Security Now! #841 - 10-19-21

Minh Duong's Epic Rickroll

This week on Security Now!

This week we, of course, update on various controversies surrounding Win11 and catch up on the aftermath of last week's Patch Tuesday. We note that REvil's brief reappearance appears to have ended — perhaps this time forever — and we examine, just for the record, the outcome of the big, virtual, 30-nation anti-ransomware meeting where the invitations for China and Russia were apparently lost in the mail. We look at the amazing results of this past weekend's Tianfu Cup 2021 hacking competition in China, at the startling success of a prolific botnet's clipboard hijacking module, and at LinkedIn's decision to dramatically pare down its offerings in China. And then, after quickly sharing Sunday's significant news about SpinRite, we're going to take a very fun and detailed look at the sophisticated senior prank orchestrated by Illinois' Minh Duong who miraculously sidestepped his own arrest.

This one speaks for itself…

Windows 11 Watch

hickeyj / @hickeyj

Hi Steve, related to your discussions on Win 11 system requirements - I was interested to see that it was possible to install on a 10 year old PC with a 2nd gen i5 processor - running pretty well and seems to be receiving updates.

I just wanted to close the loop on that question and note that all reports are that all machines, regardless of how they have Windows 11 installed, are reporting that updates are being received. And given the number of people who I see citing that not receiving updates is a strong reason not to move to Windows 11 without Microsoft's full blessing, it's clear that just the threat of being cut off from the continual IV drip of Windows mistake corrections serves as a powerful deterrent ... exactly as Microsoft knew it would... while they proceed to deliver updates to every instance of Windows 11, as of course they will.

Patch Tuesday

And so, last week was October's Patch Tuesday and it was as eventful as most have been recently for Windows. Threatpost characterized one of last week's updates as: "A PrintNightmare Fix to Fix the Other PrintNightmare Fix." And as it turns out, that fix broke other things...

After applying last Tuesday's patches users and administrators of Windows 10 have started reporting wide scale network printing problems. The culprit appears to be KB5006670. Now, different releases of Windows get different cumulative monthly updates. Windows 11 gets KB5006674. Windows 8 receives '6714 and the oldest Windows 10 supported, 1909, receives '6667. But it's '6670 which is used by Windows 10 2004, 20H1, and 21H1 that people are having trouble with. Given the nature of the trouble it's more likely that all releases will be seeing trouble and that reports surround those three Win10 editions since they are by far the most prevalent.

In any event, after installing the '6670 within Win10 networks, users are reporting that they cannot print to network print servers, with some users receiving 0x00000709 or 'Element not found' errors when attempting to print. In online forums, Windows admin have been airing their frustration with the continual printing bugs and have come to a unanimous conclusion: uninstalling this week's updates resolves the problem.

As we know, ever since July, following the PrintNightmare flaws becoming public in June, Microsoft has been scrambling and releasing a stream of apparently half-baked security updates intended to fix the various PrintNightmare vulnerabilities, some in the Windows Print Spooler.

After seeing a bit of this, and looking into the nature of the trouble, I made the observation on this podcast that we appeared to be seeing a true collapse of Windows' printing infrastructure because the bad guys had figured out how to leverage Windows' traditional cross-network print driver auto-installation, which provided Windows with some very popular printing features. Unfortunately, these features had always been exploitable by anyone on the network to elevate their rights and execute their code. But no one had, until recently.

Ever since, Microsoft has been attempting to patch the unpatchable. It's unpatchable, because it's not a bug... It truly is a feature... Point and Print is just a feature that Microsoft probably now regrets... at least as it's currently implemented. So, Microsoft has been attempting to change Windows' Point and Print feature operation. And while these changes at least somewhat mitigate the vulnerabilities, we've been watching as they have created their own set of new problems for enterprise users. Imagine being an Admin for a good sized enterprise whose printing systems keep being broken over and over month after month. It would become a bit tiresome after a while.

Meanwhile, a different issue has beset new Windows 11 users:
Microsoft has confirmed a new and different printing issue for Windows 11; it causes printer installations to fail on systems commonly found in enterprise environments. Microsoft explains that a printer installation might fail when attempted over the network on devices that access printers via print servers using HTTP connections. And that installing printers might also fail when using the Internet Printing Protocol (IPP) in organizations sharing an IPP printer using printer connections. These problems are said to be specific to Windows 11 because they were fixed in either the September or October updates for the earlier operating systems... but not yet for Windows 11. A fix for this is slated for later this month.

Windows received a fix for a spoofing vulnerability:
Last week's release includes a fix for CVE-2021-36970, which is a spoofing vulnerability in, yes, Microsoft Windows' Print Spooler that has a CVSS score of 8.8. Chris Morgan, senior cyber threat intelligence analyst at Digital Shadows, said that the spoofing vulnerability fix Microsoft put out is meant to fix new problems that previous patches have introduced. Okay. Chris said: "While Microsoft provided a fix in their September 2021 update, the patch resulted in a number of new management problems. Certain printers required users to repeatedly input their admin credentials every time an application attempted to print or had a client connect to a print server." Chris added that "Other problems included event logs recording error messages and denying users the ability to perform basic prints. As a result, many users skipped the update due to its operational impact [which is putting it kindly], ultimately leaving the risk posed by PrintNightmare in place."

So, it appears that Windows printing remains tangled. We'll check back next month. The best news is that none of this affects typical end users who generally have local printing environments that have never been impacted by any of this.

A Critical Remote Code Execution affecting Word, Office, SharePoint was fixed:
Another vulnerability of note is CVE-2021-40486, an RCE affecting Word, Office and some versions of SharePoint Server that can be exploited via the Preview Pane.

The vulnerability is reportedly **not** completely new to Microsoft, with several other similar CVEs documented earlier this year. So it sounds as though this might be another of these recent cases of a partial quick fix that didn't repair the underlying problem. The vulnerability has worried security experts because the attack has low complexity, merely requiring a user to open a specially crafted file either by email or via a website. And if on a website, either hosted by the attacker or through a compromised website that accepts or hosts user-provided content. An attacker who successfully exploits this vulnerability can use it to perform actions in the context of the current user. So the code in the opened file could take actions on behalf of the logged-on

user with the same permissions as that user. This doesn't give them administrative access, but as we know, that's where attacks begin, not where they end.

So, there were a total of 74 vulnerabilities of various severities fixed with one being a true 0-day:
The tech press was reporting that there were four 0-days, but there were four critical problems, only one of which was known to be actively exploited. Thus, one 0-day. But that one, tracked as CVE-2021-40449, is an elevation of privilege vulnerability in the Win32k DLL. Earlier this year, Kaspersky researchers discovered that an exploit of this vulnerability was being used to elevate privileges and commandeer Windows servers as part of a Chinese-speaking advanced persistent threat (APT) campaign from the APT threat actor known as "IronHusky".

The exploit chain was observed to terminate with the installation of a newly discovered Remote Access Trojan (RAT) dubbed "MysterySnail" being installed on compromised servers, with the goal of stealing data. A senior manager of vulnerability and threat research at Qualsys said that, if left unpatched, "MysterySnail has the potential to collect and exfiltrate system information from compromised hosts, in addition to other malicious users having the ability to gain complete control over the affected system and launch further attacks."

So, even though Microsoft appears to be chasing their tail with printing, other good things are being fixed.

Ransomware News

REvil may finally be gone for good...
REvil gang has shut down for the second and final time, if we are to believe the group's new leader.

In a message posted on an underground hacking forum, the group's new leader who uses the handle 0_neday — which I'll pronounce as "Oneday" — posted that they lost control over their Tor-based domains.

As we know, the group shutdown without any notice for the first time on July 13 this year, coincident with one of their affiliates attacking Kaseya's servers over the 4th of July US holiday weekend and hit thousands of businesses. Being called the largest set of ransomware attacks in history drew a great deal of unwanted attention.

We were later to learn that the decision to shut down operations was taken by the group's leader known as UNKN, who took down servers and disappeared with the group's money, which left them unable to pay many of their affiliates—other groups who were helping REvil execute attacks and were splitting the profits.

Then, early last month, the group, minus UNKN, made a formal return using the same REvil name. To prove they were the same group as before, this new REvil incarnation restored all of their former Tor-hosted portals, such as their victim payment/extortion portal and data leak site. And as soon as they returned, the group's members began launching new attacks.

But Sunday, the day before yesterday, in a series of messages spotted by an analyst with Recorded Future, the group's new administrator, that guy calling himself 0_neday, said that a third party had compromised their Tor-based portal. He posted: *"The server was compromised and they were looking for me."* He said: *"To be precise, they deleted the path to my hidden service in the torrc file and raised their own so that I would go there."* So, 0_neday was saying that someone had created a clone of the legitimate REvil Tor backend panel in the hopes of catching him.

Things were not going well for the REvil gang anyway, as they were still dealing with the fallout following their July shutdown and theft of affiliate funds. Several affiliates were still trying to recover funds stolen by UNKN, and the group's developers were also accused of hiding a backdoor inside their code. The backdoor allegedly allowed the REvil admins to provide decryption keys to victims directly and force affiliates out of ransom negotiations and payment.

Since the cybercriminal underworld is primarily driven by reputation and trust (what was that about honor among thieves?), this may have been inevitable with the writing on the wall for 0_neday, who decided to shut down the REvil operation permanently, rather than deal with the gang's ever-increasing reputational trouble which probably became unsurvivable once its servers were known to have been compromised over the weekend.

The Recorded Future analyst told The Record: *"I really hope we just witnessed an offensive operation by the US government. That is how you deal with cybercriminals – using their own methods against them. Release the Hounds!"*

Over 30 Countries Pledge to Fight Ransomware Attacks

Representatives from the U.S., the European Union, and 30 other countries have pledged to mitigate the risk of ransomware and harden the financial system from exploitation with the goal of disrupting the ecosystem, calling it an *"escalating global security threat with serious economic and security consequences."* Yeah, no kidding. I'm highly skeptical about whether anything can have any measurable effect, but I wanted to quickly share what was produced and the nature of the saber rattling. There were also some interesting bitcoin account translation statistics:

So, in a statement released last week following their meeting, officials said: *"From malign operations against local health providers that endanger patient care, to those directed at businesses that limit their ability to provide fuel, groceries, or other goods to the public, ransomware poses a significant risk to critical infrastructure, essential services, public safety, consumer protection and privacy, and economic prosperity."*

[Right... Blah blah blah]

And, to that end, efforts are expected to be made to enhance network resilience by adopting cyber hygiene good practices, such as using strong passwords, securing accounts with multi-factor authentication, maintaining periodic offline data backups, keeping software up-to-date, and offering training to prevent clicking suspicious links or opening untrusted documents.

[As we know, none of that's bad, but it's also all already well established best practice.]

Besides promoting incident information sharing between ransomware victims and relevant law enforcement and cyber emergency response teams (CERTs), the initiative aims to improve mechanisms put in place to effectively respond to such attacks, while also countering the abuse of financial infrastructure to launder ransom payments.

[Okay, now that might be interesting.]

The joint bulletin was issued by Ministers and Representatives of Australia, Brazil, Bulgaria, Canada, Czech Republic, the Dominican Republic, Estonia, European Union, France, Germany, India, Ireland, Israel, Italy, Japan, Kenya, Lithuania, Mexico, the Netherlands, New Zealand, Nigeria, Poland, Republic of Korea, Romania, Singapore, South Africa, Sweden, Switzerland, Ukraine, the U.A.E, the U.K., and the U.S. Notably absent from the list were China and Russia. I guess you don't want the fax guarding the chicken coop!

The international counter-ransomware collaboration comes as illicit payments topped nearly $500 million globally in the last two years alone — $400 million in 2020 and $81 million in the first quarter of 2021 — necessitating the payment flows that make the activities profitable are subject to anti-money laundering regulations and the networks that facilitate these payments are held accountable.

In late September 2021, the U.S. Treasury Department imposed sanctions on Russian cryptocurrency exchange "Suex" for helping threat actors launder transactions from at least eight ransomware variants, marking the first instance of such an action against a virtual currency exchange. The U.S. government said: *"Treasury will continue to disrupt and hold accountable these ransomware actors and their money laundering networks to reduce the incentive for cybercriminals to continue to conduct these attack."*

The development also comes following an independent report published by the department's Financial Crimes Enforcement Network (FinCEN), which potentially tied roughly $5.2 billion worth of outgoing Bitcoin transactions to 10 most commonly reported ransomware variants, in addition to identifying 177 unique wallet addresses used for ransomware-related payments based on an analysis of 2,184 suspicious activity reports (SARs) filed between January 1, 2011, and June 30, 2021.

[Okay, so that's $5.2 billion worth of outgoing Bitcoin transactions. But it's spread over a period of the last ten years and I wonder whether the historical or the present value of Bitcoin was used for that summary. As we know, it's worth way more now than it was back then. At the same time, the bulk of the high-value ransoms have been recent.]

In the first half of 2021 alone, ransomware-based financial activity is estimated to have extracted at least $590 million for the threat actors, with the mean average total monthly suspicious amount of ransomware transactions pegged at $66.4 million. The most commonly reported variants were REvil (aka Sodinokibi), Conti, DarkSide, Avaddon, and Phobos.

The Counter-Ransomware Initiative hopes to drain their funding and take down their operations by disrupting the ransomware groups' funding channels.

<quote> "We acknowledge that uneven global implementation of the standards of the Financial Action Task Force (FATF) to virtual assets and virtual asset service providers (VASPs) creates an environment permissive to jurisdictional arbitrage by malicious actors seeking platforms to move illicit proceeds without being subject to appropriate anti-money laundering (AML) and other obligations."

"We are dedicated to enhancing our efforts to disrupt the ransomware business model and associated money-laundering activities, including through ensuring our national AML frameworks effectively identify and mitigate risks associated with VASPs and related activities."

The efforts to disrupt ransomware groups' abuse of cryptocurrency will include regulators, financial intelligence units, and law enforcement regulating, supervising, investigating, and taking action against virtual asset exploitation.

"We will also seek out ways to cooperate with the virtual asset industry to enhance ransomware-related information sharing"

The statement noted that *"Financial institutions play an important role in protecting the U.S. financial system from ransomware-related threats through compliance with BSA obligations. Financial institutions should determine if a SAR filing is required or appropriate when dealing with a ransomware incident, including ransomware-related payments made by financial institutions that are victims of ransomware."*

We've observed that the ransomware scourge has largely been enabled by the existence of a means of transferring payment anonymously and untraceably. So attacking the payment chain, to whatever degree that's possible, might at least have some hope of success. But I doubt that telling potential victims to alter their behavior will have any discernible effect. There was just too many fish in the Internet sea.

Security News

Tianfu Cup 2021

Windows 10, iOS 15, Google Chrome, Apple Safari, [not surprisingly] Exchange Server, and Ubuntu 20 were all successfully hacked, broken into and compromised using original, never-before-seen exploits during the just completed 2021 Tianfu Cup, the 4th edition of the international cybersecurity contest held in Chengdu, China.

The competition's targets this year included Google Chrome running on Windows 10 21H1, Apple Safari running on Macbook Pro, Adobe PDF Reader (choose your platform), Docker CE, Ubuntu 20/CentOS 8, Exchange Server 2019, Windows 10, VMware Workstation, VMware ESXi, Parallels Desktop, iPhone 13 Pro running iOS 15, domestic mobile phones running Android, QEMU VM, the Synology DS220j DiskStation, and the ASUS RT-AX56U router.

Our long-time listeners will recall that this Chinese version of Pwn2Own was started three year ago, in 2018, after Chinese government regulations barred their wonderfully competent and capable home grown security researchers from participating in international hacking competitions due to national security concerns. (Or perhaps over worries that they might not choose to return home.)

Chinese security researchers took home $1.88 million after competing and hacking over this past weekend, October 16 and 17. The grand winners were researchers from the Chinese security firm Kunlun Lab, who took home $654,500, a third of the total purse.

In July, the organizers announced a series of targets, and participants had until last weekend to target and prepare exploits that they would execute on devices provided by the organizers on the contest's stage. Each team had three 5-minute attempts to run their exploits, and they could register to hack multiple devices if they wished to increase their winnings.

Overall, there were 16 possible targets and 11 participants mounting successful exploits against 13 of those 16. With the exception of the Synology DS220j NAS, the Xiaomi Mi 11 smartphone, and an unnamed Chinese electric vehicle which no one elected to target, attacks were successfully mounted against every other target:

- Windows 10 – hacked 5 times
- Adobe PDF Reader – 4 times
- Ubuntu 20 – 4 times
- Parallels VM – 3 times
- iOS 15 – 3 times
- Apple Safari – 2 times
- Google Chrome – 2 times
- ASUS AX56U router – 2 times
- Docker CE – 1 time
- VMWare ESXi – 1 time
- VMWare Workstation – 1 time
- qemu VM – 1 time
- Microsoft Exchange – 1 time

Most of the exploits were privilege escalation and remote execution bugs; however, 2 of the exploits presented stood out:

- The first was a zero-click zero-interaction remote code execution attack chain against a fully patched iOS 15 running on the latest iPhone 13.

- The second was a simple two-step remote code execution chain against Google's Chrome, which is something that has not been seen in any hacking competition in years.

And two competition-related tweets were noteworthy:

- "The iPhone 13 Pro Safari escaped from prison remotely, and Chian Pangu won the highest single bonus of $300,000 in history."

- "First confirmed entry for day 1 of TianfuCup, Kunlun Lab pwned Google Chrome to get Windows system kernel level privilege with only two bugs. First time since 2015 as I remembered."

And aside from the competition, many Western eyes were on this year's contest for another reason: One of the iOS exploits showcased at **last** year's competition was used in a cyber-espionage campaign carried out by the Beijing regime against its Uyghur (wee-guhr) population. That observation has reinforced the belief among Western security experts that Beijing may have forbidden Chinese security researchers from participating in hacking contests held abroad in order to better harness their exploit-creating capabilities for its own purposes.

Clipboard Hijacking for fun and profit

We've talked about clipboard hijacking, the process whereby malware waits patiently in a system, silently pinging the system-wide clipboard, looking for the appearance of a valid cryptocurrency wallet address. And, when found, would wait for the user to paste that contents, then replace the pasted contents on the fly with one of its own addresses that it controlled. In this sneaky way, the unwitting user would be irretrievably sending their money to the hackers.

But just how much cryptocurrency could such attacks net?

Would you believe: $24.7 million in Bitcoin, Ether, and Dogecoin?!?!

First spotted in 2016, the MyKings botnet has been one of the most sprawling malware operations in recent years. The gang behind this Botnet, also referred to as the Smominru or DarkCloud botnet, operates by scanning the Internet for exposed Windows or Linux systems running unpatched software.

Using known exploits for those unpatched vulnerabilities, the MyKings gang infects these servers and then moves to move laterally inside their networks. Reports published through the years by Guardicore, Proofpoint, Qihoo 360, VMWare's Carbon Black, and Sophos have described MyKings as one of the largest malware botnets that has been created over the past decade, with the number of infected systems sometimes easily totaling more than 500,000 hacked systems.

In its early years, the botnet was primarily seen deploying a Monero cryptocurrency miner on infected hosts to directly generate profits for the botnet's operators. And a January 2018 report by Proofpoint estimated the group's profits at the time at around $3.6 million, based on the Monero funds they found in some wallets they linked to the group.

But through the years, the MyKings group's operations and malware have evolved from a hack-and-mine operation, the botnet became a Swiss army knife of nastiness, with all sorts of modules for moving across internal networks, spreading like a worm, and carrying out various attacks.

In 2019, Sophos said that one of the new modules it spotted was that "clipboard hijacker" and at that time Sophos had concluded that this MyKings clipboard hijacking module wasn't that successful or widely used, "never received more than a few dollars," and that stealing cryptocurrency by hijacking the clipboard didn't look like "the most profitable operation of MyKings."

But, in a report published just this week, Avast said that since 2019, MyKings appears to have perfected this module, which now detects addresses for 20 different cryptocurrencies. The Avast

researchers said they analyzed more than 6,700 samples of the MyKings malware to identify and extract more than 1,300 cryptocurrency addresses used by the gang to collect funds.

In these addresses, researchers said they found more than $24.7 million in Bitcoin, Ether, and Dogecoin.

Cryptocurrency	Earnings in USD	Earnings in cryptocurrency
Bitcoin	$ 6,626,146.252	132.212 [BTC]
Ethereum	$ 7,429,429.508	2,158.402 [ETH]
Dogecoin	$10,652,144.070	44,618,283.601 [DOGE]

The Avast researchers said: "We can safely assume that this number is in reality higher, because the totals we show consist of money gained in only three cryptocurrencies from the more than 20 in total used by the malware." While the researchers said that some funds were linked to MyKings' past cryptocurrency mining activity, the vast majority appears to have come from the overwhelming success of the clipboard hijacking module.

Avast said that since the beginning of 2020, its own A/V software had detected and flagged MyKings malware attacks on more than 144,000 computers. But since the users of their A/V represent a small fraction of all users, the number of system attacks is certainly much higher.

As a result of these just-published findings, malware analysts have completely changed how they are viewing this botnet. With the ability to carry out large-scale exploitation attacks, a way to profit from their operations, a large number of infected hosts, and the ability to download and run any additional payload the MyKings operators wish, the botnet has established itself as one of the most dangerous malware operations today.

LinkedIn to dramatically pare down its offering in China.
LinkedIn has, for some time, been the only major American social-network allowed to operate in China. But last Thursday the Microsoft-owned company announced that it would be dramatically slimming down its offering and, until that's ready, pulling up stakes and shuttering its platform in China. This marks the biggest departure from China by a major tech company in years.

LinkedIn said that *"significantly more challenging operating environment and greater compliance requirements"* by Beijing authorities were behind the decision. Other social-media services like Twitter and Facebook have been blocked in China for years. Those companys' inability to control what is posted on their sites disqualified them for presence within China. Until now, LinkedIn had been able to maintain its presence only because it censored many of the posts made by its users.

They said: *"While we found success in helping Chinese members find jobs and economic opportunity, we have not found that same level of success in the more social aspects of sharing and staying informed."*

Even so, LinkedIn ran afoul of Chinese Internet content regulators in March when China's Internet watchdog, the Cyberspace Administration of China (CAC) warned LinkedIn that it was failing to control what CAC saw as objectionable political content. The regulator told LinkedIn it

had to do better. In response, the company wrote a kind of self-criticism and filed it with the CAC. Around that same time, the company announced publicly that it would "temporarily" suspend new sign-ups.

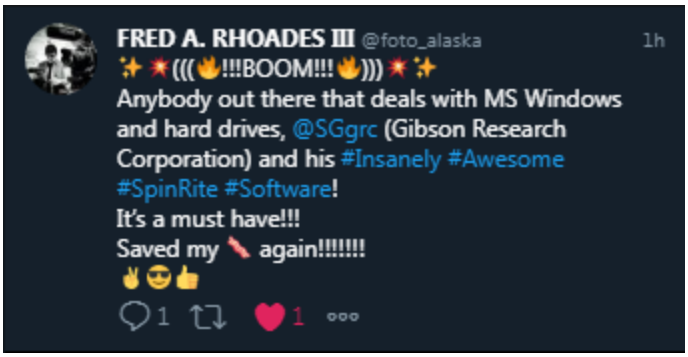
In their statement Thursday, LinkedIn made clear that while it was trying to abide by local regulations, in the end doing so became too much. "We're also facing a significantly more challenging operating environment and greater compliance requirements in China."

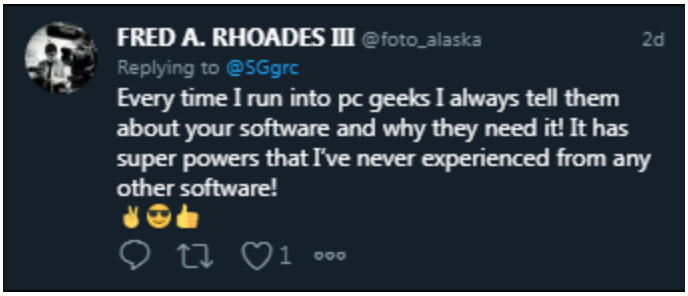
And as I said, LinkedIn is not abandoning China completely. The company said it will eventually offer its 50 million Chinese members a slimmed down version of the platform: an app focused just on job postings. Chinese users will not be able to share or comment on posts — which has been a key social media feature of LinkedIn's platform elsewhere in the world

In 2014, when LinkedIn began working in China, it said it was a global platform "with an obligation to respect the laws that apply to us, including adhering to Chinese government regulations for our localized version of LinkedIn in China."

The company even sold a stake of its Chinese operation to local venture capital partners and said it would be able to abide by local law by using software algorithms and human reviewers to make sure posts did not offend Beijing. But, apparently, that was insufficient.

Closing the Loop





SpinRite

Yesterday, I posted the 4th pre-release of SpinRite to the GRC spinrite.dev newsgroup. The gang there has been having a field day running in one all of their multiple PCs and reporting their results. I have a punch list of things to fix, and I have an idea for a cool new surprise feature for the benchmark's conclusion screen. So I'll be working to get everything resolved and running before I move forward again.

This is a key juncture, because everything from here on out builds upon the foundation that we're working to make bulletproof now. In other words, there isn't anything else that can go wrong once a descendant of this 4th release is running for everyone on all of their hardware.

We're locating all of their system's controllers wherever they are, and all of the drives attached to each one. We're determining how to best communicate with each drive through its controller, then doing so with that method and performing read and write confidence tests to verify everything. The fact that we're now benchmarking these drives in SpinRite and seeing hundreds of megabytes per second of throughput means that we'll be seeing many gigabytes per minute and terabytes per hour to deliver vastly faster performance than we've ever had while also improving SpinRite's ability to recovery data from trouble mass storage devices. So... Yay!

Minh Duong's Epic Rickroll

Just to start us all off on the same page, Wikipedia explains "Rickrolling" as...

> *Rickrolling, alternatively Rick-rolling or Rickroll, is a prank and an Internet meme involving an unexpected appearance of the music video for the 1987 song "Never Gonna Give You Up," performed by the English singer Rick Astley. The meme is a type of bait and switch using a disguised hyperlink that leads to the music video. When victims click on a seemingly unrelated link, the site with the music video loads instead of what was expected, and in doing so they are said to have been "Rickrolled". The meme has also extended to using the song's lyrics, or singing it, in unexpected contexts.*

Upon learning that a quite industrious Illinois High School Senior, by the name of Minh Duong (he's Vietnamese), had deeply hacked not only his own high school's network, but the networks of his entire High School District, my initial dread was imagining that he'd been immediately arrested by District officials who lacked any sense of perspective or humor. After all, these days apparently just using the "View Source" menu item on a web browser and then... viewing the source all it takes. The good news is, the nature and intent of this prank was kept in perspective and District 214's cybersecurity was improved as a result.

So what did Minh do, and how did he do it?

I should first explain that this didn't just happen. The Rickrolling event began at 11:55am on Friday, April 30th, a time carefully chosen so as to be minimally disruptive and intrusive. Minh has since graduated and is now studying cybersecurity at the nearby University of Illinois. But the nature, breadth and depth of this epic hack came to the cybersecurity industry's attention when he, for the first time, posted the whole backstory on his "WhiteHoodHacker.net" blog.

https://whitehoodhacker.net/posts/2021-10-04-the-big-rick

To discharge the suspense of what Minh's fellow students experienced, this was the culmination of several years of planning. The operation was code named "Big Rick." At 10:55 AM on Friday, April 30th, all of the presentation screens and projectors in every class in every high school in the district switched on. The district uses the AvediaPlayer IoT device as the common interface for all classroom screens. At first, the only thing displayed, simultaneously on every screen in the district, was a message stating that an important announcement was forthcoming with a timer patiently counting down from 5 minutes.

Nothing like this had ever appeared before, so naturally, when the timer hit zero, everyone was waiting to see what the announcement would be. They immediately realized that it was a sophisticated prank when Rick Astley appeared on every screen and began crooning the well known lyrics to "Never Gonna Give You Up." The video ran for 10 minutes then shutdown and the entire system reverted to its normal operation as though nothing out of the ordinary had happened.

But the prank wasn't quite complete. At 2:05 PM, all of the school bells rang signalling the end of a class, just as they should . . . except that instead of a bell sound, they played the song again. So everyone got Rickrolled a second time.
After that, Minh immediately sent a 26-page report to the school district, outlining exactly how he and his friends had pulled it off. And because of that, the district decided not to press charges. The director of technology had the class to **thank** them for finding a flaw in their system.

So now that we know what Minh & Company did, let's get his perspective…

> But before I go any further, for any of our younger listeners, PLEASE do not take this one-off success, which fortunately had a very happy ending, as **ANY** form of permission to do anything similar. Minh was fortunate. He was not entitled to receive the leniency that he was given. Make no mistake, there are computer and network intrusion laws on the books that Minh **absolutely** violated. It could so very easily have gone so very wrong for him and his friends. The decision could have been to make an example out of them with a zero-tolerance policy — and I would bet that there was some discussion to that end. So I do not mean to be glamorizing something that I myself would **never** consider doing today. Given my history, I'm quite certain that I would have been foolish enough to do it when I was 18. Everyone knows the story of the portable dog killer and some of my other youthful antics which I somehow survived without a police record. But it's been 48 years and the Internet since I was 18, and times have really changed since then. The grown-ups are terrified of the technology they don't understand and fear that they cannot control. So poking them with a stick, or with a ping packet, these days is probably not the best idea. PLEASE don't do it. Really.

Okay. With that said, here's how Minh recently described the "Big Rick" hack. I want to share this with this podcast's listeners because there are some wonderfully fun techie details that really service to bring it to life...

On April 30th, 2021, I rickrolled my high school district. Not just my school, but the entirety of Township High School District 214. It is one of the largest high school districts in Illinois, consisting of 6 different schools with over 11,000 enrolled students.

This story isn't one of those typical rickrolls where students sneak Rick Astley into presentations, talent shows, or Zoom calls. I did it by hijacking every networked display in every school to broadcast "Never Gonna Give You Up" in perfect synchronization. Whether it was a TV in a hall, a projector in a classroom, or a jumbotron displaying the lunch menu, as long as it was networked, I hacked it!

In this post, I'll be explaining how I did it and how I evaded detection, as well as the aftermath when I revealed myself and didn't get into trouble.

Okay... now, clearly recognizing the same danger I do, Minh then places a clear "Disclaimer" in his posting, writing:

> *This post is for educational purposes only. Do not perform similar activities without explicit permission.*

We prepared complete documentation of everything we did, including recommendations to remediate the vulnerabilities we discovered. We sent a comprehensive, 26-page penetration test report to the D214 tech team and worked with them to help secure their network.
With that said, what we did was very illegal, and other administrations may have pressed charges. We are grateful that the D214 administration was so understanding.

Initial Access

This story starts with my freshman year when I did not have much technical discipline — a time that I can only describe as the beginning of my script kiddie phase. I didn't understand basic ethics or responsible disclosure and jumped at every opportunity to break something.

So obviously, I became curious about the technology at my high school. And by "curious," I mean port scanning the entire IP range of the internal district network.

I had a few friends help out with this project — and oh boy, did we scan! Our scanning generated so much traffic that our school's technology supervisor caught wind of it and came in at one point to ask us to stop. Of course, we did so immediately, but by then, we had finished scanning the first half of the district's 10.0.0.0/8 address space — a total of 8,388,606 IPs!

From the results, we found various devices exposed on the district network. These included printers, IP phones... and even security cameras without any password authentication!

This is where I state the disclaimer again: never access other systems in an unauthorized manner without permission.

The district tech team was informed about the issue, which they resolved by placing the cameras behind ACL restrictions. However, many devices remained exposed to the student network — more importantly for this post, the IPTV system!

[I'll just interject here to observe that the phrase "many devices remained exposed to the student network" should horrify any IT administrator. Having high school students sharing a network that also contains administrative functions is insane all by itself. It's about 1,000 times worse than having IoT power outlets and light switches phoning home to hostile foreign nations. If ever there was a case to be made for network segmentation, elementary school, junior high and high schools are it. No one should even consider allowed those precious little darlings anywhere near administrative network functions. Any network that students have access to should be able to touch the Internet and nothing more.]

Exterity IPTV System

Before moving on, I will briefly explain the IPTV system. The system is composed of three products:

- *AvediaPlayer (receivers)*
- *AvediaStream (encoders)*
- *AvediaServer (management)*

AvediaPlayers are small blue boxes that connect to projectors and TVs. They can send serial commands to their respective device to turn the display on/off, change inputs/volume, switch channels, etc. These receivers include both a web interface and an SSH server to execute the serial commands. Additionally, they run embedded Linux with BusyBox tools and use some obscure CPU architecture designed for IoT devices called ARC (Argonaut RISC Core).
Next, AvediaStream encoders connect to devices that broadcast live video. They encode the live feed coming from these devices to the AvediaPlayer receivers, which display the stream. Encoders are attached to computers that need to broadcast a stream, such as text carousels or morning announcements. These also have embedded software similar to the AvediaPlayers.

Last but not least, AvediaServers allow administrators to control all receivers and encoders at once. These have typical x86_64 processors and run the enterprise Linux distribution, CentOS. Like the receivers and encoders, they also have web interfaces and SSH servers.

Since freshman year, I had complete access to the IPTV system. I only messed around with it a few times and had plans for a senior prank, but it moved to the back of my mind and eventually went forgotten.

Preparation
Fast forward to the second semester of senior year, early 2021: all the schools were doing hybrid instruction because of the COVID-19 pandemic. Up to this point, in-person instruction was opt-in, with most students staying remote, including myself. But in March, the superintendent announced that in-person instruction would switch to an opt-out model on April 5th.

Since almost all students would be back in school, I realized that a senior prank involving the IPTV system was now worthwhile. A few days later, I decided to share my thoughts with a few close friends. I gathered a small team across the district and started preparing. We began to refer to the operation as "the Big Rick."

1. C2 Payload and Exploitation
The first thing we focused on was figuring out how to control all the projectors at once. While we could send commands to each receiver using a web interface, it would not be ideal spamming HTTP traffic to every receiver simultaneously.

Instead, I used the SSH access on each receiver as the command-and-control (C2) channel. I developed a simple shell script that would serve as a staged payload to be uploaded to each receiver ahead of time. This script contained various functions that could execute requests to the web interface locally on the receiver. Thanks to the increased flexibility from the payload, I could also back up and restore receiver settings to the filesystem after the rickroll was over.

In the actual payload, I repeatedly looped commands to keep the rickroll running. For example, every 10 seconds, the display would power on and set the maximum volume. This way, if someone attempted to power off the projector or mute it, it would revert and continue playing. The only way to shut it off would be to pull the plug or change the input source. (Looping input

changes causes flashes even if the current source is the same as the latest source. I had to rely on a failsafe input switch that activated right before the rickroll started to ensure everyone was tuned in. You can see this flash in the video at the 48-second countdown.)

The vulnerabilities exploited to gain initial access were implementation-specific. (In other words, the district's tech team was at fault for using default passwords). However, I discovered vendor privilege escalation vulnerabilities in all of Exterity's IPTV products, allowing me to gain root access across all systems. One of these bugs was a simple GTFO-bin, but the other two are novel vulnerabilities that I cannot (and should not) publish.

["GTFOBin" is a reference to a curated list of Unix (and Linux) binaries that can be used to bypass local security restrictions in misconfigured systems. So Minh found a command on the system that the could leverage due a fault in some security confirmation.]

2. RTP Multicast Stream

The next issue we tackled was setting up a custom video stream to play the rickroll in real-time. We needed to broadcast multicast traffic, but only the AvediaStream encoders or the AvediaServers could do this because of ACL restrictions.

Setting up the stream was arguably the most time-consuming part of preparation because testing was an absolute pain. I only needed a single projector for development, but it's not easy when classes are using them during the day.

So I tested at night instead! I would remotely connect to one of the PCs in the computer lab with the front camera facing the projector. Then, I would record a video to test if the projector displayed the stream correctly!

[So, just to expand upon that a bit: Minh setup a PC in the computer lab, to which he would be able to gain remote access from home in the evening, with its web cam pointing at the classroom's presentation screen to record whatever the screen would show as he was developing the code to takeover the entire district.]

The lag seen in the video is one of the earlier issues I faced with the stream. It turned out trying to redirect UDP traffic through the AvediaStream encoders added too much latency. I fixed this by broadcasting to multicast directly from an AvediaServer using ffmpeg.

Hopefully, I didn't scare any late-night staff!

3. An Unexpected Development

It was April 27th, a mere three days away from the Big Rick finale, when one of my peers discovered a new IP range full of IoT devices after a scan. It turns out it was the recently installed bell system, called Education Paging and Intercom Communications (EPIC). The majority of the devices in this range were speakers found in hallways, classrooms, etc.

Similar to how AvediaPlayers linked to AvediaServers, each speaker connected to an EPIC server for their respective school. These servers had a web interface locked behind a login page.

Only a single EPIC server had default credentials configured. We were able to modify the bell

schedule at will, as well as upload custom audio tones. We could change the bells to play "Never Gonna Give You Up" instead!

However, we only had access to this individual school's EPIC system since it was the only one with vulnerable credentials. Or was it?

I discovered that the EPIC server we compromised performed weekly backups of its configuration to an external SMB file share. The credentials for this SMB server were the same default credentials for the EPIC system. Each backup included an SQL dump of account usernames and password hashes.
Well, what if the other EPIC systems have backup servers as well? And since these backup servers are separate from the EPIC servers, they might still use default credentials!

This scenario was precisely the case! From there, I was able to access the password hashes for the other EPIC servers and identify a local admin account available across all the EPIC servers. After some password cracking, we effectively had control over all the bell schedules in the district!

Execution
One of our top priorities was to avoid disrupting classes, meaning we could only pull off the prank before school started, during passing periods, or after school. Before the pandemic, some schools would start earlier, some would start later, some had block scheduling, and some would have all their periods in one day. Conveniently, due to COVID-19, all the high schools in the district were now on the same block schedule, so we didn't have to worry about scheduling on a per-school basis.

Another thing was that final exams were right around the corner. The biggest concern was standardized testing, which wouldn't have breaks during passing periods. We decided on April 30th, which was the Friday before AP exams started. We surveyed extensively to check if any significant tests were happening on this day. We were fully prepared to abort if we learned any standardized testing was taking place.

In the weeks before the Big Rick, we staged the C2 payload on all the AvediaPlayers in an automated manner, carefully spreading our actions to avoid detection. On the day of the Big Rick, we used two of the seven AvediaServers as the C2 masters, which would connect to all the AvediaPlayers and execute the payloads.

On April 30th:

Time	***Event***
10:40 AM	*Rickroll stream goes live with a 20-minute countdown.*
10:55 AM	*AvediaPlayer systems are initialized, turning on displays and changing the Active channel to the rickroll stream.*
11:00 AM	*The stream finishes the countdown with the rickroll playing at the end of the first block.*
11:10 AM	*The payload restores the AvediaPlayer systems to their previous state and removes itself.*
2:05 PM	*The end of the third block bell plays a rickroll instead of the dismissal bell.*
2:15 PM	*The penetration test report is automatically sent to the technical supervisors.*

We also scheduled another modified bell for 3:25 PM. If district tech still hadn't figured out what had happened to revert the bells, a 1-minute version of the 3-second dismissal bell would play at the end of the day. They did figure it out.

The Aftermath

A few days after sending the report through the anonymous email account, we received an email response from D214's Director of Technology. The director stated that because of our guidelines and documentation, the district would not be pursuing discipline. In fact, he thanked us for our findings and wanted us to present a debrief to the tech team! Later, he revealed the superintendents themselves reviewed and were impressed by our report! 😯

I was ecstatic that the administration was open to remediating their problems and auditing them with us. Although the D214 administration communicated good intentions (and they did hold in the future), my peers did not trust the administration and were skeptical of the true nature of the meeting — one of them referred to the whole thing as a sting operation!

We decided I would reveal myself to present our debrief slides with the others remaining anonymous in the Zoom meeting. I had planned on announcing my involvement from the beginning since I wanted to publish this blog post. (I was also pretty much the prime suspect anyways.) But, just in case, I scheduled the debrief to take place after I graduated.

In all seriousness, the debrief went extremely well and was productive for everyone. We answered clarifying questions from the tech team and gave additional tips for remediation. We even managed to get the district to look into expanding the IT/cybersecurity program and hopefully, sponsoring a D214 CTF? :o [CTF = Capture the Flag competition]

This has been one of the most remarkable experiences I ever had in high school and I thank everyone who helped support me. That's all and thanks for reading!