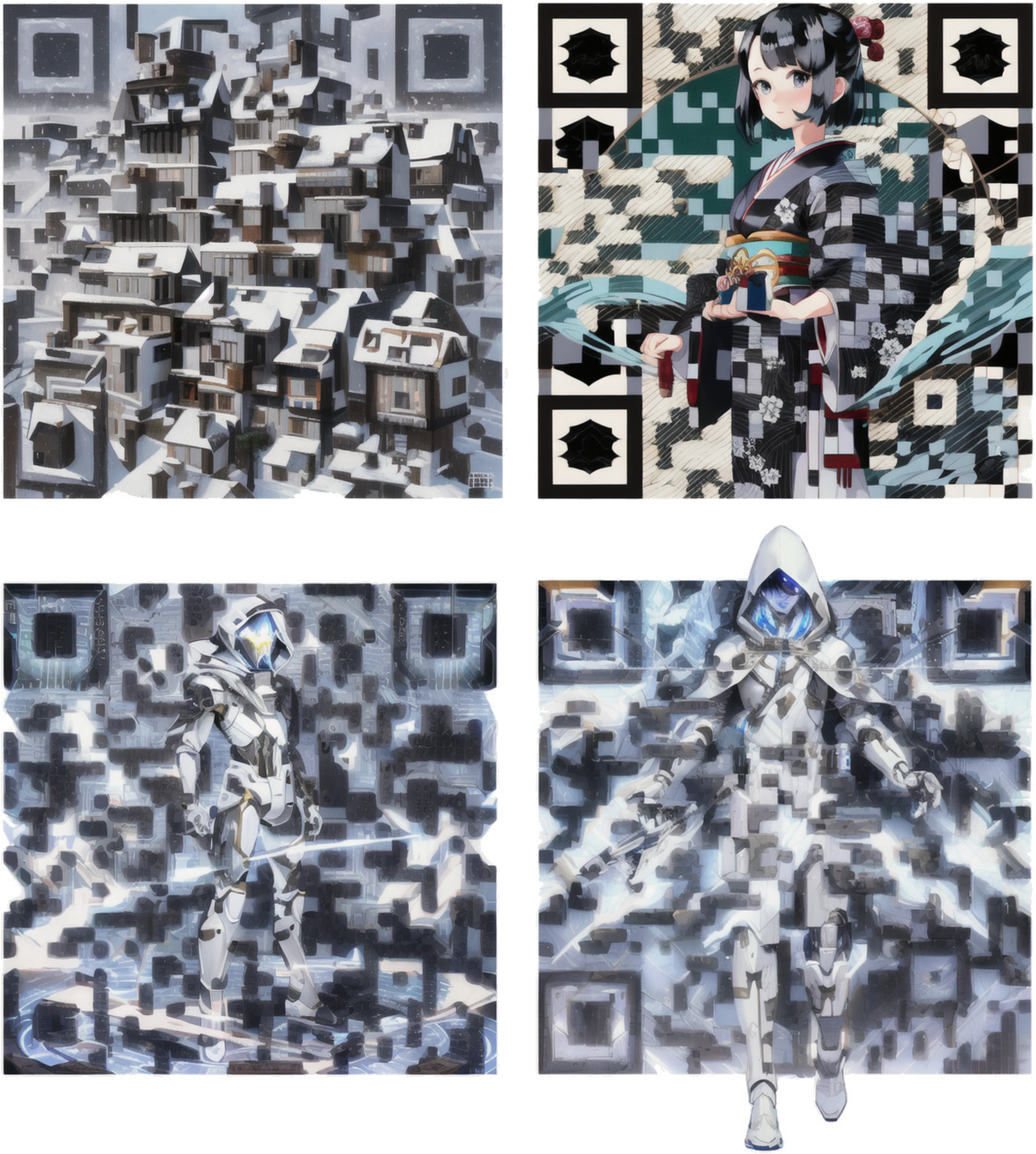
Security Now! #927 - 06-13-23
Scanning the Internet

This week on Security Now!

This week we examine what happens to your monthly cloud services bill if you're infected by cryptomining malware? And speaking of cloud services, is Elon paying his bills? Just how fast are IoT-based DDoS attacks rising? What was the strange tale of wayward Chinese certificate authority? What useful new privacy and security features will Apple be adding to their services with their net OSes this fall? And why has France headed in another direction? How does Russia feel about foreign Internet probes and what can they do about it? And after a bit of miscellany, listener feedback and a SpinRite update, we're going to take a deep dive into the backstory and current capabilities of the Internet's premiere scanning and indexing service: Censys.

Stable Diffusion meets QR Codes



https://stable-diffusion-art.com/qr-code/

"A recent Reddit post showcased a series of artistic QR codes created with Stable Diffusion. Those QR codes were generated with a custom-trained ControlNet model. Just like another day in the Stable Diffusion community, people have quickly figured out how to make QR codes with Stable Diffusion WITHOUT a custom model."

Security News

Cryptomining Rude Surprise Billing

What happens, more often than not, when a cloud computing account is compromised? The bad guys typically waste little time setting up and running a cryptocurrency mining operation. The bad news for unwitting users is that, as we know, the reason this is being done is to mine on someone else's dime. And the more computational resources that are available, the greater the rate of currency minting. Consequently, mining on stolen accounts is typically not throttled and can consume massive amounts of compute time in a short while.

And that brings us to the question: *"Who pays for that stolen compute resource usage?"* That question brings us to Google's June 8th announcement, titled: *"New Cryptomining Protection Program offers $1 million for costly cryptomining attacks."* Even though their announcement reads more like a promotional advertisement, it does contain some useful information. Here's what Google explained:

> Cryptomining is a pervasive and costly threat to cloud environments. A single attack can result in unauthorized compute costs of hundreds of thousands of dollars in just days. Furthermore, the September 2022 Threat Horizons Report published by Google's Cybersecurity Action Team revealed that 65% of compromised cloud accounts experienced cryptocurrency mining.
>
> Stopping a cryptomining attack requires effective detection, which is why we have made it a focus of Security Command Center Premium, our built-in security and risk management solution for Google Cloud. To strengthen our customers' confidence in their ability to quickly detect and stop cryptomining attacks, we are introducing a new Cryptomining Protection Program, which offers financial protection up to $1 million to cover unauthorized Google Cloud compute expenses associated with undetected cryptomining attacks for Security Command Center Premium customers.
>
> We are able to offer financial protection because Security Command Center Premium includes specialized detection capabilities that are engineered into the Google Cloud infrastructure. To detect cryptomining attacks, Security Command Center scans virtual machine memory for malware. It does this without agents, which can slow performance and increase an organization's attack surface. Our approach enables us to detect attacks that could be missed by bolt-on security tools that rely on analysis of cloud logs and information gathered from APIs.
>
> Security Command Center can also detect compromised identities, which allow attackers to gain unauthorized access to cloud accounts and quickly deploy cryptomining malware. This means Security Command Center can detect possible threats before an adversary can exploit compromised information to begin an attack. This full set of advanced detection capabilities for cryptomining can only be delivered by a product built into the cloud infrastructure.
>
> Google Cloud's shared fate approach to risk management puts our skin in the game when it comes to delivering security outcomes on our platform. By providing our customers with effective, built-in tools to detect one of the most common and costly cloud threats, we offer financial protection if our efforts are unsuccessful.

Google's article quoted Philip Bues, IDC's research manager for cloud security. He said:

> *“Cryptomining attacks continue to be a serious security and financial issue for organizations who do not have the right preventative controls and threat detection capabilities in their cloud environments. Google Cloud is taking an important step by providing built-in threat detection of unauthorized cryptomining, backed by real financial protection available to Security Command Center Premium customers, if an attacker evades their detection defenses. This shared fate approach to cloud security helps increase confidence among enterprise buyers when moving to the cloud.”*

Musk's Twitter is refusing to pay for Cloud Services

Meanwhile, since we're on the topic of Google Cloud Services billing, and even though this is a bit more gossipy than is our usual fare, but since it is potentially an intriguing event in our industry, I decided to share the news that Elon Musk's Twitter has reportedly been refusing to pay both its Google Cloud and Amazon AWS bills in an apparent strong-arm play to renegotiate its existing multi-year contracts with both service providers. This has been going on long enough, now, to lead both companies to independently begin threatening termination of services. In the case of Google Cloud, this reportedly leaves Twitter's trust and safety systems hanging in the balance as Twitter’s contract with Google Cloud Services comes up for renewal this month.

Although Twitter hosts some services on its own servers, the company has long contracted with both Google and Amazon to complement its own infrastructure, and prior to Musk's acquisition of Twitter last year, Twitter had signed an extensive multi-year contract with Google to host services related to, among other things, fighting spam, removing child sexual abuse material, and protecting its users' accounts.

But after acquiring Twitter, Musk reportedly issued a mandate to cut $1 billion dollars from Twitter's infrastructure costs. So it may be that he feels that playing hardball with Google and Amazon is the way to at least begin the process of renegotiating those agreements which predated that acquisition. We’ve previously heard stories about Twitter choosing to default on its existing lease agreements. If Twitter manages to cut $1 billion dollars from its infrastructure costs it may be by eliminating the cost of various protective services it's able to provide to its user community. Over on the Amazon side, since Twitter has also been delaying its payment for Amazon's Web Services, Amazon has reportedly been threatening to withhold its advertising payments to Twitter which would, of course, impact Twitter’s revenue. It all seems like a big mess.

IoT DDoS rapidly rising

As our listeners know, the risk posed by the rapid uptake and proliferation of today's not-yet-secure IoT devices has been a constant source and topic of concern here. Last Wednesday, June 7th, a report published by Nokia's Threat Intelligence team gave these concerns some numbers. Here's what Nokia's report explained:

> *The latest Nokia Threat Intelligence Report released today (last Wednesday) has found that IoT botnet DDoS traffic, originating from a large number of insecure IoT devices with the aim of disrupting network services for millions of users, increased fivefold over the past year, following Russia's invasion of Ukraine and stemming from the growing increase in profit-driven hacking collectives operated by cybercriminals.*
>
> *This sharp increase, also supplemented by the increased use of IoT devices by consumers around the world, was first noticed at the beginning of the Russia-Ukraine conflict but has since spread to other parts of the world, with botnet-driven DDoS attacks being used to disrupt networks as well as other critical infrastructure and services. The number of IoT devices (bots) engaged in botnet-driven DDoS attacks rose from around 200,000 a year ago to approximately 1 million devices, generating more than 40% of all DDoS traffic today.*
>
> *The most common malware was found to be a bot malware that scans for vulnerable devices, a tactic associated with a variety of IoT botnets. There are billions of IoT devices worldwide, ranging from smart refrigerators, medical sensors, and smart watches; many of which have lax security protections.*
>
> *The Threat Intelligence Report also found that the number of trojans targeting personal banking information in mobile devices has doubled to 9%, putting millions of users around the world at heightened risk of having their personal financial and credit card information stolen.*
>
> *The report, however, did find some encouraging news, showing that malware infections in home networks declined from a Covid-high of 3% to 1.5%, close to the pre-pandemic level of 1%, as malware campaigns targeting the wave of at-home workers tapered off, and more people returned to office work environments.*
>
> *These findings are based on data aggregated from monitoring network traffic on more than 200 million devices globally where Nokia NetGuard Endpoint Security product is deployed.*
>
> *Nokia's Senior Vice President for Business Applications said: "The key findings in this report underline both the scale and sophistication of cybercriminal activity today. A single botnet DDoS attack can involve hundreds of thousands of IoT devices, representing a significant threat to networks globally. To mitigate the risks, it is essential that service providers, vendors, and regulators work to develop more robust network security measures, including implementing threat detection and response, as well as robust security practices and awareness at all company levels."*

H1CA found executing code on client machines.

Okay... Get a load of this one! So a guy named Matt Holt wrote a nice little web server in the Go language. He calls it "Caddy Server" and describes it as an extensible, cross-platform, open-source web server written in Go. The name "Caddy" refers both to a helper for tedious tasks (you know, like carrying one's golf clubs), and a way to organize multiple parts into a simplified system. Okay. So we've established that Matt knows his way around web server

technology.
He was experimenting with ACME which is, as we know, the protocol created by the EFF's LetsEncrypt project to automate the issuing of TLS certificates. A low budget Chinese certificate authority named HiCA only supported one particular ACME client for its customers' servers, which Matt found odd. The client is open source and over on GitHub as ACME.sh. Here's a bit of what Matt wrote:

> *HiCA's documentation explains that it only supports acme.sh as a client. This was curious to me so I tried to learn why, if it is using ACME (and the ACME logo!) it should be basically compatible with the majority of ACME clients. While obtaining a certificate using ACMEz, I discovered that the Directory was blocked unless the User-Agent is set to a string that starts with Mozilla or acme.sh/2.8.2.*
>
> *Once I faked the User-Agent in my own client and got that working, certificate issuance still failed. Curiously, the error message involved trying a URL of ../pki-validation. This doesn't make any sense to me even though that kind of appears in their docs because it is not standard ACME, so I dug a little deeper to figure out what the Challenge object consisted of that would cause my client to try making a request to ../pki-validation.*
>
> *It turns out that the Challenge objects look unusual and it became immediately obvious to me why HiCA only supports acme.sh. They are not conforming to ACME at all! (Bugs the heck outa me that they're using the official ACME logo on their site even though they don't implement the ACME standard.)*
>
> *Instead, HiCA is stealthily crafting curl commands and piping the output to bash. acme.sh is (being tricked into?) running arbitrary code from a remote server!*

Okay, let me make that a bit more clear and fill-in some additional details:

A small Chinese certificate authority requires their clients to only run a specific acme.sh ACME client, specifically because this particular open source client has a bug which the CA has been exploiting to cause their client's web servers to remotely execute arbitrary code and commands -- on their server. Wow.

Obviously, no one should ever run code they do not trust on theit servers. And if some CA tells you that they support ACME, but only with one specific client, even if their certificates are free, run away as fast as you can.

The guy behind HiCA became involved in the GitHub thread discussion and he appears to be benign and good hearted. He explained that doing this allowed him to have more flexibility. Yeah, you bet that's flexibility if he's able to run whatever code he wants on your server. The acme.sh maintainers immediately fixed the bug that HiCA was exploiting for their service and H1CA shutdown and closed its doors.

Apple's WWDC Redux
Apple's 2023 World Wide Developer Conference is behind us and Apple did not disappoint with their continuing focus upon the privacy and security of their users. It's very clear that they intend to offer both privacy and security as features of their products and technologies.

For example, during the presentation of their new mixed-reality Vision goggle system, they made a point of noting that the system's quite powerful eye-tracking technology creates an inherent privacy risk since, as social scientists have long understood, where a user's eyes look when confronted with an image reveals, with surprisingly high fidelity, the innate emotional power of the content of various parts of an overall image. Interpreting what that means may be problematic, but it's still an unintended gateway into a user's mind. So then we ask the question: Do you want a web page you visit to know where you looked? Less privacy centric developers might think that would be quite cool, and might sell that as a feature — like having a virtual mouse pointer automatically jump to that location where a page's JavaScript is then able to obtain its coordinates and relay them back to the mothership. But that's not Apple. Apple was quite clear that where a user's eyes were looking was private information that would never leave their device. Only when they looked and clicked would the location of that click be returned.

It's this pervasive attitude across Apple that led me last week to opine that there's no way Apple has deliberately supplied anyone, including our own NSA, with a robust backdoor through iMessage to launch iDevice malware.

But back to last week's WWDC. Although this year's advancements did not explicitly focus upon user security as they did last year, Apple still demonstrated that this continues to be a selling point for them.

For their Safari browser, Apple says that they've added additional tracking and fingerprinting protections go even further to help prevent websites from using the latest techniques to track or identify a user’s device. And Safari's Private Browsing mode now locks when it's not in use to allow a user to leave private tabs open even when they've stepped away from the device. Safari will show a locked browsing window and request a Touch ID or a password to view the tabs. And Safari's Private Browsing windows now automatically lock if they have not recently been in use.

In Photos, a new embedded Photos picker can help users share specific photos with apps while keeping the rest of their library private. When apps ask to access the user’s entire photo library, the user will be shown more information about what they’ll be sharing, along with occasional reminders of their choice. And I think that is so important. This notion of occasional reminders that previous permissions remain in effect represents a significant advancement in our understanding of the human factors side of security and privacy. It's so easy for us to grant a permission in the moment when we want to make something specific happen, but to then leave that permission enabled well after it's no longer appropriate. So a gentle nudge to ask “uh... is this still want you want?” is a brilliant privacy-enhancing tactic.

So, in the case of Photos, the Photos permission prompt now tells users how many photos and videos they’d be giving access to, as well as providing a sample of those photos.

Apple is also moving to curtail the use of surreptitious link tracking in Messages, Mail, and

Safari's Private Browsing. It's becoming commonplace for websites to append extra information to their URLs as a means of tracking users across sites. We've talked years ago about how the "Referer" header informs advertisers of the URL of the site which is pulling the ad. If this URL is needlessly embellished with tracking info, that information is sent. And there's been no way to automatically limit this. Apple says that they're changing this by silently removing this unnecessary information from the links users share in Messages and Mail, and from the links in Safari Private Browsing.

And what Apple calls Communications Safety is also being further advanced. Communication Safety, which has been designed to warn children when receiving or sending photos in Messages that contain nudity, now also covers video content in addition to still images. And a new API lets developers integrate Communication Safety into their apps. This would allow these warnings to be present in non-Apple apps, too.

And this Communication Safety with now also help keep kids safe when they're sending and receiving an AirDrop, a FaceTime video message, and when using the Phone app to receive a Contact Poster and the Photos picker to choose content to send. All image and video processing for Communication Safety occurs on device, so that neither Apple nor any third party gets access to the content. And as we've talked about before, these warnings will be turned on for the child accounts in their Family Sharing plan, and can be disabled by the parent.

A Sensitive Content Warning is shown in Messages on the 12.9-inch iPad Pro.
Communication Safety now protects children receiving or attempting to send videos or photos containing nudity in AirDrop, a Contact Poster in the Phone app, a FaceTime video message, and when using the Photos picker, in addition to Messages.

These same protections are available for adult users in the form of a Sensitive Content Warning. The feature is optional and can be turned on by the user in Privacy & Security settings. And as with Communication Safety, all image and video processing for Sensitive Content Warning occurs on device, meaning neither Apple nor any third party gets access to the content.

Apple has also added Passwords and Passkey sharing with the creation of sharing groups. Users can create a group to share a set of passwords, and everyone in the group can add and edit passwords to keep them up to date. And in a slick new feature that I want to see in action, one-time verification codes received in Mail will now automatically autofill in Safari without the user leaving the browser. Huh? It sounds as though you're on a web page that sends you a link, asking you to enter it into a field which it presents. iOS observes that empty and waiting field, and also that in the background an eMail then arrives containing a code. So it parses the eMail for the code and populates the one-time-code field in the browser? Really? I cant wait to have that happen!

We've talked about Apple's “Lockdown Mode” which significantly reduces the iPhone attack surface by dramatically restricting the content that the phone will accept and process. Apple is pushing this technology even further. This was also the first time I've seen the term “mercenary spyware”, which I love. These new Lockdown protections encompass safer wireless connectivity defaults, media handling, media sharing defaults, sandboxing, and network security optimizations. So now, enabling Lockdown Mode will further harden device defenses and strictly limit functionality in the name of security. And, Lockdown Mode is now also coming to watchOS.

What Apple calls “Check In” is an interesting new feature. Here's how Apple describes it:

> *Check In makes it easy for users to let friends or family members know they’ve reached their destination safely. Once turned on by the user, Check In automatically detects when the user has reached their intended destination, and will let selected contacts know via Messages. In the case that something unexpected happens while the user is on their way, Check In will recognize that the user is not making progress toward their declared destination and check in with them. If they don’t respond, the feature will share useful information — like the user’s precise location, battery level, cell service status, and the last active time using their iPhone — with the contacts the user selected. In addition to making it easier to get help if needed, Check In is designed around privacy and security, keeping the user in control by letting them choose whom to share their information with, including the destination and time duration that they set. Users can end the Check In session at any time. Information sent with Check In is end-to-end encrypted so only the user’s family member or friend can read it, not Apple or anyone else.*

“NameDrop” allows for tightly controlled contact information sharing from device to device, presumably enabled through NFC since devices need to be in very close proximity.

And a brilliant innovation they call “Live Voicemail” allows the recipient of a phone call that they have chosen to let "go to voicemail" observe a real-time textual transcript of the voicemail as its being left, and to then change their mind on the fly and pick up the call. It's the brilliant modern equivalent of how we used to use residential telephone answering machines to screen calls and would then grab the phone receiver to pick up the phone (claiming that we had just walked in the door) when we heard who it was, or what the call was about. Now we have the same thing with our smartphones.

All of these new goodies will be arriving later this year, presumably with iOS 17.

France takes a different approach...

Not exactly following Apple's example, last Wednesday evening, June 7th, the French Senate passed an amendment to its so-called “Keeper of the Seals” justice bill. The approved changes allow law enforcement agencies to secretly activate the cameras and microphones of remote devices, and specifically including smartphones, without notifying the device's owner. Officials say they plan to use this new provision to capture sound and images of suspects of certain types of crimes. The measure would be reserved for cases of delinquency, organized crime, and terrorism. The same update to the bill text would also allow law enforcement agencies easier access to geolocation data to track criminals suspected of committing offenses punishable by at least ten years in prison.

What's not mentioned is how exactly they intend to make this actually happen in practice. I had Google translate the French new webpage, and they were saying that without this provision, investigators would need to plant physical bugs on the premises of their investigation targets. So this was being sold as a safer means for their investigators to accomplish the same already legal surveillance by turning their targets' phones or other devices into surveillance devices.

We know that Apple's iPhones will actively resist any such abuse. But one wonders whether this might be paving a legal framework for the use of, to use Apple's term “mercenary spyware” such as Pegasys which would subvert smartphone protections and would then, within the bounds of this legislation, no longer represent illegal spying which the country needs to deny,

Russia: Scanners stay out!

It should come as no surprise to anyone that Russia has decided to begin blocking foreign vulnerability scanning at the incoming border of RuNet. Very much like their continuing use of Microsoft Windows, my reaction to that is “you’re only getting around to doing that now?”

So these are services like Shodan and Censys, other security companies and proprietary operated scanners which are more or less constantly poking around the Internet to see what they can find. When some security firm notes that, for example, some new vulnerability in a Cisco device affects more than 34,000 of them, this number comes from scanning.

So it’s entirely reasonable for an increasingly hostile foreign nation not to want anyone poking around in their backyard. And wouldn’t you know it, the responsibility for limiting such scans falls to our favorite Russian Internet watchdog “Roskomnadzor”!

In their announcement of this plan, they stated that more than 10 such services are constantly scanning their Russian RuNet for vulnerable systems that are then exploited in cyberattacks. And that number of scanner sounds about right. The trouble is, to at least some degree, the scanners you know about are not the scanners you need to worry about. Shodan and Censys operate above board and scan from publicly known blocks of IP space. So blocking them, if one wanted to, would not be difficult.

But as anyone knows, who has ever tried logging all of the individual IP packet traffic arriving at any arbitrary IPv4 address, today there’s a more or less continual flux of incoming noise. And our long time listeners know that I coined the phrase “Internet Background Radiation” to remind us of that. My point is, all of the IP space of Russia’s RuNet is also constantly receiving this random noise. And it doesn’t make sense to block it all, even if you could. There’s no way for any central authority to know which traffic to which port is part of the services being offered there. Look what a mess some cable providers make when they device to block some ports that they don’t think their subscribers should be using.

So my point is, if some of those random-seeming packets were actually carefully aimed NSA probes, Russia would never be the wiser. The packets they DO need to worry about would never be the ones belonging to well known public scanning services.

Miscellany

Tavis Ormandy / @taviso (Friday, June 9th)

> *Quick personal update, it's nearly 10 years since @scarybeasts and I started Project Zero! A lot has changed since then, and I've decided there are teams where I can have a bigger impact.*
>
> *I'm still at Google, and still working on vulnerability research! I'm going to work on CPU security with Google ISE (Independent Security Evaluators). We've already got 🔥🔥🔥 0day, reports are on the way 😎*

Closing the Loop

Dave Johnston / @TechNerdDave

> *Hey Steve! Big fan of the show. I heard your stat in Ep 926 about school districts lacking security staff. Having worked in K-12 and Comm Colleges, I have some background. Many school districts are small, i.e. 1 or 2 schools. They might have 1 or 2 technology staff running the whole show. My last job was IT Director for a district with 7000 students, 14 schools which had 2 desktop techs, server admin, network admin, database admin, secretary and a director. That's not an unusual load. It's not that the districts don't care about cybersecurity. They're having a tough time just keeping all the technology running on a daily basis.*

Mrlinux11 / @mrlinux11

> *Getting error going to grc.sc/926*

Andrew Drapper / @adrapper

> *While the iPhone is quite locked down, the Mac less so. iMessege accounts are synced. Why can’t these self deleting messages be captured on a Mac.*

Jared Neaves / @MV_ENVY

> *Hi Steve, hope you are well! I was wondering if you have any more book recommendations for us, I have enjoyed all of the ones I have heard so far from you but the last one I think I heard you talk about on SecurityNow was the Bobiverse series which was a while ago! Ps just re-read one of my all time favourites - The Mote in Gods Eye and its sequel the Gripping Hand - amazing to think they conceptualised smart phones and AI home assistants back in the 70's!*

I’d already forgotten about the Bobiverse novels. They were definitely fun, and I read them all. But somehow they didn’t stick with me as much as some others, both old and new, have. The series that I’ve most recently read was Scott Jucha’s (spelled Jucha) Silver Ships. It was recommended by one of our listeners...

The Silver Ships - Scott Jucha
There are 24 books in that series (20 in the main line and 4 that are an offshoot which then merges with the rest of the initial 20.) That story had some truly wonderful moments and many terrific new ideas. So I'm glad that I read all 24 of those books. There were an additional 6 books in his "Gate Ghosts" series that caused me to do something that I almost never do, which was to quit without finishing. This tendency annoys my wife because we'll start-in on some video streaming series which, after a few episodes, turns out not to be very good. She'll want to abort, but my inclination is to see it through to the end. So it's very unusual for me to quit any Sci-Fi story in the middle – though I did quit the Gate Ghosts series after two of its six books. (I also quit Apple's "Foundation" series, to Lorrie's great relief, because it was so disappointing.) But the reason I dropped out of Gate Ghosts was a clear lack of action with no sign of any impending action. It turned into a mostly political narrative about the rights of sentient AIs and cloned humans as slaves. I'm going to get bored after a while unless something blows up from time to time...

... which leads me nicely back to my absolute favorite #1 series in a long time:

The Frontiers Saga - Ryk Brown
Ryk was a bit slowed down by a heart attack he suffered at 2:30am in the morning last December 4th. Fortunately, thanks to very good EMS response, he suffered almost no lasting cardiac damage. But the cause was severe blockages in three of his coronary arteries. So he underwent triple-bypass surgery on January 16th. Happily, his recovery was complete and he picked up right with his ambitious plan for 75 full-length novels right where he left off.

While reading the Silver Ships series, I had fallen four books behind. Now I'm caught up... and if I may have been just a bit unsure after starting into this third 15-book arc, that was quickly dispelled once we got to book #3. I would **LOVE** to share a bit about what happens, but there are some fabulous surprises awaiting anyone who still has some catching up to do. I'll just say that in Ryk's work there is no lack of action and plenty of things are blowing up!

So those are the most recent two series. Earlier during this podcast I talked about the **Honor Harrington** novels by David Weber, there are 13 of those. They are wonderful and she's a great character. There's also Jack Campbell's "Lost Fleet" Series. Which is also a ton of fun.

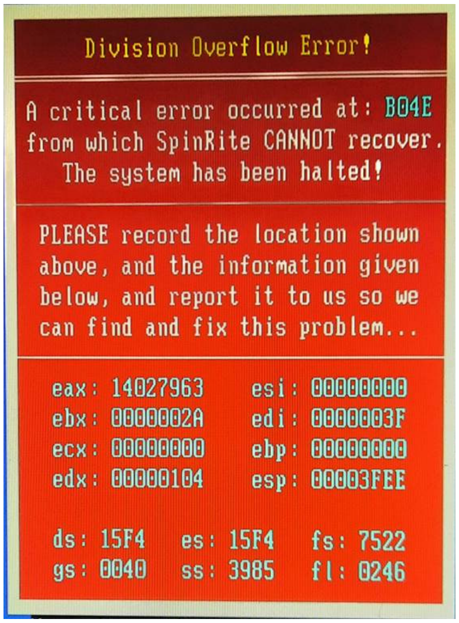
Nearly ten years ago, in a PDF dated December 19th, 2013, I captured all of my favorite reading recommendations at the time, which we had discussed here on the podcast, into a Sci-Fi Novels Guide which I just reviewed. It has the Honor Harrington and the Lost Fleet series and many others. So I've given it a shortcut using GRC's shortcut service: **https://grc.sc/scifi**. That PDF is not just a list of books, but also some commentary about each book or series to help you to decide whether or not they might be for you.

SpinRite

I don’t recall sharing any SpinRite testimonials since before I began work on SpinRite 6.1. But we received one yesterday which I thought was interesting.

Meir in Montreal wrote:

> Hello Steve & Team
>
> My son became a university lecturer and was very busy developing his course materials. After many hours of work, he discovered that the server he was using had not synced the data reliably, so he pulled out his backup 1TB USB memory stick only to discover that the computer could not even see it. Asking me for help, I pulled out my copy of spinrite 6.0 and it too did not identify the device.
>
> I then remembered that a few years ago you read a testimonial from a user who cooled his hard disk in a freezer! After 3 hours in the freezer spinrite had no problem seeing the USB device. It told me it would take 25 hours in level 2 so I left it for the night. But in the morning, I saw this:



```
Division Overflow Error!

A critical error occurred at: B04E
from which SpinRite CANNOT recover.
   The system has been halted!

 PLEASE record the location shown
 above, and the information given
 below, and report it to us so we
 can find and fix this problem...

  eax: 14027963    esi: 00000000
  ebx: 0000002A    edi: 0000003F
  ecx: 00000000    ebp: 00000000
  edx: 00000104    esp: 00003FEE

  ds: 15F4   es: 15F4   fs: 7522
  gs: 0040   ss: 3985   fl: 0246
```

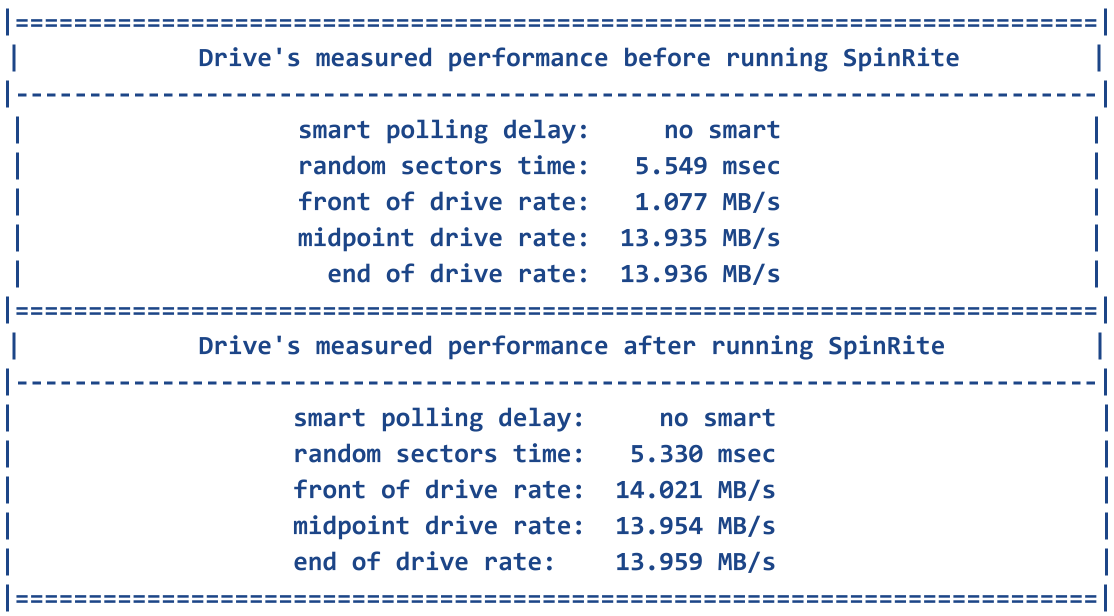
> Disappointed, I told myself “maybe it recovered some of the data.” And it did! All the 35 or so GBs of it! Freezing a USB stick? It worked. Thank you, Steve. Waiting anxiously for version 6.1! Meir in Montreal

So, first of all, the error that Meir encountered was that infamous problem in 6.0 that we now have a fix for. It’s the result of SpinRite from 2004 encountering a drive 19 years later that it was never designed to handle. Thanks to the reverse engineering work of Paul Farrer, we have a simple patch utility that can be run before SpinRite 6.0 which will prevent this from occurring.

Secondarily, it is because of recoveries like this that I decided to continue moving SpinRite forward even past v6.1. Although the days of spinning mass storage may be numbered, though that’s been predicted for quite some time, I’ve previously shared many user stories of SpinRite recovering and repairing solid state drives. I remember initially being surprised by those reports.

Now we just shrug and say “yeah, it works on those, too.” And what’s even cooler is that while not everyone always needs recovery, everyone could always use more speed...

Over the weekend, one of our v6.1 testers from Australia, a guy named Peter Hancock, ran SpinRite on one of his thumb drives. I have a clip from the SpinRite log he posted in GitLab:



```
|=========================================================================|
|            Drive's measured performance before running SpinRite         |
|-------------------------------------------------------------------------|
|                   smart polling delay:     no smart                     |
|                   random sectors time:   5.549 msec                     |
|                   front of drive rate:   1.077 MB/s                     |
|                   midpoint drive rate:  13.935 MB/s                     |
|                     end of drive rate:  13.936 MB/s                     |
|=========================================================================|
|            Drive's measured performance after running SpinRite          |
|-------------------------------------------------------------------------|
|                   smart polling delay:     no smart                     |
|                   random sectors time:   5.330 msec                     |
|                   front of drive rate:  14.021 MB/s                     |
|                   midpoint drive rate:  13.954 MB/s                     |
|                   end of drive rate:    13.959 MB/s                     |
|=========================================================================|
```

Among SpinRite 6.1’s many new features is the ability to have it automatically run pre- and post-benchmarks on the drive. Although that option is not enabled by default, I wanted to provide the option since it helps SpinRite’s users to recognize that SpinRite is really not only about data recovery – this is a clearly valuable maintenance aspect of the product.

Look at the read transfer rate at the front of Peter’s drive. Before running SpinRite on it, it was reading at 1.077 megabytes per second. After running SpinRite, it jumped up to 14.021 megabytes per second. A bit more than 13 times faster.

And this is not some distant future SpinRite X. This is the free upgrade that everyone’s getting.

And as for putting a thumbdrive in the freezer, that’s interesting, too. Meir in Montreal may have discovered something that will turn out to be useful for recovery!

Scanning the Internet

With the news that Russia wants to block scanning of its internal networks — and who could blame them for that? — I thought it would be interesting to take a look at a modern state of the art Internet scanning service to see where that state of the art is today. How quickly can the entire Internet be scanned? What ports are checked? What controls are available to be placed on such scanning. Can Roskomnadzor ask them to please not scan Russia?

The story of the Internet's current state of the art scanning begins ten years ago, in 2013, when a Turkish-American Ph.D. student at the University of Michigan named Zakir Durumeric looks at the existing and quite famous, though also quite an old kludge, NMAP network scanner and thinks (quite correctly) *"this could be done much better."* Four years later, after being one of 14 students to receive a Google PhD Fellowship in Security for the 2014-15 academic year, Doctor Durumeric finishes his Ph.D. thesis titled: *"Fast Internet-Wide Scanning: A New Security Perspective"*. The result of this work is a new scanner which Zakir names "ZMAP".

Having created GRC's ShieldsUP! service back in 1999, which required the creation of an IP stack from scratch for that purpose, I know my way around packets and I've seen a packet or two. So I can attest that Zakir's work is beautiful. This thesis demonstrates that he has an absolutely thorough grasp of the many various problems, asks all the right questions, performs all of the right experiments, and winds up developing extremely robust whole-Internet scanning and assessment technology. It's not really rocket science, but no one had taken the time to sit down and really do it right until he did. And he did.

In fairness to NMAP, the world had changed dramatically since NMAP was first conceived. The biggest change was to the bandwidth available to such a scanner. By the time Zakir came along, gigabit Internet connections were common and affordable. When I first created ShieldsUP!, my 1.54 megabit T1 line was the envy of my friends. Since no scanner wants to create its own bandwidth denial of service on itself, the available bandwidth dictates everything else about the system's architecture. So Zakir was able to re-conceptualize Internet-wide scanning at a time when doing so was feasible.

To gain an appreciation for the potential importance of the ability to have true near real-time visibility into the Internet, I want to share the Introduction to Chapter 7 of Zakir's 216-page PH.D. thesis. It's a topic that all long-term Security Now! listeners will be able to relate to, since it happened on our watch. Zakir's Chapter 7 is titled: "Understanding Heartbleed's Impact". He writes:

> *In March 2014, researchers found a catastrophic vulnerability in OpenSSL, the cryptographic library used to secure connections in popular server products including Apache and Nginx. While OpenSSL has had several notable security issues during its 16 year history, this flaw—the Heartbleed vulnerability—was one of the most impactful. Heartbleed allows attackers to read sensitive memory from vulnerable servers, potentially including cryptographic keys, login credentials, and other private data. Exacerbating its severity, the bug is simple to understand and exploit.*

> *In this work, we analyze the impact of the vulnerability and track the server operator community's responses. Using extensive active scanning, we assess who was vulnerable, characterizing Heartbleed's scope across popular HTTPS websites and the full IPv4 address space. We also survey the range of protocols and server products affected. We estimate that 24–55% of HTTPS servers in the Alexa Top 1 Million were initially vulnerable, including 44 of the Alexa Top 100. Two days after disclosure, we observed that 11% of HTTPS sites in the Alexa Top 1 Million remained vulnerable, as did 6% of all HTTPS servers in the public IPv4 address space. We find that vulnerable hosts were not randomly distributed, with more than 50% located in only 10 ASes that do not reflect the ASes with the most HTTPS hosts. In our scans of the IPv4 address space, we identify over 70 models of vulnerable embedded devices and software packages. We also observe that both SMTP+TLS and Tor were heavily affected; more than half of all Tor nodes were vulnerable in the days following disclosure.*
>
> *Our investigation of the operator community's response finds that within the first 24 hours, all but 5 of the Alexa Top 100 sites were patched, and within 48 hours, all of the vulnerable hosts in the top 500 were patched. While popular sites responded quickly, we observe that patching plateaued after about two weeks, and 3% of HTTPS sites in the Alexa Top 1 Million remained vulnerable almost two months after disclosure.*

Think of how valuable it is to have this sort of information in the wake of a significant Internet-wide security event like Heartbleed. And there's only one way to get it, which is to have the tools that are able to go out onto the Internet and look.

The introduction continues with a bit more:

> *In addition to patching, many sites replaced their TLS certificates due to the possibility that the private keys could have been leaked. We analyze certificate replacement and find that while many of the most popular websites reacted quickly, less than a quarter of Alexa Top 1 Million sites replaced certificates in the week following disclosure. Even more worryingly, only 10% of the sites that were vulnerable 48 hours after disclosure replaced their certificates within the next month, and of those that did, 14% neglected to change the private key, gaining no protection from certificate replacement.*
>
> *We also investigate widespread attempts to exploit Heartbleed, as seen in extensive bulk traffic traces recorded at four sites. We find no evidence of exploitation prior to the vulnerability's public disclosure, but we detect subsequent exploit attempts from almost 700 sources, beginning less than 24 hours after disclosure. Comparing attack attempts across sites, we observe that despite the large number of sources and scans, only a handful appear to reflect exhaustive Internet-wide scans.*
>
> *We draw upon these observations to discuss both what went well and what went poorly in the aftermath of Heartbleed. By better understanding the lessons of this security disaster, the technical community can respond more effectively to such events in the future.*

Chapter 4 of Zakir's thesis which I won't go into here is titled "Detecting Widespread Weak Keys in Network Devices." It's another example of how crucial having this sort of visibility can be. A new vulnerability and/or attack is discovered on some core aspect of our global Internet and we need to be able to assess its impact and to begin to know how to remediate its effects.

In an interview Zakir gave to the Turkish American Scientists & Scholars Association, Zakir was asked *"Could you describe your innovation in layman's terms, and how it relates to everyday life?"* Zakir replied:

> *The cornerstone of this research is ZMap, a tool that I introduced in 2013 that enables researchers to rapidly measure how every device connected to the public Internet is configured. ZMap reduces the time required to perform Internet-wide measurements from months to minutes---10,000 times faster than previous techniques---and allows us to reason about the devices that make up the Internet for the first time. Previously, many decisions were made anecdotally or through sampling. Now, we are able to perform comprehensive measurements, which has allowed us to uncover new types of bugs and understand some of the more complex interactions between devices at scale.*

Today, Zakir is an Assistant Professor of Computer Science at Stanford University and Chief Scientist of Censys, which is the inevitable commercial spin-off of his work. But before we get to that, let's look at the non-commercial side, which is "The ZMap Project" at https://zmap.io.

The ZMap project describes itself as:

> *... a collection of open source tools for performing large-scale studies of hosts and services on the Internet. The project was started in 2013 with the release of ZMap, a fast single-packet scanner that enabled scanning the entire public IPv4 address space on a single port in under 45 minutes. A year later, we released ZGrab, a Go application-layer scanner that works in tandem with ZMap. Since then, the team has expanded and we have built nearly a dozen open source tools and libraries for performing large-scale Internet measurements. Continued development is supported by the National Science Foundation (NSF).*

The Project has published a series of papers that describe how the suite of ZMap tools are designed:

- ZMap: Fast Internet-Wide Scanning and its Security Applications
- ZDNS: A Fast DNS Toolkit for Internet Measurement
- ZLint: Tracking Certificate Misissuance in the Wild
- LZR: Identifying Unexpected Internet Services

That last paper was delivered during the 2021 USENIX Security Symposium. The synopsis of this *"Identifying Unexpected Internet Services"* paper is has some interesting findings:

> *Internet-wide scanning is a commonly used research technique that has helped uncover real-world attacks, find cryptographic weaknesses, and understand both operator and miscreant behavior. Studies that employ scanning have largely assumed that services are hosted on their IANA-assigned ports, overlooking the study of services on unusual ports. In this work, we investigate where Internet services are deployed in practice and evaluate the security posture of services on unexpected ports. We show protocol deployment is more diffuse than previously believed and that protocols run on many additional ports beyond their*

> *primary IANA-assigned port. For example, only 3% of HTTP and 6% of TLS services run on ports 80 and 443, respectively. Services on non-standard ports are more likely to be insecure, which results in studies dramatically underestimating the security posture of Internet hosts. Building on our observations, we introduce LZR ("Laser"), a system that identifies 99% of identifiable unexpected services in five handshakes and dramatically reduces the time needed to perform application-layer scans on ports with few responsive expected services.*

(They give an example of achieving a 5500% speedup on port 27017 for MongoDB.)

Think about that! Who would have imagined that only 3% of HTTP and 6% of HTTPS run on ports 80 and 443? The rest must be scattered all over the remaining 65,534 ports.

Two papers later in The ZMap Project's list of published papers see a paper titled: *"Censys: A Search Engine Backed by Internet- Wide Scanning"* and the synopsis of this paper shows us how we move from an Internet-wide scanner to an Internet-wide search engine:

> *Fast Internet-wide scanning has opened new avenues for security research, ranging from uncovering widespread vulnerabilities in random number generators to tracking the evolving impact of Heartbleed. However, this technique still requires significant effort: even simple questions, such as, "What models of embedded devices prefer CBC ciphers?", require developing an application scanner, manually identifying and tagging devices, negotiating with network administrators, and responding to abuse complaints. In this paper, we introduce Censys, a public search engine and data processing facility backed by data collected from ongoing Internet-wide scans. Designed to help researchers answer security-related questions, Censys supports full-text searches on protocol banners and querying a wide range of derived fields (e.g., 443.https.cipher). It can identify specific vulnerable devices and networks and generate statistical reports on broad usage patterns and trends. Censys returns these results in sub-second time, dramatically reducing the effort of understanding the hosts that comprise the Internet. We present the search engine architecture and experimentally evaluate its performance. We also explore Censys's applications and show how recent questions become simple to answer.*

And this brings us to the second part of this, which is https://censys.io/. The CENSYS mission statement reads:

> *"At Censys we believe that cybersecurity is critical to the future of our global economy. And in order to evolve cybersecurity defenses, both the public and private sector need access to best-in-class intelligence data. By arming our customers with the visibility and insights that they need to protect against critical threats, Censys provides the intelligence needed to bolster cybersecurity capabilities worldwide."*

So, finally, what does CENSYS tell us about their Internet scanning? Their timeline notes that ZMap was invented in 2013, that Censys was founded four years later in Ann Arbor, Michigan where Zakir had gone to University, and it also shows that two years later in 2019, the original ZMap scanner was replaced by their proprietary scanning technology.

That makes sense since the world is constantly changing and six years since Zakir create ZMap is nearly forever in Internet time. So, finally, here's what we know about today's state of the art Internet scanning as embodied by Censys. Under the topic of "Host Scanning Introduction" they explain:

> *Censys continually scans the entire public IPv4 address space on 3,592+ ports using automatic protocol detection to present the most accurate representation of the Internet's current state.*
>
> *Censys also leverages redirects and the Domain Name System to discover and scan (~79M) in-use IPv6 addresses.*

That's interesting, since while it's entirely possible to scan all IPv4 addresses which occupy a 32-bit address space, there's no possibility of scanning the IPv6 128-bit IP space. So it's necessary to discover, hold, and build up a sparsely-populated map over time of active IPv6 addresses. And it's also interesting that this number is still apparently as few as around 79 million. That's a lot. But it's still not close to IPv4's fully used 4.3 billion. They continue...

> *Censys scans only obtain information: Censys never attempts to log into any service, read any database, or otherwise gain authenticated access to any system.*
>
> ***Q: How Often Does Censys Scan For New Services?***
>
> *Discovery means finding a service on an IP/port that was not there last time we looked. Censys has several schedules for discovery based on our experience scanning the Internet:*
>
> - *Global Scan of Popular Ports. We scan the whole IPv4 space on 137 ports with IANA-assigned services every day.*
> - *Cloud Provider Scans. Since many cloud hosts are ephemeral, we scan the 1,440 most popular ports on Amazon, Google, and Azure hosts every day.*
> - *Global Scan of Less Popular Ports. We scan the whole IPv4 space on 3,455 additional ports on a regular basis, completing a walk every 10 days.*
> - *Global Scan of Every Other Port Number. We scan the entire IPv4 address space across ALL ports (65535) at a low background rate.*
>
> ***How Often Does Censys Refresh Data for Known Services?***
>
> *Once a service has been discovered, Censys prioritizes refreshing the information about that service to ensure it is accurate and up to date.*
>
> *Once a day, the age of each of the ~2.1 Billion services in our data set is checked. Any (unnamed) service with an observation timestamp older than 24 hours is rescanned. With this process, the average age of high-value service data is about 16 hours!*
> ***How Does Censys Scan?***

[They don't really answer their own question, saying only:] *Censys has invested time and technology into setting up multiple global perspectives and developing sophisticated scanning techniques to produce the richest, most useful data set for the security community.* [Okay. So they're being a bit coy there.]

Censys peers with and scans from five Tier-1 ISPs (NTT, Tata, Hurricane Electric, Telia, Orange) to produce nearly 99% coverage of listening hosts across the globe with enhanced protection against packet drop. The ISP that Censys scanned any given service from is recorded in the services.perspective field.

Deep Protocol Scans

On ports with IANA-assigned protocols, Censys attempts to complete a handshake with the assigned protocol (e.g., Telnet on port 23). If that fails, we try additional handshakes according to our experience with protocol and port pairings. [So, for example, of a Telnet handshake fails on port 23, they might try an SSH handshake since its foreseeable that someone might have SSH running at a Telnet port.] *On ports without an assigned service, we start by sending an HTTP request and attempt to automatically detect the protocol based on the response.*

[I got a kick out of that. Since learning that only 3% of HTTP is on port 80, and only 6% of HTTPS is on 443 – they must be somewhere else. So why not give them a try as a first guess?]

Automatic Protocol Detection

The Censys scanner analyzes every server response to identify its service, even if it's non-standard for the port, which allows us to uncover the vast majority of services in unexpected places. For example, if an HTTP request results in an SSH banner, Censys will close the HTTP connection and reattempt an SSH handshake. Censys can detect 25 protocols on any port.

Lightweight Protocol Scans

Some protocols do not have a lot of data to parse and index. [Meaning, if an initial TCP connection handshake succeeds on some random unassigned port... now what?] *Censys identifies 47 lightweight services and collects a banner.*

What Protocols (Services) Does Censys Detect?

Censys can detect and complete scans for over 100 Layer 7 protocols. The default Layer 4 protocol used by our scanners is TCP, although some protocols, such as DNS, are scanned with UDP, and HTTP can be detected over QUIC.

Service names represent the most specific service information we have. For example, a generic HTTP service has a service name of HTTP, while an HTTP service that's actually an

Elasticsearch server has a service name of ELASTICSEARCH.

There is also an UNKNOWN fallback, which means that Censys could not identify the protocol in use by an open service, either because the service is not adhering to a protocol (there are a lot of HTTP-like services in there) or because Censys does not have a protocol-specific scanner written.

So, once a day around 2.1 billion individual Internet services which they maintain in their data set is checked and they maintain a searchable index of everything that they have found.

And what if someone like Russia doesn't want to be scanned and indexed? About this Censys says: *"Censys strives to be a good citizen of the security industry. We never attempt to log in to any service, read any database, or otherwise gain authenticated access to any system."*

Can I opt out of Censys data collection?

Censys scans help the scientific community accurately study the Internet. The data Censys gathers is sometimes used to detect security problems and to inform operators of vulnerable systems so that they can be fixed. If you opt out of data collection, you might not receive these important security notifications.

If you wish to opt out, you can configure your firewall to drop traffic from the subnets we use for scanning:

- *162.142.125.0/24*
- *167.94.138.0/24*
- *167.94.145.0/24*
- *167.94.146.0/24*
- *167.248.133.0/24*
- *2602:80d:1000:b0cc:e::/80*
- *2620:96:e000:b0cc:e::/80*

Additionally, our HTTP-based scans use a Censys-specific user agent, which can be used to filter requests from our scanners.

Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)

Configuring your services to drop connections from Censys' subnets will prevent our scanners from indexing your services. Historical data is not removed from Censys data sets as part of this change. Host services are typically pruned from Censys Search within 24-48 hours of their last observation timestamp, while Virtual Host services can remain in the data set for up to 30 days.

And finally, they explain:

> *Censys started as a research project at the University of Michigan, and we continue to provide free Internet data to the research community. We provide verified researchers the same access to our data as our highest-tiered commercial customers.*

https://support.censys.io/hc/en-us/articles/360038761891-Research-Access-to-Censys-Data

So while they do sell access to their databases and datasets to commercial entities, which they make available through an API, they also make this access available to pretty much anyone who has a justifiable use for such access. A university affiliation or published papers are not needed. You'll just need to explain what it is that you want to do and why it will be good for the Internet and society... and you'll be granted access.

I've been noticing that we've been running across this group more and more, with their name being cited by other security researchers. So I've been wanting to do a bit of a deep dive into who they are and where they came from. Now we all know.

The next time I refer to them it won't be *"who?"* it'll be... *"Oh yeah, those guys. They're good."* I believe they are.