontally. When this method is used, a frame/picture is not actually composed of a single snapshot of a scene taken at a specific point in time but rather is composed of multiple snapshots taken of vertical/horizontal pieces of the scene at different times. With a vertical rolling shutter, a sensor's pixels are exposed and read out row-by-row sequentially at different times from top to bottom (or left to right) according to a configurable shutter speed which determines the amount of time that the sensor is exposed to light. Because each row (or a group of adjacent rows) in a sensor with a rolling shutter is captured at a different time, attackers can increase the sampling rate from the camera's FPS rate (60/120 FPS) to the rate at which rows are recorded, a rate which is based on the shutter speed.*

The biggest limitation is that for this clever rolling shutter rate up-sampling to work, the LED's image must fill the entire camera frame. But there are tiny external lenses that can be added to a smartphone to make that easier, and there are doubtless many applications where the installation of a device, such as a smart card reader, is made with the assumption that its internal secret cryptographic computations are not being broadcast outside of the device. Ben's latest work has, amazingly enough, shown the world that this long standing assumption has always been wrong.

We've often commented about the fact that the Ethernet "activity LEDs" which are ubiquitous on all networking equipment do not reveal **anything** about the data that's passing through their interfaces.

We've been looking at the wrong LED.

Closing the Loop

Kevin / @sharpestmarble

> *Listening to Security Now 927 and you're praising Apple for Live Voicemail, which they're going to be developing. But this is something Google phones have had for over a year now.*

vincent stacey / @vincent_stacey

> *Think Apple could use the same on iPhone technology protecting children from unwanted images to catch and analyze the iMessage vulnerability?*

I suspect not, because one aspect – scanning for unwanted material – is within the bounds of normal operation while the operation of this malware is explicitly "out of bounds." The trouble with catching it is that whatever it is that the iMessage vulnerability is doing, it's breaking supposedly unbreakable rules. It's somehow escaping from the rigid controls and sandboxing that Apple has explicitly and deliberately erected to contain and neuter exactly such exploits. We know only that the malicious iMessage is bringing along an attachment. The fact that this is a zero-click exploit means that for the benefit of its user, iOS must be automatically attempting to render whatever it is that iMessage has brought along with it... and that it is during this auto-rendering that iOS loses control.

That said, if iOS were somehow able to record all iMessage attachments – before performing any processing of the attachment – into an immutable audit log on the target's device, then if a target's device were compromised, and if that compromise was recognized, then it would seem possible for a forensic analysis to be made. This theoretical immutable log would need to be immutable even to Apple's own software, once written, otherwise malware could arrange to erase the evidence, just as traditional OS attackers erase the logs of their own OS penetration.

Jason Egan / @beguil3d

> *Hey Steve! After hearing about the grc.sc situation (and please forgive me if I'm late to giving this option) I wondered if you couldn't just solve the problem with a RewriteRule on your host? I've had to do this in the past with some of my projects.*

I could solve the problem with a rewrite rule IF my DNS included *.grc.sc in addition to grc.sc. But it never occurred to me that anyone would stick an arbitrary "www." in front of "grc.sc."

As I've mentioned before, I **DO** have a *.grc.com in my DNS since that's able to handle all of the many prefixes that we use such as "www.", "forums.", "dev.", "sqrl." and whatever else. And also, I **DO** redirect any other wayward querie over to www.grc.com. Many many years ago, GRC's web servers accepted connections on either "grc.com" or "www.grc.com". I figured, why not? But then we noticed that Google's search results were coming up with a mixture of either grc.com or www.grc.com. Google was seeing these as separate distinct websites. And people were linking to GRC somewhat arbitrarily as either grc.com or www.grc.com. And that was having the unintended side effect of reducing our Google page ranking by diluting the number of incoming links among both sites. So I changed GRC's web server to return an HTTP/301 redirect from any plain grc.com over to www.grc.com. And Google's spiders quickly learned that this was a single site and then consolidated all links around that single domain.

At the time I had to choose one or the other, and I'm not sure I did the right thing by choosing the longer "www.grc.com" as the enforced default. Yes, it's technically the more accurate of the two, but whenever I talk about the domain I just say "grc.com" not "www.grc.com", and I'm sure everyone who manually enters grc.com into their browser's URL field leaves off the "www."

And, by the way, the mystery of how this "www.grc.SC" came about was answered:

CPUGuru / @cpuguru

> *I found the source of the errant "www" that you discussed in the podcast - the hyperlink in the 926 PDF actually includes it!*

Brilliant observation, CPUGuru. Nice going!

José Javier Vegas / @VegasJoseJavier

> *Hi Steve, regarding SN927, there is no official Let's Encrypt client today. They transferred their implementation to the EFF and it's now called "Certbot", which is the one they recommended.*

That's good to know. I'm still issuing certs for my servers the old fashioned way through my favorite CA, DigiCert. But, assuming that the world is going to switch to 90-day maximum life certificates, I'm glad that DigiCert also supports the ACME protocol and I will certainly, then, need to find automation for this, as will everyone.

Defensive Computing - Michael Horowitz / @defensivecomput

> *Steve: an FYI about HP plus printers - they must be online all the time. even if connected to a PC via usb. And, you must have an HP account. Perfect for spying. Details on this page*
>
> *"A Defensive Computing Checklist" by Michael Horowitz*
> https://defensivecomputingchecklist.com/printers.php

The first line of Michael's page on printers reads: *"I hate printers. So too, does Leo Laporte, who is known as the Tech Guy on the radio. He will not take phone calls about printers."*

I read through some of what Michael wrote on that page and I recommend it to any of our listeners who may be curious to know more, and who might be in the market for a printer. Michael has collected many reviews and anecdotes... and he tells a horror story of spending half a day trying to install an HP printer/scanner for a friend. Mind you, Michael knows his way around PCs as well as any of us, yet he repeatedly hit wall after wall. His experience further supported my earlier statement that I have long found Hp's software to be unconscionably atrocious.

SpinRite

It is with no small amount of pride, a feeling of accomplishment and some pent up relief, that I can finally assert that, as far as I know, the work on the business end of SpinRite 6.1 – its DOS executable – is finished. There are presently no remaining known bugs, great or small. Last Sunday afternoon I announced the availability of the 29th alpha release to GRC's 696 registered SpinRite 6 owners who have been testing 6.1, and I suggested for the first time that the code we all now have likely would and certainly could, with few changes, be moved into beta status to soon become SpinRite 6.1's shipping code.

6.1 contains a great many new features, one of them being an integrated FAQ which explains the choice of SpinRite's five redefined operating levels, its many command-line options, and other useful tidbits. So, while the dust settles on this latest alpha, I'm currently writing that FAQ. That'll give some time for any testers who may have become bored with the seemingly endless interim development releases to give this proposed-final code one last check.

Once we have the finished DOS code, that code, along with the very first work I did to create GRC's InitDisk USB drive prep utility, will be integrated into an updated SpinRite Windows app and then we'll have v6.1.

Any new purchasers of SpinRite will automatically be receiving 6.1. But, since I want to let this new code breathe a bit, I plan to hold off on announcing it to all of SpinRite's past purchasers until it's had a bit more time among a wider audience. So, as soon as the official upgrade path is established, I'll be inviting all of this podcast's listeners, many of whom have previously purchased SpinRite, to update to v6.1. Then, once it appears that it's going to be clear sailing, I'll begin the process of informing everyone who owns 6.0.

So, no new action needs to be taken by anyone who hasn't already jumped onto testing the pre-release code. I just wanted to note that the project had achieved a significant milestone along the way.

The Massive MOVEit Maelstrom

Our main topic today arrives about three weeks after the first signs of this significant problem arose. I've been aware of and watching what's been happening. But it wasn't until the past week that the scope and scale of the problem became fully apparent. So this week we need to do a bit of catching up with what's been going on.

The trouble surrounds a globally popular file transfer facility named MOVEit. MOVEit is a local and cloud-based file sharing and management solution from a company named Progress.

https://www.progress.com/moveit

MOVEit describes itself as "Managed File Transfer Software"

- *Secure File Transfer and Automation Software for the Enterprise*
- *Guarantee the reliability of core business processes and transfer sensitive data between partners, customers and systems the secure and compliant way with MOVEit.*
- *Secure, Auditable, Automated, and Compliant File Transfer - On-Premise and In the Cloud*
- *MOVEit provides secure collaboration and automated file transfers of sensitive data and advanced workflow automation capabilities without the need for scripting. Encryption and activity tracking enable compliance with regulations such as PCI, HIPAA and GDPR*

We've just read that it's compliant with PCI, HIPPA and GDPR. Unfortunately, another abbreviation its web front-end is fully compliant with S.Q.L., and not in the way they intended. That's right, the industry has been hit with another very powerful and significant SQL Injection attack. I was tempted to title today's podcast "Little Bobby Drop Tables."

The last time we were on this subject, I railed against the fundamentally broken design of the SQL command model, which exposed a fully capable command language to a web server that typically only needed to issue queries against that data. Yet the unrestrained nature of this powerful command interface meant that the web server could do anything it wished. And if someone could arrange to get the web server to pipe their own user-supplied text through to the back-end SQL server, such a remote user could do pretty much anything they wished. My denigration of SQL generated some pushback from some of our listeners who quite correctly noted that there were several other much safer and much more proper ways to do this with modern SQL servers. And those listeners were 100% correct. But I wasn't saying that safer and more proper ways had not since been developed to do this, but that the original unsafe ways also continued to be present in the interest of backward compatibility and not breaking legacy systems. Consequently, nothing prevents the original horrible and fundamentally insecure approach from continuing to be used. And so, today, we have the latest example of this bad architecture striking once again.

Even if someone was being as responsible as they could be with this hot potato, it would be the programmer's responsibility to try to think of all possible ways bad guys might attempt to sneak commands through under the cover of data by encoding them strangely, using an unusual language locale, or who knows what?

As is always the case with security, the battle is asymmetric: the programmer must block every possible avenue of conquest whereas the bad guys only need to find one way in.

To Progress's credit, from the start they have not tried to hide this in any way. Right there on their main product promo page they write: "PRODUCT ADVISORY: MOVEit Transfer and MOVEit Cloud Vulnerability, click for mitigation measures and patch information."

So here's what we knew three weeks ago as May was coming to an end:

As we already know, MOVEit's solution includes a web-based front to manage the sharing, uploading and downloading of files. This makes sense in an era of JavaScript which is able to accept drag and drop uploads and managed local downloads. When we mix in web authentication, which is, as we know, an entire discipline of its own, it's quite possible to create a fully functional web-based file management and distribution system. And when the alternative was eMail, there really was no competition. MOVEit also supports FTP, but I assume that's these for legacy purposes since FTP support has finally been deprecated or removed from web browsers and it's unclear what advantage FTP has over any modern web browser soltuion. Once upon a time, sure. Today? No.

As for this vulnerability, the bad news is that it was a true 0-day. Progress learned of it after and because the bad guys were already exploiting it. Being responsible, Progress quickly patched all of their supported MOVEit versions and their cloud-based service.

There are essentially four things that can be done with this vulnerability: The deletion of existing data, the exfiltration of existing data, the modification of existing data and the implantation of new files and malware. As it has turned out, two of those four have been seen to be happening.

Mandiant, now owned by Google Cloud, has been tracking the MOVEit breach activity under the uncategorized moniker UNC4857 and posted that the opportunistic attacks have singled out a wide range of industries based in Canada, India, the U.S., Italy, Pakistan, and Germany. And Mandiant also wrote that it was "aware of multiple cases where large volumes of files have been stolen from victims' MOVEit transfer systems" and adding that the web shell left behind, which they call LEMURLOOT, is also capable of stealing Azure Storage Blob information.

So we have massive data exfiltration. And Mandiant's mention of a web shell brings us to the second of the two things that is being done: the implantation of new files and malware. Because the bad guys have also been found to be dropping web shell malware before they leave.

We've previously talked about web shells, but here's a bit of history: In the early days of the web, web servers only delivered static HTML web pages. You gave them a URL which was the location of the page's text on the server and it returned that page. After several years of that, browser-side scripting which was embodied by and enabled by JavaScript, introduced the concept of client-side scripting. This brought the user's client alive, giving it some of the capabilities of a local application. A perfect and simple example is that a user's entered password could be hashed locally by script in the browser so that a web server never received anything but the hash. These days we're seeing the logical evolution of scripting on the client with amazingly complete web apps. But even browser-side scripted pages could be delivered by static files, where the file stored on the server's drive was simply sent to anyone requesting it.

The big change on the server side occurred when web servers started running code to respond to queries being made by their web clients. The earliest implementation of this was known as CGI which stands for “Common Gateway Interface”. The idea behind CGI was simplicity: a web client would make a query, and the web server would essentially serve as an intermediary between the user's web browser and some code on the server. To do this, the web server would launch and run a separate CGI program in the background. The web server would provide the CGI program with what the user had queried, and whatever the CGI program returned through its Standard Output would be piped back by the web server to the user's web browser. So with CGI, rather than delivering static textual web pages, the output of a pre-compiled program was returned to the user's web browser. This was a big application for Larry Wall's PERL which was often used in early CGI applications.

This model was clean and simple and it remains in heavy use today where PHP is the back-end recipient of a client's CGI queries. GRC's web forums and link shortener are all PHP, as is WordPress which, as we know, runs a huge portion of the web. Modern web servers provide for many ways to generate dynamic content. GRC's ShieldsUP!, Perfect Passwords and Perfect Paper Passwords, the DNS Spoofability test and GRC's support for SQRL are all implemented using DLL's that I wrote in assembly language using the Microsoft ISAPI API to obtain and return data to and from user's web pages.

Microsoft also promotes their own server-side interpreter which implements their scripting language Active Server Pages (ASP). When a web server which has Active Server Pages enabled, encounters a URL referring to a file ending in .asp or .aspx, the web server will look for that file on the URL's provided patch and will run the code that it contains.

Which brings us back to the LEMURLOOT web shell that's left behind by these attackers. In machines that have been attacked, the attackers leave behind a file named “human2.aspx.” They chose that name since a “human.aspx” file is already present in the system as part of the authentic file set. They also sometimes leave additional files with the file extension “.cmdline”. This “human2.aspx” file functions as a web shell. It’s a sophisticated script that will provide future access to any ASP-script capable web server that’s unlucky enough to host it. The bad guys know its name and its location on the server, so they’re able to invoke it remotely at any time in the future simply by querying the server for a URL and path ending in “human2.aspx”. The web server will immediately run that code which typically gives remote attackers control of the system and its network. The web shell is also engineered to add new admin user account sessions with the name “Health Check Service” in an effort to appear benign and expected.

This means that just patching against the attack after the fact will **not** be sufficient protection, since a previously vulnerable server may have already been quietly infected with the human2 web shell. After patching it will be necessary to search for any “IOC's” as we call them today: “Indications of Compromise.”

CISA quickly issued a nationwide alert to let everyone know and to demand that all government agencies using the MOVEit Transfer system update and check for evidence of past incursions.

Our friends at the web scanning search engine CENSYS, who were the subject of last week's podcast, have identified more than 3,000 vulnerable instances of MOVEit, the majority of which are located in the U.S.

Huntress Labs was all over this at the start of the month. Excerpting from and editing what they wrote:

> *On June 1, 2023, Huntress was made aware of active exploitation attempts against the MOVEit Transfer software application. Previously, on May 31, 2023, the vendor Progress had just released a security advisory expressing there is a critical vulnerability that could lead to unauthorized access.*
>
> *On June 2, the industry dubbed this vulnerability as CVE-2023-34362.*
> *Progress brought down MOVEit Cloud as part of their response and investigation.*
>
> *Huntress has fully recreated the attack chain exploiting MOVEit Transfer software. We have uncovered that the initial phase of the attack, SQL injection, opens the door for even further compromise -- specifically, arbitrary code execution. We use our exploit to receive shell access with Meterpreter, escalate to NT AUTHORITY\SYSTEM and detonate a cl0p ransomware payload. (We'll be talking more about Cl0p in a moment.)*
>
> *This means that any unauthenticated adversary could trigger an exploit that instantly deploys ransomware or performs any other malicious action. Malicious code would run under the MOVEit service account user moveitsvc, which is in the local administrators group. The attacker could disable antivirus protections, or achieve any other arbitrary code execution.*
>
> *Another demonstration showcased compromising the MOVEit Transfer API and application itself. With that alone, we upload, download, and potentially exfiltrate files as a threat actor would.*
>
> *The behavior that the industry observed, adding a human2.aspx webshell, is not necessary for attackers to compromise the MOVEit Transfer software. It's "an option" that this specific threat chose to deploy for persistence, but the attack vector offers the ability to detonate ransomware right away. Some have already publicly reported attackers pivoting to other file names.*
>
> *The recommended guidance is still to patch and enable logging. From our own testing, the patch does effectively thwart our recreated exploit.*

Microsoft attributed the MOVEit Transfer zero-day attacks to Lace Tempest, a threat actor previously linked to Cl0p ransomware, data theft, and extortion attacks.

On June 6, the Cl0p gang posted a communication to their leak site demanding that victims contact them before June 14 to negotiate extortion fees for deleting stolen data:

Dear Companies.

CLOP is one of top organization offer penetration testing service after the fact.

This is an announcement to educate companies who use Progress MOVEit product that chance is that we download a lot of your data as part of exceptional exploit. We are the only one who perform such attack and relax because your data is safe.

We are to proceed as follow and you should pay attention to avoid extraordinary measures to impact you company.

IMPORTANT! We do not wish to speak to media or researcher. Leave.

STEP 1 - If you had MOVEit software continue to STEP 2, else leave.
STEP 2 - Email our team unlock@rsv-box.com or unlock@support-multi.com.
STEP 3 - Our team will eMail you with dedicated chat URL over TOR.

We have information on hundreds of companies so our discussion will work very simple.

STEP 1 - If we do not hear from you until June 14th we will post your name on this page.
STEP 2 - If you receive chat URL go there and introduce you.
STEP 3 - Our test will provide 10% proof of data we have and price to delete.
STEP 4 - You may ask for 2-3 files random as proof we are not lying.
STEP 5 - You have 3 day to discuss price and if no agreement you custom page will be created.
STEP 6 - After 7 days all your data will start to be publication.
STEP 7 - You chat will close after 10 not productive day and data will be publish.

WHAT WARRANTY? Our team has been around for many years. We have not even one time not do as we promise. When we say data is delete it is cause we show video proof. We have no use for few measle dollars to deceive you.

Call today before your company name is publish here.

FRIENDLY CLOP. PS. If you are a government, city or police service do not worry. We erased all your data. You do not need to contact us. We have no interest to expose such information.

And then, presumably after the initial June 14th contact deadline passed, we have "Updates"

BISSELL.COM 50TB Company data get ready for something interesting.
EMERALDC.COM 100TB Company data get ready for something interesting.

And a bit later...

360EquipmentFinance.COM Files Part 1 PUBLISHED
PrecisionMedicalBilling.NET Files Part 1 PUBLISHED
HCI.EDU Files Part 1 PUBLISHED

So by this point in the story of the past three weeks, we have the internal private data of thousands of U.S. companies who have been using Progress's MOVEit Transfer software, exfiltrated from their servers and in some cases as much as 100TB worth.

This was not good, but we're not done yet. After a week had passed, believe it or not, Progress announced the news of additional discovered vulnerabilities:

> *June 9, 2023, In addition to the ongoing investigation into vulnerability (CVE-2023-34362), we have partnered with third-party cybersecurity experts to conduct further detailed code reviews as an added layer of protection for our customers. As part of these code reviews, cybersecurity firm Huntress has helped us to uncover additional vulnerabilities that could potentially be used by a bad actor to stage an exploit. These newly discovered vulnerabilities are distinct from the previously reported vulnerability shared on May 31, 2023.*
>
> *All MOVEit Transfer customers must apply the new patch, released on June 9. 2023. All MOVEIt Cloud customers, please see the MOVEit Cloud Knowledge Base Article for more information.*
>
> *The investigation is ongoing, but currently, we have not seen indications that these newly discovered vulnerabilities have been exploited.*

Then, exactly one week after that, last Friday, June 16th there's more...

> *June 16, 2023, Yesterday we reported the public posting of a new SQLi vulnerability that required us to take down HTTPs traffic for MOVEit Cloud and to ask MOVEit Transfer customers to take down their HTTP and HTTPs traffic to safeguard their environments. We have now tested and deployed a patch to MOVEit Cloud, returning it to full service across all cloud clusters. We have also shared this patch and the necessary deployment steps with all MOVEit Transfer customers.*
>
> *All MOVEit Transfer customers must apply the new patch, released on June 16. 2023. Details on steps to take can be found in the following Knowledge Base Article. All MOVEIt Cloud customers, please see the MOVEit Cloud Status Page for more information.*
>
> *The investigation is ongoing, but currently, we have not seen indications that this newly discovered vulnerability has been exploited.*

Okay. So what do we know about the victims so far? CNN Business had some reporting on this. Excerpting from what CNN reported:

A growing number of businesses, universities and government agencies have been targeted in a global cyberattack by Russian cybercriminals and are now working to understand how much data was compromised.

CISA said Thursday that "several federal agencies… have experienced intrusions." The U.S Department of Energy said it "took immediate steps" to mitigate the impact of the hack after learning that records from two department "entities" had been compromised. It's also impacted state governments in Minnesota and Illinois. And on Thursday, state agencies said 3.5 million Oregonians with driver's licenses or state ID cards had been impacted by a breach as well as anyone with that documentation in Louisiana. British Airways confirmed that its staffers' names, address, national insurance numbers and banking details were exposed because its payroll provider Zellis used MOVEIt. The BBC said its staff had also been afflicted because Zellis was its payroll provider. The UK's beauty and health company Boots said some of its team members' information was also stolen.

Brett Callow, threat analyst at cybersecurity firm Emsisoft, said the hackers have also listed Aon and The Boston Globe as victims. "By my count, there are now 63 known/confirmed victims plus an unspecified number of USG agencies." The hacking campaign has also spread to academia. Johns Hopkins University in Baltimore and the university's renowned health system said in a statement that "sensitive personal and financial information," including names, contact information, and health billing records may have been stolen in the hack. Meanwhile, Georgia's state-wide university system – which spans the 40,000-student University of Georgia along with over a dozen other state colleges and universities – confirmed it was investigating the "scope and severity" of the hack.

TechCrunch added:

> *Clop, the ransomware gang responsible for exploiting a critical security vulnerability in a popular corporate file transfer tool, has begun listing victims of the mass-hacks, including a number of U.S. banks and universities.*
>
> *The victim list, which was posted to Clop's dark web leak site, includes U.S.-based financial services organizations 1st Source and First National Bankers Bank; Boston-based investment management firm Putnam Investments; the Netherlands-based Landal Greenparks; and the U.K.-based energy giant Shell. GreenShield Canada, a non-profit benefits carrier that provides health and dental benefits, was listed on the leak site but has since been removed.*
>
> *Other victims listed include financial software provider Datasite; educational non-profit National Student Clearinghouse; student health insurance provider United Healthcare Student Resources; American manufacturer Leggett & Platt; Swiss insurance company ÖKK; and the University System of Georgia (USG).*
>
> *A spokesperson for German mechanical engineering company Heidelberg, which Clop listed as a victim, told TechCrunch in a statement that the company is "well aware of its mentioning on the Tor website of Clop and the incident connected to a supplier software."*
>
> *Clop, which like other ransomware gangs typically contacts its victims to demand a ransom payment to decrypt or delete their stolen files, took the unusual step of not contacting the organizations it had hacked. Instead, a blackmail message posted on its dark web leak site told victims to contact the gang prior to its June 14 deadline.*

> *Multiple organizations have previously disclosed they were compromised as a result of the attacks, including the BBC, Aer Lingus and British Airways. These organizations were all affected because they rely on HR and payroll software supplier Zellis, which confirmed that its MOVEit system was compromised.*
>
> *The Government of Nova Scotia, which uses MOVEit to share files across departments, also confirmed it was affected, and said in a statement that some citizens' personal information may have been compromised. However, as we know, Clop's leak site said "if you are a government, city or police service… we erased all your data."*
>
> *Ofcom, the U.K.'s communications regulator, also said that some confidential information had been compromised in the MOVEit mass-hack. In a statement, the regulator confirmed that hackers accessed some data about the companies it regulates, along with the personal information of 412 Ofcom employees.*
>
> *Transport for London (TfL), the government body responsible for running London's transport services, and global consultancy firm Ernst and Young, are also impacted, according to BBC News. Neither organization responded to TechCrunch's questions.*

Since we haven't yet enumerated the literal **thousands** of individual companies, government and educational agencies and other organizations that that have been compromised in this mass attack, many more victims are expected to be revealed in the coming days and weeks.

And now that they knew what to look for, security researchers looking back through their logs have determined that someone had been experimenting with the exploitation of these MOVEit vulnerabilities for the past two years. And Clop was also responsible for previous mass-attacks exploiting flaws in Fortra's GoAnywhere file transfer tool (remember that?) and Accellion's file transfer application.

So, welcome to our new normal:

- A serious flaw is silently discovered in a popular highly used web-connected application.

- Its discoverer remains quiet for years while patiently working out exactly how to set up the attack for maximum effect in the greatest number of cases. There's a bit of a risk/reward tradeoff here, since it's always possible that by not jumping on and exploiting a vulnerability immediately will give it time to be discovered and remediated before an attack. We've often seen this effect when, for example, a well-planned Pwn2Own exploit fails because patches were coincidentally released on the eve of the competition to foreclose the exploit.

- Then, all vulnerable service instances are located ahead of time and the attack is staged and readied.

- And finally, effectively all at once, all of that vulnerable service's users have their data silently exfiltrated and stored. In the case of Clop, all of that sensitive data probably lands in Russia.

- Finally, victim companies are notified, threatened en mass, publicly shamed and eventually, if they don't accede to the extortionist's demands, have their potentially sensitive internal and client data released to the world.

The industry has observed that in this instance the traditional "don't call us, we'll call you" model has been reversed, with the attackers asking their victims to initiate contact. It's been suggested that this is due to the fact that there are just too many victims for the attackers to manage proactively. So they have chosen to be more passive and wait to be contacted. This is probably an optimal strategy, since what the attackers want is maximum extortion payments, and the likelihood of being paid, and being paid a larger amount, is far higher if they are proactively contacted by a concerned victim than if they reach out to cajole.

At this point we don't know how much money this campaign will net for Clop, but the numbers are distressing for ransomware and cyber-extortion gangs in general:

Ryuk	$150 million
REvil	$123 million in 2020
LockBit	$91 million
Darkside	$90 million between October 2020 and May 2021
Maze/Egregor	$75 million
Cuba	$43.9 million throughout 2021
Conti	$25.5 million between July and November 2021
Netwalker	$25 million between March and July 2020
Dharma	$24 million between November 2016 and November 2019

All told, just shy of $650 million dollars.

As this podcast has often observed, malicious hacking is no longer being done for sport. It's now all about money. And, unfortunately, money creates incentive and, as we've also frequently observed here, there are a sufficient number of undiscovered vulnerabilities lurking within much of today's software to incentivize the bad guys into finding and exploiting them for profit.

And it's not as if the bad guys are smarter. As soon as equally talented security researchers began taking a closer look at Progress's MOVEit Transfer software, additional previously unsuspected vulnerabilities started falling out of the thing weekly. That widely used software turned out to be a mess. Yet good guys were never given sufficient prior reason to examine it because the economics of doing so didn't make sense. No security researcher was going to earn millions of dollars by discovering those problems and turning them in for a bounty. But the economics for the bad guys **did and does** make sense, since they will likely manage to extort millions of dollars, overall, from their newly acquired victim base.

Something needs to change. Academics in their ivory towers are busily inventing and developing new computer technologies that have none of these problems. But what we know is that down here on the ground nothing changes unless it is forced to. When I was previously complaining about the utterly and obviously broken traditional model of SQL database access by web servers, I was scolded by our listeners and told "Oh, Steve... don't you know? That was the old way of using SQL?" Right. Old.

Tell that to the thousands of victims of this latest catastrophe of SQL database usage.

And speaking of old. Does everyone know just how old this attack is? Just how old is the exploit that created the "Little Bobby Drop Tables" joke? The operation of the SQL injection exploit was first documented in 1998 by cybersecurity researcher and hacker Jeff Forristal. His findings were published in the hacker magazine Phrack. Writing under the moniker Rain Forest Puppy, Jeff explained how someone with basic coding skills could piggyback unauthorized SQL commands onto legitimate SQL commands to pull sensitive information out of a website's database.

Gee... doesn't that sound a lot like what just happened last week? And that warning came 23 years ago. 23 years ago. This is a fundamental database architecture that was horribly bad then and nothing has changed since. It happened again last week.

In 1998, when Jeff Forristal notified Microsoft about how the vulnerability impacted their very popular SQL Server product, Microsoft didn't see it as a problem. As Forristal put it at the time in his article for Phrack, "According to Microsoft, what you're about to read is not a problem, so don't worry about doing anything to stop it."

So how's that worked out?

In 2007, the biggest convenience store chain in the United States, 7-Eleven, fell victim to a SQLI attack. The Russian hackers used SQL injections to hack into the 7-Eleven website and use that as a stepping stone into the convenience store's customer debit card database. This allowed the hackers to then withdraw cash back home in Russia. Wired magazine reported that the culprits absconded with two million dollars.

That same year, cybercriminals used SQLI to gain administrative control over two US Army-related websites and redirect visitors to websites with anti-American and anti-Israeli propaganda.

The 2008 MySpace data breach ranks as one of the largest attacks on a consumer website. Cybercriminals stole emails, names, and partial passwords of almost 360 million accounts. Thanks to that attack we learned that it wasn't a good idea to reuse passwords across sites.

The award for the most egregious lack of security probably goes to Equifax. The 2017 Equifax data breach which yielded extremely personal information (i.e., names, social security numbers, birth dates, and addresses) for 143 million consumers was, you guessed it, a SQL injection attack. And what's worse, prior to the data breach a cybersecurity research firm had warned Equifax that they were susceptible to a SQLI attack. Whoops.

Every three years the OWASP Open Web Application Security Project ranks the Top 10 Most Critical Web Application Security Risks. Guess where SQL injection ranks? Yep. Number 1.

Our listeners, who kindly took the time to educate me about there being much better and more secure ways to use SQL were absolutely correct. Which only serves to underscore the tragedy of the fact that SQL will still happily operate today the same way it did 23 years ago, in 1998.

Using SQL the horribly and fundamentally insecure way is, unfortunately, also the obvious and easy way. This thing... should have been strangled in its crib the moment it was born in 1998. But, instead, Microsoft and others blessed it and said: “Oh, it’s so wonderfully easy and powerful. Those worry warts are just trying to get some ink. It’s fine, just be careful.”

Right. Be careful walking on ice while carrying that dynamite.