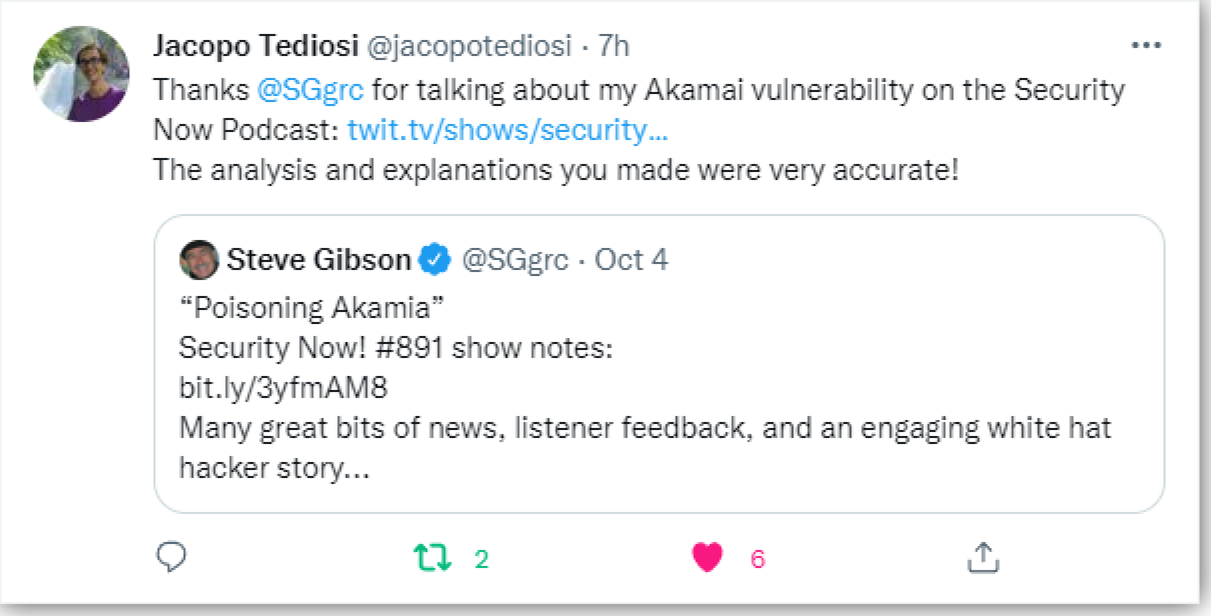
Security Now! #892 - 10-11-22

Source Port Randomization

This week on Security Now!

This week we look at a massive customer information leak from a surprising source. Meta notes where their users are being harvested and in an industry first, Uber's CTO has been convicted. We have more, much more, cryptocurrency industry turmoil and a new appointee in the UK wants to drop their use of the GDPR. The NSA is looking for next summer interns, IBM learns that incident responders are feeling quite stressed out, and Microsoft continues to fumble their Exchange Server response. I have news of SpinRite and of my discovery of a lovely little Single Board Computer. And after sharing some listener feedback we're going to look at a recent mistake made in the Linux kernel that allowed its users to be tracking online.

DIY Hybrid?



Jacopo Tediosi @jacopotediosi · 7h

Thanks @SGgrc for talking about my Akamai vulnerability on the Security Now Podcast: twit.tv/shows/security...
The analysis and explanations you made were very accurate!

Steve Gibson @SGgrc · Oct 4

"Poisoning Akamia"
Security Now! #891 show notes:
bit.ly/3yfmAM8
Many great bits of news, listener feedback, and an engaging white hat hacker story...

Security News

A massive breach of customer information

It turns out that there's a non-security breach way for a user of a cryptocurrency exchange to have their name, account balance and all transaction IDs exposed to the public: That's if the currency exchange files for bankruptcy.

The "Celsius Network" cryptocurrency platform deliberately exposed the names and complete transaction histories of hundreds of thousands of its customers. Okay now, just as an aside, "hundreds of thousands of its customers"?? What most mystifies me is how these random also-ran startups acquire hundreds of thousands of customers? What are people thinking? Who are these people? Anyway... The company filed a [get this] 14,532-page document as part of its bankruptcy proceedings the week before last that contained the names and recent transactions of every user on the platform. The judge allowed the company to redact the document, but only their customers' physical and email addresses because the rest of the information was required in their disclosure during regular bankruptcy procedures. The document is available via PACER and other legal document portals.

So, not so private if the cryptocurrency platform you're using goes belly-up.

Meta-targeted Malware

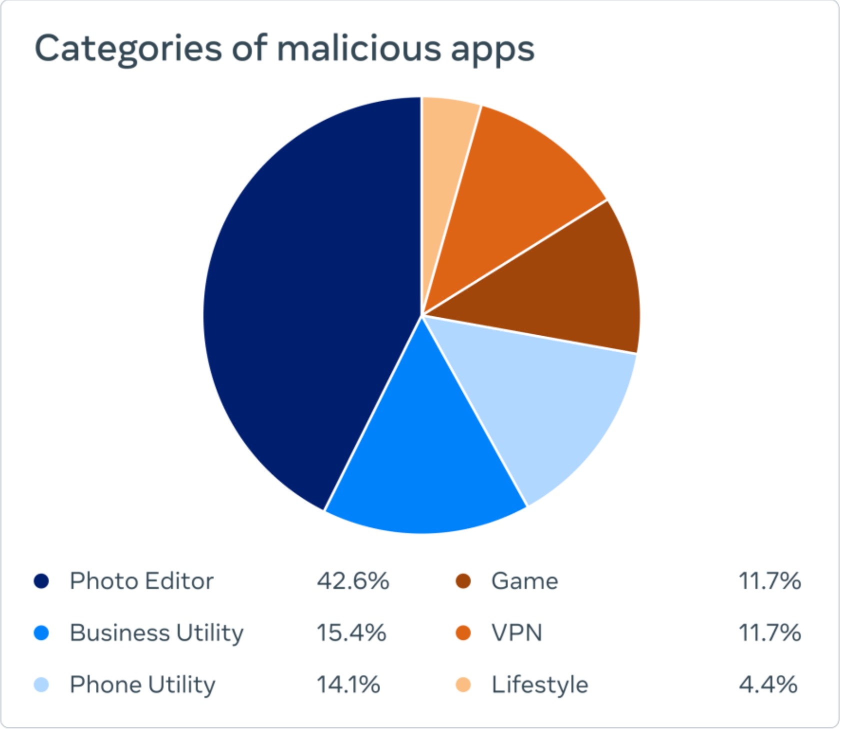
A posting last Friday by two security-focused employees of Meta (you know, Facebook's parent) disclosed the results of a recent search through the Apple and Google app stores. They explained that they had identified more than 400 malicious Android and iOS apps targeting Facebook's users and being used to steal their Facebook login credentials. They reported their findings to Apple and Google and have asked the users they identified to change their passwords since their credentials have almost certainly been compromised.

The nature of the come-ons to entice the downloads was interesting, they were:

- Photo editors, including those that claim to allow you to “turn yourself into a cartoon”
- VPNs claiming to boost browsing speed or grant access to blocked content or websites
- Phone utilities such as flashlight apps that claim to brighten your phone’s flashlight
- Mobile games falsely promising high-quality 3D graphics
- Health and lifestyle apps such as horoscopes and fitness trackers
- Business or ad management apps claiming to provide hidden or unauthorized features not found in official apps by tech platforms.

Interestingly, by far the majority at nearly half – 42.6% – of the 400+ apps were Photo Editors. Next at 15.4% was business utilities, then phone utilities at 14.1% and games at 11.7%.

The standard advice applies. Try hard to avoid downloading tasty looking goodies. And don’t immediately click on the download link without doing as much reputational research as possible.



Uber's Chief Security Officer found guilty
Uber's former CSO — Chief Security Officer — Joe Sullivan was found guilty at trial due to his actions following a 2016 data breach at Uber. Reading from a statement made on August 20th, 2020 in the Northern District of California:

> *The complaint describes how Sullivan played a pivotal role in responding to FTC inquiries about Uber’s cyber security. Uber had been hacked in September of 2014 and the FTC was gathering information about that 2014 breach. The FTC demanded responses to written questions and required Uber to designate an officer to provide testimony under oath on a variety of topics.*

> *Sullivan assisted in the preparation of Uber's responses to the written questions and was designated to provide sworn testimony on a variety of issues. On November 14, 2016, approximately 10 days after providing his testimony to the FTC, Sullivan received an email from a hacker informing him that Uber had been breached again. Sullivan's team was able to confirm the breach within 24 hours of his receipt of the email.*
>
> *Rather than report the 2016 breach, Sullivan allegedly took deliberate steps to prevent knowledge of the breach from reaching the FTC. For example, Sullivan sought to pay the hackers off by funneling the payoff through a bug bounty program. Uber paid the hackers $100,000 in BitCoin in December 2016, despite the fact that the hackers refused to provide their true names. In addition, Sullivan sought to have the hackers sign non-disclosure agreements. The agreements contained a false representation that the hackers did not take or store any data. When an Uber employee asked Sullivan about this false promise, Sullivan insisted that the language stay in the non-disclosure agreements. Moreover, after Uber personnel were able to identify two of the individuals responsible for the breach, Sullivan arranged for the hackers to sign fresh copies of the non-disclosure agreements in their true names. The new agreements retained the false condition that no data had been obtained. Uber's new management ultimately discovered the truth and disclosed the breach publicly, and to the FTC [nearly a year later], in November 2017. Since that time, Uber has responded to additional government inquiries.*
>
> *The criminal complaint also alleges Sullivan deceived Uber's new management team about the 2016 breach. Specifically, Sullivan failed to provide the new management team with critical details about the breach. In August of 2017, Uber named a new Chief Executive Officer. In September 2017, Sullivan briefed Uber's new CEO about the 2016 incident by email. Sullivan asked his team to prepare a summary of the incident, but after he received their draft summary, he edited it. His edits removed details about the data that the hackers had taken and falsely stated that payment had been made only after the hackers had been identified.*
>
> *The two hackers identified by Uber were prosecuted in the Northern District of California. Both pleaded guilty on October 30, 2019, to computer fraud conspiracy charges and now await sentencing. The criminal complaint makes clear that "both [hackers] chose to target and successfully hack other technology companies and their users' data" after Sullivan failed to bring the Uber data breach to the attention of law enforcement.*

In other words, by not dealing with law enforcement forthrightly, the hackers, who had been identified, continued to roam free to hack and damage other companies.

So, at trial, Sullivan was found guilty of lying to authorities and obstruction of justice. The trial was a landmark case, being the first time a CSO faced criminal charges for a security breach. Though it was only indirectly about the breach itself. Joe's big mistake was his attempt to cover-up and mislead investigators that ultimately landed him in some very hot water. Interestingly, Joe was once a prosecutor in the same office that charged him. He may have thought he knew how to finesse the system. But he now faces up to 8 years in prison and half a million dollars in fines, to be determined at his upcoming sentencing hearing.

What has not been made clear in the reporting I've seen is WHY Joe did this? He was a C-level executive for a major corporation. Guys at that level aren’t pulling wires and getting their hands dirty. They attend meetings and golf. So, it was almost certainly not directly Joe’s fault that somewhere in a back room two attackers somehow crawled into Uber's network. Did he have a big hunk of Uber stock that he worried would fall in value if news of this got out? If so, perhaps he believed that he could cover it up from the top to protect Uber's market value. In any event, I imagine that he regrets that decision now.

More Cryptocurrency Chaos

I believe that this podcast's listeners would be well served for me to periodically note the ongoing chaos that exists within the cryptocurrency world. It's not my position to advise anyone of anything. But being armed with a realistic viewpoint can only be valuable.

To that end, the news is that the multi-cryptocurrency exchange platform, Binance was hacked. Binance has paused its Binance Smart Chain (BSC) blockchain bridge after a threat actor used an exploit there to generate and steal 2 million Binance Coins (BNB), currently worth around $560 million. The thieves were unable to make off with all $560 million, because Biance reacted quickly. But they still absconded with 20% of the $560 million in illegitimately created funds. So, $112 million. Not bad for a day’s work.

And still more Cryptocurrency Chaos

While we're on the subject of bad ideas, I'll note that the Zcash blockchain has been subjected to a spam attack. Spam isn't just for eMail anymore. This was done by creating bloated but cheap "shielded transactions" on the Zcash blockchain. And as a consequence of this attack, which has been underway since June, the size of the Zcash blockchain has more than tripled to over 100GB. As the Zcash blockchain has grown huge, cryptocurrency experts expect Zcash node servers, which must retain a full local copy of the entire blockchain, to start failing due to memory shortage.

ALL of this points to an extremely immature technology and a gold rush attitude. Recall that in the actual California Gold Rush, 1848 - 1855, with few exceptions, the only people who made money were those selling the gold digging, panning and sluicing supplies to the hopeful miners.

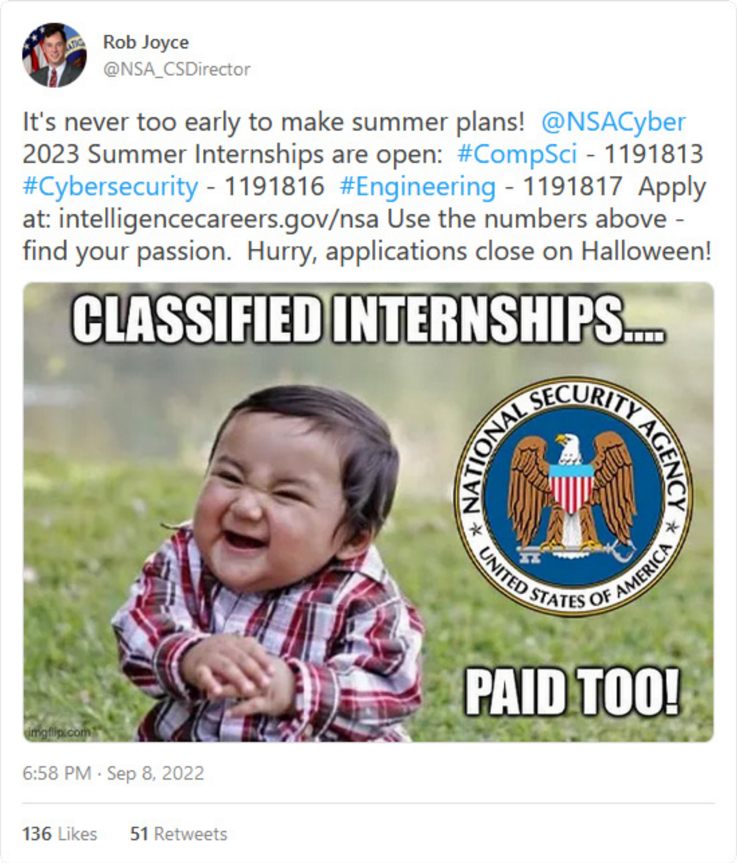
The UK to drop GDPR?

I'm not sure whether this is good or bad – though I’m leaning heavily toward bad. But last Monday, Michelle Donelan, the UK Secretary of State for Digital, Culture, Media and Sport, who was appointed to the position about a month ago, announced plans for the UK to drop the EU's GDPR in favor of designing their own new data protection system. [This is the point where I started to groan.] Michelle was speaking at the Conservative Party Conference in Birmingham where she said that the UK government will look to pass new legislation inspired by data protection laws used in Israel, Japan, South Korea, Canada, and New Zealand.

On the one hand that sounds maybe better than the GDPR, but the concern is that we only have the one single global Internet—that was the whole point of the Internet in the first place. That's what makes it so useful and amazing. But now governments are getting into the act of deciding how the Internet should uniquely treat their own precious citizens, even if that differs from how the Internet treats everyone else. Governments want to have borders, but the Internet was designed to ignore them.

A Summer Internship with the NSA?

A Summer Internship with the NSA
Rob Joyce is the Director of Cybersecurity at @NSAgov. He recently tweeted that the NSA is looking for next summer interns. He wrote: It's never too early to make summer plans! @NSACyber 2023 Summer Internships are open: #CompSci - 1191813 #Cybersecurity - 1191816 #Engineering - 1191817 Apply at: intelligencecareers.gov/nsa Use the numbers above - find your passion. Hurry, applications close on Halloween!



Rob Joyce
@NSA_CSDirector

It's never too early to make summer plans! @NSACyber 2023 Summer Internships are open: #CompSci - 1191813 #Cybersecurity - 1191816 #Engineering - 1191817 Apply at: intelligencecareers.gov/nsa Use the numbers above - find your passion. Hurry, applications close on Halloween!

6:58 PM · Sep 8, 2022

136 Likes 51 Retweets

That would be one cool way to spend a summer!

Many Incident Responders are Stressed Out

We've talked a lot about the job opportunities available across the security industry. The opportunities are there and they show no sign of diminishing. Quite the opposite. But it can be demanding and it can interfere with other life priorities. IBM recently conducted a survey of

1100 professional cyber incident responders. Here are the 7 takeaways from that survey:

1. Cybersecurity Incident Responders said that the sense of duty to help and protect others and the businesses was by far the most influential factor attracting them to the profession. Continuous opportunity to learn and being rooted in problem solving followed as the most influential factors.

2. At the same time, “sense of responsibility toward their team/client” and “managing stakeholder expectations” were ranked as the most stressful aspects of responding to cyber incidents – around half selected these among their top 3 stressors.

3. According to 48% of respondents, the average incident response engagement is 2-4 weeks. And nearly 30% say an incident response engagement lasts more than 4 weeks on average. The overwhelming majority states that it’s not uncommon to be assigned to respond to two or more incidents that overlap.

4. The first three days of responding to an attack are seen as the most stressful. Additionally, more than a third say they are working more than 12 hours a day during the most stressful period of the engagement.

5. 81% of Cybersecurity Incident Responders think the rise of ransomware has exacerbated the stress/psychological demands required during a cybersecurity incident response.

6. 67% of Cybersecurity Incident Responders said they experience stress/anxiety in their daily lives as a result of responding to an incident.

7. Nearly 65% of Cybersecurity Incident Responders have sought mental health assistance as a result of responding to cybersecurity incidents. And to that end, the majority of respondents (84%) also say they have access to adequate mental health support resources.

I think this suggests two things: First, cyber security incident response may not be for everyone. So that should be a consideration that's kept front of mind. Second, although you'll want to be salaried, if there's any way to work in bonuses for when the job does disrupt your life, that should be a consideration, too. Being the only one left at work overnight while everyone else is home laughing and sleeping is much easier if you know that your special contribution is being valued with some additional compensation.

Microsoft's newest dual 0-day Exchange Fumbles

Speaking of being stressed out, something is going on over at Microsoft and it's not good. The topic is the status of Microsoft's mitigations for that pair of 0-day Exchange Server vulnerabilities we discussed last week. Those were the new pair discovered while being used in the wild in the networks of clients of the Vietnamese cybersecurity firm GTSC.

First, I checked to see whether patches for these two new bad problems were available. That would be the optimal answer, right? Get it fixed. But after at least a week and a half, the answer to that is no. No emergency patch for Exchange Server, yet.

Then, since there was news last week that the initial mitigations proposed by Microsoft had immediately been bypassed, as I noted last week, I went to see what Microsoft had done since then and they really appear to be chasing their tail. They updated their guidance for scripts for IIS mitigation on the 4th, 5th, 6th, 7th, and 8th. Each time they're correcting typos or making small tweaks to the script, apparently trying to get it right. Nothing about this response feels like the A team has been brought in. And then we learn that Microsoft has been aware of this problem for much longer than was previously known. They were “investigating it” after becoming aware of it back in August.

In their posting titled: “Analyzing attacks using the Exchange vulnerabilities CVE-2022-41040 and CVE-2022-41082” they wrote:

> *MSTIC observed activity related to a single activity group in August 2022 that achieved initial access and compromised Exchange servers by chaining CVE-2022-41040 and CVE-2022-41082 in a small number of targeted attacks. These attacks installed the Chopper web shell to facilitate hands-on-keyboard access, which the attackers used to perform Active Directory reconnaissance and data exfiltration.* [Oh, so apparently nothing much to worry about.] *Microsoft observed these attacks in fewer than 10 organizations globally. MSTIC assesses with medium confidence that the single activity group is likely to be a state-sponsored organization.*
>
> *Microsoft researchers were investigating these attacks to determine if there was a new exploitation vector in Exchange involved* [make that "yes!"] *when the Zero Day Initiative (ZDI) disclosed CVE-2022-41040 and CVE-2022-41082 to Microsoft Security Response Center (MSRC) in September 2022.*

“Gee, look at that. Exchange Server is being attacked. Hmmmm. What's for lunch?”

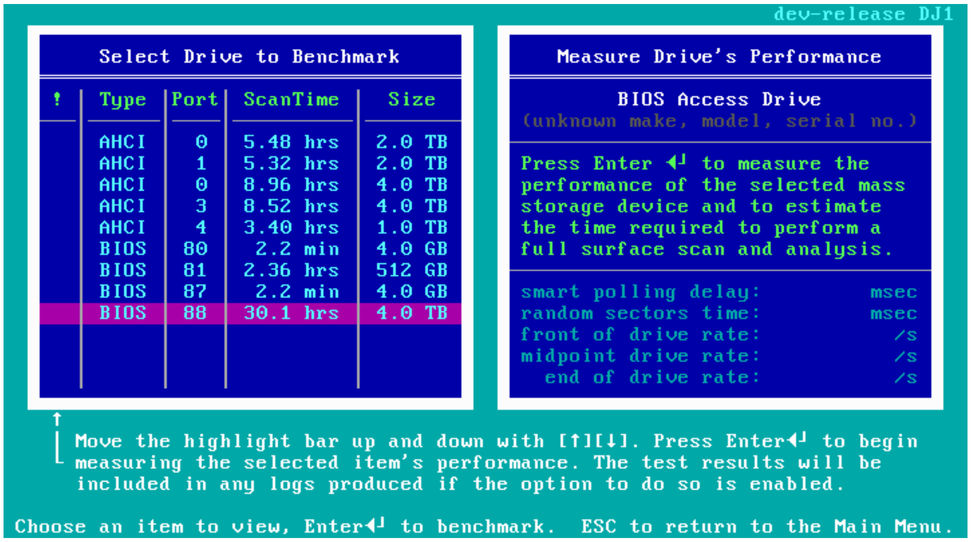
SpinRite

In SpinRite news, I’ve finished all of the redesign and SpinRite is working as far as I know. But that knowledge doesn’t yet go very far. So now I start the final work of inducing known data errors and watching SpinRite perform its sector-by-sector data recovery. That’s what I’ll be working on tonight and subsequently until I’ve demonstrated to myself that SpinRite is working. At that point I’ll release it to the GRC newsgroup gang, we’ll find the various things I’ve missed, I’ll get those fixed, and we’ll move it from Alpha to Beta state.

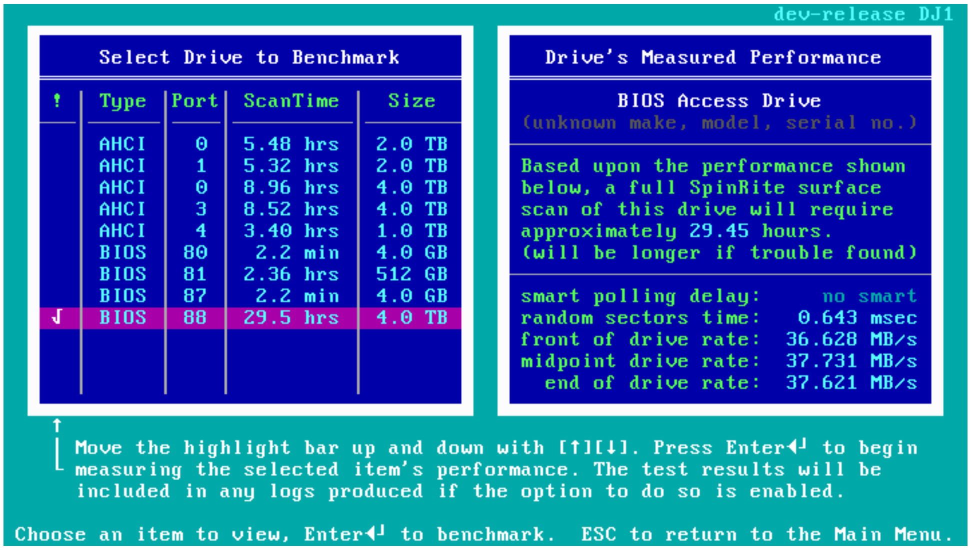
When we’re there, anyone who owns SpinRite will be able to download the DOS executable that we’ve been developing and testing. Since I won’t yet have it packaged as a turnkey Windows app, you’ll need to use GRC’s InitDisk or ReadSpeed utilities to create a BIOS bootable DOS thumb drive, then copy the SpinRite EXE to that drive, boot it and run. At that point you’ll be running the real, essentially final, SpinRite v6.1.

Something else happened last week that was interesting. Although we won’t get to high-speed native USB support until v7, probably v7.1, I designed SpinRite v6.1 to work with any size USB drive through the motherboard’s BIOS if the motherboard supports it. But it occurred to me that

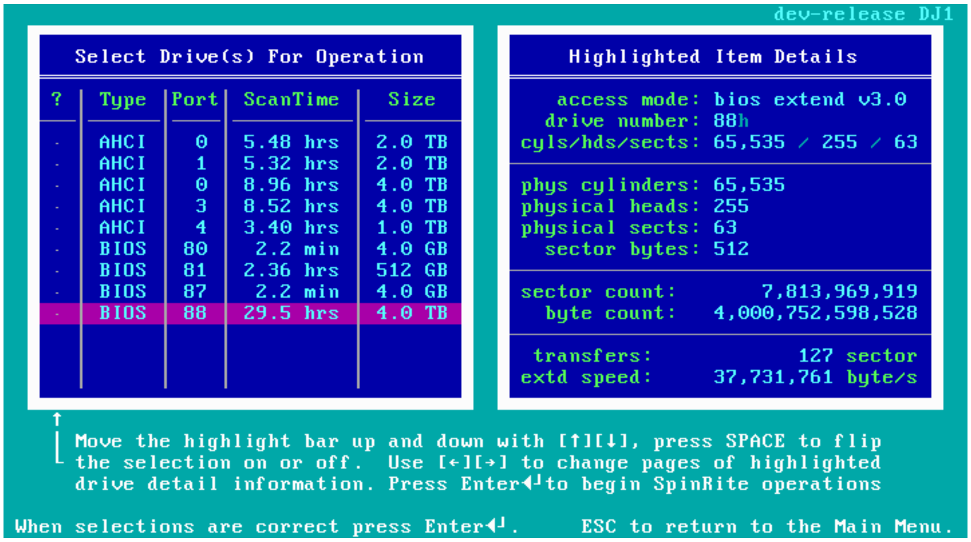
I had never explicitly asked any of our testers to try attaching a huge drive, larger than 2.2 terabytes which is the largest drive addressable with 32-bits, to a USB port to see whether SpinRite sees the large drive and can work with it. We learned that it does and it can:



```
                                                       dev-release DJ1
   Select Drive to Benchmark           Measure Drive's Performance

 ! | Type |Port| ScanTime |  Size            BIOS Access Drive
                                        (unknown make, model, serial no.)
     AHCI    0   5.48 hrs  2.0 TB
     AHCI    1   5.32 hrs  2.0 TB       Press Enter ◄┘ to measure the
     AHCI    0   8.96 hrs  4.0 TB       performance of the selected mass
     AHCI    3   8.52 hrs  4.0 TB       storage device and to estimate
     AHCI    4   3.40 hrs  1.0 TB       the time required to perform a
     BIOS   80    2.2 min  4.0 GB       full surface scan and analysis.
     BIOS   81   2.36 hrs  512 GB
     BIOS   87    2.2 min  4.0 GB       smart polling delay:        msec
     BIOS   88   30.1 hrs  4.0 TB       random sectors time:        msec
                                        front of drive rate:          /s
                                        midpoint drive rate:          /s
                                          end of drive rate:          /s

  Move the highlight bar up and down with [↑][↓]. Press Enter◄┘ to begin
  measuring the selected item's performance. The test results will be
  included in any logs produced if the option to do so is enabled.

Choose an item to view, Enter◄┘ to benchmark.  ESC to return to the Main Menu.
```

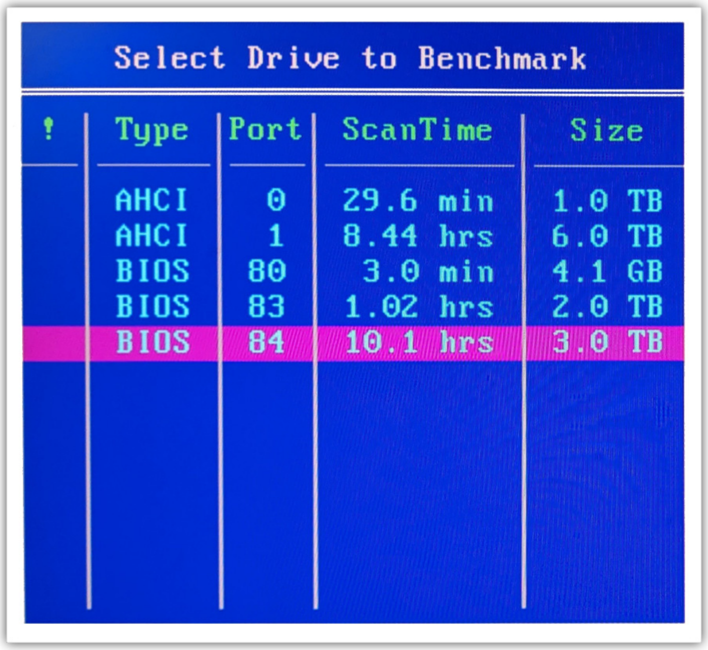


```
                                                       dev-release DJ1
   Select Drive to Benchmark           Drive's Measured Performance

 ! | Type |Port| ScanTime |  Size            BIOS Access Drive
                                        (unknown make, model, serial no.)
     AHCI    0   5.48 hrs  2.0 TB
     AHCI    1   5.32 hrs  2.0 TB       Based upon the performance shown
     AHCI    0   8.96 hrs  4.0 TB       below, a full SpinRite surface
     AHCI    3   8.52 hrs  4.0 TB       scan of this drive will require
     AHCI    4   3.40 hrs  1.0 TB       approximately 29.45 hours.
     BIOS   80    2.2 min  4.0 GB       (will be longer if trouble found)
     BIOS   81   2.36 hrs  512 GB
     BIOS   87    2.2 min  4.0 GB       smart polling delay:    no smart
 √   BIOS   88   29.5 hrs  4.0 TB       random sectors time:  0.643 msec
                                        front of drive rate: 36.628 MB/s
                                        midpoint drive rate: 37.731 MB/s
                                          end of drive rate: 37.621 MB/s

  Move the highlight bar up and down with [↑][↓]. Press Enter◄┘ to begin
  measuring the selected item's performance. The test results will be
  included in any logs produced if the option to do so is enabled.

Choose an item to view, Enter◄┘ to benchmark.  ESC to return to the Main Menu.
```



```
                                                       dev-release DJ1
  Select Drive(s) For Operation           Highlighted Item Details

 ? | Type |Port| ScanTime |  Size          access mode: bios extend v3.0
                                          drive number: 88h
 ·   AHCI    0   5.48 hrs  2.0 TB       cyls/hds/sects: 65,535 / 255 / 63
 ·   AHCI    1   5.32 hrs  2.0 TB
 ·   AHCI    0   8.96 hrs  4.0 TB       phys cylinders: 65,535
 ·   AHCI    3   8.52 hrs  4.0 TB       physical heads: 255
 ·   AHCI    4   3.40 hrs  1.0 TB       physical sects: 63
 ·   BIOS   80    2.2 min  4.0 GB         sector bytes: 512
 ·   BIOS   81   2.36 hrs  512 GB
 ·   BIOS   87    2.2 min  4.0 GB       sector count:        7,813,969,919
 ·   BIOS   88   29.5 hrs  4.0 TB         byte count:    4,000,752,598,528

                                         transfers:           127 sector
                                        extd speed:     37,731,761 byte/s

  Move the highlight bar up and down with [↑][↓], press SPACE to flip
  the selection on or off.  Use [←][→] to change pages of highlighted
  drive detail information. Press Enter◄┘to begin SpinRite operations

When selections are correct press Enter◄┘.     ESC to return to the Main Menu.
```

Here (below) is someone's screen shot who had faster BIOS support for USB drives. In this screenshot made with the v6.1 pre-Alpha, we see a 2 terabyte drive taking only one hour for a SpinRite scan (which is sort of astonishingly fast) and also a 3 terabyte drive, again demonstrating that SpinRite is now, finally, aware of drivers larger than 2.2 terabytes. In this case, SpinRite's performance measuring estimates that this 3 terabyte drive will require 10.1 hours to scan. That's not great, but it's over USB and it sure beats 10 months.



Select Drive to Benchmark

!	Type	Port	ScanTime	Size
	AHCI	0	29.6 min	1.0 TB
	AHCI	1	8.44 hrs	6.0 TB
	BIOS	80	3.0 min	4.1 GB
	BIOS	83	1.02 hrs	2.0 TB
	BIOS	84	10.1 hrs	3.0 TB

ZimaBoard — "Steve's Dream SBC"
I want to take a moment to talk about a beautiful little affordable ($120 plus shipping) single board computer that I'll be using for SpinRite's development going forward. It's called ZimaBoard and in many ways it's the perfect little platform for SpinRite. But I'll get to that in a second...

To get SpinRite to the point where it is today, which is its ability to talk directly to any and all PC hardware owned by every single one of our hundreds of SpinRite development testers — we currently have 367 registered testers in GRC's GitLab instance — I've gladly purchased innumerable old motherboards and drives from eBay. When I've been unable to duplicate some obscure problem that any of our hundreds of testers was experiencing out in the field, buying what they had was often the only way to get to the bottom of some really bizarre behavior.

But that's now behind us, at least until SpinRite starts being used by its entire owner-base. I fully expect that I'll be encountering some new mysteries. That's the nature of bypassing the BIOS. But it's clear that we've reached the 99.999% point. So it's time for the next stage.

What I wanted, going forward, was a completely silent testing platform and this little ZimaBoard looks perfect for that — no more incessant whirring fan noise while I'm trying to focus. The ZimaBoard is fanless, with a custom heatsink fin design and just the right number of ports and expandability. It started out on Kickstarter where it was 4905% over funded – in other words, more than 49 times the number of project backers than they were hoping for – people went nuts over it – and now it's a going commercial concern.

Through the years, the recurring question that we've been asked over and over is what GRC would recommend as a perfect PC platform for running SpinRite on a drive in lieu of dedicating their main machine to that task. I think this ZimaBoard is likely the ultimate answer to that question. Years ago, when I was writing the TechTalk column for InfoWorld magazine I stumbled upon a wonderful motherboard, the ultimate keyboard, an RLL controller and MFM drives that worked perfectly under RLL encoding. So I conceived of something I called "Steve's Dream Machine" and it was a hit with my column's readers. A PC supplier, NorthGate Computer Systems took up the idea of purchasing and bundling all of the components and offering them as "Steve's Dream Machine." What I think I've found here, with this ZimaBoard, is Steve's Dream SBC – Single Board Computer.

It is 100% Intel chipset with the exception of its dual Gig network adapter, which is a Realtek 8168 chip. That's perfect for my development needs, since I have DOS network drivers for that chip. It has a pair of 6 gigabit SATA 3 connectors with a cable to provide power for one drive. And it has a pair of USB 3.0 ports. So SpinRite will be able to run drives attached to either. And it has a single PCIe x 4 connector for the expansion of anything else. That could be a PCIe to IDE adapter if SpinRite needed to repair any older IDE drives, or an NVMe adapter if SpinRite needed to be run on NVMe drives once that's supported. It has built-in video through a mini DisplayPort which can do 4K video at 60hz.

And critically, the ZimaBoard offers both UEFI and traditional BIOS support. It has a very comfortable Award BIOS with all of the bells and whistles, drive boot order and so forth, so that SpinRite v6.1 will be able to boot FreeDOS and run without trouble. It could boot from an attached USB thumb drive if you wanted to leave the Debian-derived CasaOS Linux that's shipped with the board in place, or FreeDOS and SpinRite could be installed onto the board's built-in roomy 16GB eMMC drive. That's what I'll be doing. Either way, I'll be able to use the same platform for SpinRite's future development under UEFI. So it's perfect for both now and for what's next.

There are three ZimaBoard models which vary in speed and size, but the smallest of the three is what I purchased. I have two of them, one for each of my locations. As I mentioned, the smallest of the three contains a 16 gig eMMC drive which is preloaded with a Debian Linux variant which they call CasaOS. The board is broadly compatible, able to run any Intel OS, Linux, Windows, pfSense, OpenWRT, NAS software and anything else.

If you click on the "Order Now" button on the home page, and then again on the page that comes up, you'll get to the place where you set the quantity and model number you want. If you scroll down **that** 3rd page to the bottom, you'll find a "BUY ONE GET ONE FREE" offer that explains "Buy ZimaBoard and get a free 12V/3A Power Adapter!" which you'll need. So that's what I would recommend. There's also a 10% off discount coupon available but you probably cannot use both. As I mentioned, the ZimaBoard comes with cabling to supply power to a single SATA drive. But there's an optional dual SATA cabling, for $4, that you may want if you intend to power two SATA drives from the board. That's also what I'm doing.

So I now have a terrific answer to the often asked question "What does GRC recommend for running SpinRite standalone."

https://www.zimaboard.com/

Closing The Loop

ZendoDeb / @ZendoDeb

> *@SGgrc re: CAPTCHA discussion from Security Now 891:*
> *I've wondered if using Firefox makes it worse, since FF is now stove-piping cookies, especially 3rd party cookies and so when you show up at new site Google can't find a cookie*

RobinR / @robinr1981

> *Hi Steve, with all these buffer overflow and use after free issues, I've seen talk of getting development to switch to Rust. My question to you is what kind of concerns or defensive techniques do you do when developing in Assembly? Is it the fact that you are so low level you are forced to be aware of everything and thus don't fall into the same traps? Additionally, would you change anything with a piece of software that you knew would be always on and be available on the internet?*

Ben Hutton / @relequestual

> *Steve, we often hear breaches could have been avoided through the implementation of a systematic software patching and update strategy. For enterprises, there are many solutions. While performing tech support for a relative today, I found ioBit Updator (ioBit being a name I had previously trusted for the better part of a decade) was showing adverts for commercial products in the same space as notifications for software updates. Finding this unacceptable, I looked for an alternative solution. I found one, and expected to pay, but the consumer / home edition was free, and it seems like there are no limitations to speak of.*
>
> *Is there a solution you would suggest for Windows users for installing updates, free or otherwise? The solution I found looked suspicious, but had attained "leader" in Gartner's magic quadrant for "patch management" Summer 2022.*
>
> *The solution is “Patch My PC”.* [*https://patchmypc.com/*](https://patchmypc.com/)
> *Only tried it today, so not an endorsement, but seems to do the job.*

Leo???

JT Rehill / @jtrehill

> *A quick question - you or Leo mentioned in a side comment a couple episodes back that uBlock Origin can block those damn GDPR cookie pop-ups. I've tried clicking on the "block all popups" button (I use chrome btw) but that doesn't do it. Can you please tell me how to do this? Or if there is another alternative that you know of? Thanks!!*

Joel Clermont / @jclermont

> *Just listened to SN 890 about Google Analytics in the EU and thought you might be interested to learn about Fathom Analytics: usefathom.com They were designed from the ground up around privacy and designed their infrastructure to comply with GDPR, including an option to have your data never leave the EU. I have switched all my sites to it over a year ago and love it.* https://joelclermont.com/post/2020-09/why-i-switched-to-fathom-analytics/

Blaine Trimmell / @blainejt

> *You talked about the safety of "Public WIFI." But that article only talked about browser traffic. So if you are only using a web browser then yes, most likely safe. But what if you're using apps that communicate unencrypted for their work? And apps on mobile devices might be making non-TLS requests. So I would say still not safe without a VPN.*
>
> *Have to remember someone in China could hack the WIFI router in San Francisco and capture the traffic, you do not need to travel and be local.*

Bob Karon / @bobkaron

> *Hi Steve! In ref to SN891. As an IT Consultant and I never use public WIFI. Not so much from fear of hacking from someone else on the same WIFI, but from the provider of the WIFI itself. An IT person who runs it could setup a proxy or man in the middle much easier and scrape all data on it. I always tell me clients, turn the hotspot on your phone on and use that for your laptop if needed. I feel there is much less chance of Verizon trying to steal my traffic than some local coffee shop IT guy or even a big airport. Unlimited Data is very common now on cell plans anyway. Thanks for the great show for all these years! -Bob Karon*

David Lemire / @dlemire60

> *I'll trouble @SGgrc with one more CISSP CPE comment: when I was way behind on CPEs for my first year of certification, I found a blog post on the (ISC)2 website that specifically listed your podcast among a number that could count for free CPEs. Really saved my behind!*

Source Port Randomization

An unintended side effect in Linux
As we know, Internet Protocol addresses endpoints by IP address and at an IP address, a 16-bit port number identifies specific services operating at that IP address. So an end-to-end connection will have an IP address and port on one end -- the source IP and source port -- and an IP address and port on the other end -- the destination IP and port. At the receiving end where a client is connecting to a service like web, e-mail, or whatever, the port is typically well known such as 443, 25, 110, etc. And on the client's connection-initiating end, it has long been the case that when a client asks its operating system for a new outbound connection, the OS's TCP/IP network stack simply moves linearly upward starting above the reserved service port barrier at port 1024 and incrementing post numbers until some upper limit, perhaps all the way up to 65,535 before wrapping around back to the bottom and starting over.

But eleven years ago, back in 2011, having the OS allocating client connection ports, which is to say source ports, linearly was seen as a potential problem since it made the next ports to be used guessable by an adversary. And that guessability might allow adversaries to hijack connections. We know this is possible since it was precisely the lack of source port randomization that alarmed Dan Kaminski about the spoofability of DNS servers. Attackers could blindly spoof replies by guessing the linearly-allocated source ports of outstanding DNS queries.

So, in response to this perceived threat, RFC 6056 was published by the IETF, titled: “Recommendations for Transport-Protocol Port Randomization” and its abstract reads:

> *During the last few years, awareness has been raised about a number of "blind" attacks that can be performed against the Transmission Control Protocol (TCP) and similar protocols. The consequences of these attacks range from throughput reduction to broken connections or data corruption. These attacks rely on the attacker's ability to guess or know the five-tuple (Protocol, Source Address, Destination Address, Source Port, Destination Port) that identifies the transport protocol instance to be attacked. This document describes a number of simple and efficient methods for the selection of the client port number, such that the possibility of an attacker guessing the exact value is reduced. While this is not a replacement for cryptographic methods for protecting the transport-protocol instance, the aforementioned port selection algorithms provide improved security with very little effort and without any key management overhead. The algorithms described in this document are local policies that may be incrementally deployed and that do not violate the specifications of any of the transport protocols that may benefit from them, such as TCP, UDP, UDP-lite, Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP), and RTP (provided that the RTP application explicitly signals the RTP and RTCP port numbers).*

So the idea was, since the source port chosen doesn't really matter at all, there's no reason not to be a lot more clever when choosing the next one. RFC 6056 presents five different algorithms for doing just that, and it states that the so-called “Double-Hash Port Selection” algorithm offers the best trade-off. Consequently, it was recently adopted, with minor modifications, in the Linux kernel (starting with kernel version 5.12-rc1).

And this prompted a trio of industrious researchers at the Hebrew University of Jerusalem to take a look at Linux's result. What they found was not good. Their paper titled *“Device Tracking via Linux’s New TCP Source Port Selection Algorithm”* will be presented during the 32nd Usenix Security Symposium.

> *We describe a tracking technique for Linux devices, exploiting a new TCP source port generation mechanism recently introduced to the Linux kernel. This mechanism is based on an algorithm, standardized in RFC 6056, for boosting security by better randomizing port selection. Our technique detects collisions in a hash function used in the said algorithm, based on sampling TCP source ports generated in an attacker-prescribed manner. These hash collisions depend solely on a per-device key, and thus the set of collisions forms a device ID that allows tracking devices across browsers, browser privacy modes, containers, and IPv4/IPv6 networks (including some VPNs).*
>
> *It can distinguish among devices with identical hardware and software, and lasts until the device restarts. We implemented this technique and then tested it using tracking servers in two different locations and with Linux devices on various networks. We also tested it on an Android device that we patched to introduce the new port selection algorithm. The tracking technique works in real-life conditions, and we report detailed findings about it, including its dwell time, scalability, and success rate in different network types.*
>
> *We worked with the Linux kernel team to mitigate the exploit, resulting in a security patch introduced in May 2022 to the Linux kernel, and we provide recommendations for better securing the port selection algorithm in the paper.*

The principle we keep seeing, playing out over and over, is that things that once seemed to be “secure enough” — mostly because we weren't trying as hard as possible — are no longer considered to be so. The mess with modern processor micro-architectures — Spectre and Meltdown and all the rest — is a perfect example. For quite some time we were all happily living with the way our processors worked, and all of the performance those optimizations delivered. But that ended overnight when some very clever academic researchers started looking much more closely.

Another example is DRAM. Same story there. Everything seemed fine until researchers began wondering whether too many bits may have been squeezed into too small a space, and whether that might create some adjacent row interference. And sure enough...

Similarly, the issue of IP source port assignment was happily ignored. Then Dan Kaminski realized that it could be a disaster for DNS. So operating systems moved to change to ephemeral key-based pseudo-random assignment... and then **these** clever researchers said “ah, not so fast” and discovered that there's a unique per-machine pattern that can be used for tracking.

I wonder what’ll be next? Stay tuned to this podcast to find out. :)