



## How ESXi Fell

**Description:** Leo used to say at the top of our Q&A episodes: "You have questions, we have answers." Now we tease most of the questions and provide their answers. This week we wonder: What is about to happen with the EU's legislation to monitor its citizens' communications? Why would a French psychotherapy clinic be keeping 30,000 old patient records online, and who stole them? What top-level domains insist upon, and enforce, HTTPS? How is Chrome's release pace about to change? When you say that Russia "shoots the messenger," is that only an expression? Were a fool and his crypto soon parted, or should that be "was"?

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-909.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-909-lq.mp3>

---

Exactly why is QNAP back in the news, and what do I really think about Synology? Would companies actually claim unreasonably low CVSS scores for their own vulnerabilities? Nooooo. What questions have our listeners been asking after all this recent talk about passwords? What's the whole unvarnished story behind this weekend's massive global attack on VMware's ESXi servers, and who's really at fault?

These questions and more will probably be answered before you fall asleep, but no guarantees.

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about, in fact at length, about the EU's new legislation to monitor citizens' communications. This is a bad one, folks.

Steve's got the details. He'll tell you why he doesn't like QNAP, but he does like Synology. If you're looking for a NAS, you want to hear that. And then a look at VMware's ESXi servers, a massive exploit that's already claimed thousands of victims, and it's just a couple of days old. All that and more coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 909 recorded Tuesday, February 7th, 2023: How ESXi Fell.

It's time for Security Now!. Yeah, you've been waiting all week for the best show on the network. Mr. Steve Gibson makes it so. Hello, Steve.

**Steve Gibson:** Yo, Leo.

**Leo:** Good to see you.

**Steve:** Good to be with you for 909, our first show of February. And of course we've got questions. Now, you used to say at the top of our Q&A episodes: "You have questions; we have answers."

**Leo:** Yes. Whatever happened to those? We stopped those.

**Steve:** Yeah, well, because now we're teasing most of the questions and then providing the answers.

**Leo:** You're asking the questions and answering them.

**Steve:** That's right. We take care of the whole job here. So this week we wonder what is about to happen with the EU's legislation to monitor its citizens' communications. Why would a French psychotherapy clinic be keeping 30,000 old patient records online, and who stole them? What top-level domains insist upon and enforce HTTPS? How is Chrome's release pace about to change? And when you say Russia "shoots the messenger," is that only an expression? Were a fool and his crypto soon parted, or should that be "was"?

Exactly why is QNAP back in the news, and what do I really think about Synology? Would companies actually claim unreasonably low CVSS scores for their own vulnerabilities? Noooo. What questions have our listeners been asking after all this recent talk about passwords? What's the whole unvarnished story behind this week's massive global attack on VMware's ESXi servers, and who's really at fault? These questions and more will probably be answered before you fall asleep, but no guarantees.

**Leo:** No guarantees. Some of them rhetorical, I might add. Great. I'm excited. It's going to be a good show. We also have a great Picture of the Week, fitting in with the usual topic of our Pictures of the Week.

**Steve:** Indeed.

**Leo:** Picture of the Week time, Mr. G.

**Steve:** So this one can - you can spend some time visually parsing this picture. It really begs many questions. So without further ado, what we have is a close-up of a chain which has been wrapped around the opening side of a fence, like to keep the fence closed. Now, and to call this a chain really doesn't do it justice. A chain is what you wear around your neck. This thing looks like it could have been the anchor for the Titanic, just in terms of the beefiness of this chain.

But what's odd is that it's actually - there's actually two pieces of chain. There's a center three links which are actually a little smaller than the main chain which goes around in order to keep this fence closed. And for reasons not at all clear, we've got, you know, your traditional Master Lock, a standard hasp-style lock that is interlinking the chain that goes around the opening to this little three-link subchain. And then there's a white nylon zip tie which is connecting the small chain to this monster chain. Or the smaller chain. They're all big chains. And so it's like, okay.

---

**Leo:** Huh?

**Steve:** So, now, and anyone who's ever like tried to manually pull one of those nylon zip ties apart knows they are really strong. In fact, I think aren't police now using them as like...

**Leo:** They use them for handcuffs, yeah, yeah, yeah.

**Steve:** Disposable handcuffs, yeah. So you're not getting out of this. But at the same time, if you had...

**Leo:** All you need is a knife or scissors.

**Steve:** ...some toenail clippers.

**Leo:** Yeah, toenail clippers work well, yeah.

**Steve:** It's like, nothing, you know. Now you're able to get in here. And Leo, it's not like you couldn't use only the big chain with the Master padlock to bridge across the last chain. That would work just fine.

**Leo:** Yeah. You don't need this little three-link chain. I don't know what that's there for.

**Steve:** Right. And no hokey white nylon zip tie to connect the two chains together.

**Leo:** Very strange.

**Steve:** So really, the more pictures of this we see, the less faith I have in humanity. And I really, you know, I would like to get the back story behind some of these. Like the one we had a couple weeks ago had been haunting me, that piece of fence across the sidewalk that had a sign on it, "Sidewalk Closed." Except there was sidewalk that was just fine on the other side. And you could go around it in either direction. It's just like, what? Who? What? Anyway.

**Leo:** There's a couple of wags in our chatroom who say, well, truthfully it'd be harder to - the Master Lock's easier to pick than the zip tie.

**Steve:** Yeah.

**Leo:** So maybe the zip tie's actually not the weakest link in this chain. Yeah, you snip right through that, yeah.

**Steve:** If you didn't have any sharp cutting tool on your person, then, yeah, that's true. Okay. So we are back to protecting the children. And I'm not making light of that at all. CSAM, as we know, Child Sexual Abuse Material, and online exploitation of children is so distasteful that it's difficult to talk about because that requires imagining something that you'd much rather not. But it's that power that gives this a bit of a Trojan horse ability to slip past our defenses, or at least past the politicians, because there's also a very valid worry surrounding this whole issue that, once we've agreed to compromise our privacy for the very best of reasons, protecting children, our government or a foreign government or law enforcement might use their then-available access to our no longer truly private communications against us.

Now, nowhere in the EU's pending legislation, pending surveillance legislation that I'll get to in a second, is there any mention of terrorists or terrorism. But it's been voiced before, and you can bet that it will come marching out again. And once everyone's communications is being screened for seductive content that might be considered "grooming," you know, photos that might be naughty, and other content that some automated bot thinks should be brought to a human's attention, then what's next? So this is, you know, this is the very definition of a slippery slope.

Document 52022PC0209 is titled "Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down rules to prevent and combat child sexual abuse." Okay. First of all, it won't prevent it; right? Nothing will. What it will do is drive that material to seek other channels. And that's not a bad thing. And I agree that it would likely combat the problem. Though, again, prevention, okay, to some degree; right? The question is, is this the best solution, and what real price are we paying to make that possible? And, of course, what could possibly go wrong?

So what is essentially happening is that the EU is taking the next step. Over and ignoring the loud and recently polled objections of 72% of European citizens, EU legislators are preparing to move their current content screening Internet communications surveillance, which until now has been voluntary, and as a consequence somewhat limited in its application, to mandatory, and therefore universal.

Okay. So now just to recap a bit about how we got to where we are now. Three years ago, in 2020, the European Commission proposed "temporary" legislation which allowed for the automated Internet communications surveillance for the purpose of screening content for CSAM (Child Sexual Abuse Material).

The following summer, on July 6th of 2021, the European Parliament adopted the legislation to allow for this voluntary screening. And as a result of this adoption, which they refer to as an ePrivacy Derogation in other words, creating a deliberate exception to ePrivacy for this purpose U.S.-based providers like Gmail, Outlook.com, and Meta's Facebook began voluntarily screening for this content on some of their platforms. Notably, however, only those very few providers did anything. The other providers of, for example, explicitly secure communications - you know, Telegram, Signal - they've not done anything.

And so last summer, on May 11th of 2022, the Commission presented a proposal to move this Internet surveillance from - this is no longer going to be temporary, and it's no longer going to be voluntary. It will be becoming mandatory for all service providers. As we noted when this was last discussed in the context of Apple's hastily abandoned proposal to provide client-local image analysis by storing the hashes of known illegal images on the user's phone, the content to be examined includes not only images, but also textual content which might be considered solicitous of minors. That's that "grooming" term. And most controversially, all of this would impact every EU citizen, regardless of whether there was any preceding suspicion of wrongdoing. Everyone's visual and textual communications would be, and apparently will soon be, surveilled.

Interestingly, the legality of this surveillance in the EU has already been challenged; and, according to a judgment by the European Court of Justice, the permanent and general automatic analysis of private communications violates fundamental rights. Nevertheless, the EU now intends to adopt such legislation. For the court to subsequently annul it can take years, by which time the mandated systems will be established and in place.

Currently, meetings and hearings are underway. They're going to be going on through the rest of the year. A parliamentary vote is being held next month in March, followed by various actions being taken throughout the rest of the year as required to move the sure passage of this legislation through a large bureaucracy. Why "sure"? After all, how does any politician defend not wishing to protect the children? I've read a great deal of this proposal, and it has clearly been written to be rigorously defensible as a child protection act, period. So how do you stand up and vote against that? It shows every indication of being adopted, with this surveillance set to become mandatory in April of next year, 2024.

So some pieces from this legislation: "By introducing an obligation for providers to detect, report, block, and remove Child Sexual Abuse Material from their services, the proposal enables improved detection, investigation, and prosecution of offenses under the Child Sexual Abuse Directive."

Another piece: "This proposal sets out targeted measures that are proportionate to the risk of misuse of a given service for online child sexual abuse and are subject to robust conditions and safeguards. It also seeks to ensure that providers can meet their responsibilities by establishing a European Centre to prevent and counter child sexual abuse" - hereinafter referred to as "EU Centre" - "to facilitate and support implementation of this regulation and thus help remove obstacles to the internal market, especially in connection with the obligations of providers under this regulation to detect online child sexual abuse, report it, and remove child sexual abuse material. In particular, the EU Centre will create, maintain, and operate databases of indicators of online child sexual abuse that providers will be required to use to comply with the detection obligations."

Okay. Why mandatory? They say: "The impact assessment shows that voluntary actions alone against online child sexual abuse have proven insufficient by virtue of their adoption by a small number of providers only, of the considerable challenges encountered in the context of public-private cooperation in this field, as well as of the difficulties faced by member states" - meaning EU member states - "in preventing the phenomenon and guaranteeing an adequate level of assistance to victims. This situation has led to the adoption of divergent sets of measures to fight online child sexual abuse in different member states. In the absence of Union action, legal fragmentation can be expected to develop further as member states introduce additional measures to address the problem at national level, creating barriers to cross-border service provision on the Digital Single Market."

And as to why they think this is a good thing? "These measures would significantly reduce the violation of victims' rights inherent in the circulation of material depicting their abuse. These obligations, in particular the requirement to detect new child sexual abuse materials and 'grooming,' would result in the identification of new victims and create a possibility for their rescue from ongoing abuse, leading to a significant positive impact on their rights and society at large.

"The provision of a clear legal basis for the mandatory detection and reporting of 'grooming' would also positively impact these rights. Increased and more effective prevention efforts will also reduce the prevalence of child sexual abuse, supporting the rights of children by preventing them from being victimized. Measures to support victims

in removing their images and videos would safeguard their rights to protection of private and family life (privacy) and of personal data."

Okay. So this is clearly something that the EU is focused upon and is committed to seeing put into action, to be in effect in the spring of next year, 2024. And apparently the EU has a legal system much like the one which has evolved, or devolved, here in the U.S., where the court system has been layered with so many checks, balances, and safeguards against misjudgments that years will then pass while challenges make their way through the courts. Meanwhile, this is mandatory starting in April.

Conspicuously missing from any of this proposed legislation is any apparent thought to how exactly this will be accomplished from a technological standpoint, which of course is what interests us. If I have an Android phone, whose job is it to watch and analyze what images my camera captures, what images my phone receives, what textual content I exchange? Is it the phone hardware provider's job? Or is it the underlying Android OS's job? Or is it the individual messaging application? It's difficult to see how Signal and Telegram are ever going to capitulate to this. And is it the possession of the content or the transmission, reception, and communication of the content? Can you record your own movies for local use, never with any intention to do anything else with them?

"The proposal establishes and funds the so-called 'EU Centre' to serve as a central clearinghouse for suspected illegal content, and providing in some fashion the samples against which material that is seen on devices, on consumer devices in the EU, is checked against. So when an EU-based provider somehow detects something which may be proscribed, the identity and the current location of the suspected perpetrator, along with the content in question, will be forwarded to the EU Centre for their analysis and further action, if any." Wow.

So as I've been saying for years, this battle over the collision of cryptography and the state's belief in its need for surveillance is going to be a mess, and it's far from over. So Leo, it moves forward.

**Leo:** It makes me really think about the long-term consequences of that. And if I were Apple or Google or Samsung, I would be fighting this tooth and nail because, in the long run, they're going to be forced to enforce it, essentially; right?

**Steve:** To compromise.

**Leo:** They're going to have to do something, yeah. And if they do, then you're going to see a migration away from their platforms to nonproprietary open platforms so that people don't have to subjugate themselves to this. So I think it hurts them badly, first because they're going to have a battle over how to enforce it. Apple's already turned on Advanced Data Protection in the U.S., which is - and here's another question.

**Steve:** And now globally. It went global a couple weeks ago.

**Leo:** Okay.

**Steve:** With iOS 16.3 it's now universal.

---

**Leo:** They'll be noncompliant in the EU. And then there's the other question is, and they haven't done this yet, but how long before they then make it illegal for me to encrypt everything? Right? Because they're going to stop the vendors. But what if I decide, well, I'm going to figure out a way that I'm going to pre-Internet - do what you call PIE, Pre Internet Encryption of everything. Am I now found guilty because I must be hiding something?

**Steve:** I know.

**Leo:** I think it pushes people into a position where they do have to now start being responsible for their own encryption. They only would choose end-to-end encrypted choices. It's going to end up driving people underground and in the dark, not just criminals, but everybody who wants privacy. I think the long-term implications of this are bad all around.

**Steve:** I know. And so from a technology standpoint we have Signal and Telegram. There's just no way that Moxie is going to compromise...

**Leo:** Right.

**Steve:** ...Signal in order to allow the - and be responsible for having a connection to the EU Centre to get a database of things it has to check its users messaging [crosstalk]...

**Leo:** Well, and that's why I'm saying...

**Steve:** This will not happen.

**Leo:** ...the burden of this is ending up on Apple and Google and Samsung because what they'll have to do is take them out of the store. They'll have to say, well, we can't have Signal in the App Store. And then we've washed our hands of it. But Signal will continue to be distributed underground. And if you are - and this is what I'm saying is ultimately, if you care about privacy, you're going to run an open platform that you control that you put your own software, you're not going to be relying on an Apple Store or an Android Store.

**Steve:** Well, it goes a little bit further, though, because Apple could be compelled to do the filtering before Signal gets it. Remember that Signal is...

**Leo:** No, no, I understand. You can't use an Apple device is what I'm saying. The burden will end up being on Apple; and Apple will, if they comply, which they probably will have to in the long run, lose customers like you and me who will say, well, I'm going to use Signal. I'm going to do encryption. And it ain't going to be on a device where I can't. So you're exactly right. That's what I'm saying. This is who should be fighting this tooth and nail right now is Apple and Google because this is going to be, not only a burden on them, but it's going to require them to reverse things they've been doing, but also it's going to lose them customers. I don't know.

Do most people care enough about this that they would actually - you said 72% of the EU is against it?

**Steve:** They're saying, yeah, we do not want this.

**Leo:** I think you can't stop encryption; right? You can only stop it on commercial platforms.

**Steve:** Mail has already escaped.

**Leo:** Yeah. So they can't stop it. They can only tell companies, Internet service providers, carriers, cell phone manufacturers to do it. So then we just say, well, I think that just creates a brisk market for...

**Steve:** Well, remember, I was all geared up to do a product called CryptoLink years ago. I saw the handwriting on the wall.

**Leo:** Yeah, you didn't want that burden.

**Steve:** It's a much slower march, but I didn't want to be in a position where governments are saying we have to have a backdoor to your secure communications.

**Leo:** Many years ago, about 20 years ago, there was a documentary which has since been suppressed about hacking in which I gave an interview. And I said really it's going to be the hackers that are the freedom fighters. They're going to be the ones who are going to be protecting us from governments and corporations who are going to want to invade our privacy, take over our lives. And that open source software and hackers, people who know how to use it, are going to be the heroes. They're going to be the heroes. It's going to be up to us to protect ourselves. I don't think we should all turn into the Unabomber. But I think we're all going to have to embrace open software because they can't stop open software.

**Steve:** No.

**Leo:** It's very, very difficult.

**Steve:** So that would mean hacking an Android device in order to sideload your own...

**Leo:** Not necessarily. There are already companies like Pine that make phones that are not Android or iOS. They run Linux.

**Steve:** Ah, okay.

**Leo:** So they're not very good. I keep buying them in hopes, and they're terrible. But this will stimulate their development. And eventually, just as you can buy a computer that, you know, you don't have to have TPM on a computer. You can buy a computer that is not...

**Steve:** Locked down.

**Leo:** Locked down, and put open stuff on it, and control it. And that's what's going to happen, I think, at least for people who care. Maybe that's [crosstalk].

**Steve:** Yeah. And obviously that is, like, yeah, a diminishing minority. I mean, maybe once upon a time Uncle Willy was asking his nephew who was the geek what was the best computer to buy and what should you do. And so maybe it'll be like, hey, I heard about governments are spying on everybody with their phone, Junior. What phone should I get? And then, you know, Junior will know because he's in college, and he's up on all this stuff.

**Leo:** Yeah. There'll be a brisk market in open hardware and software, I think. And then the sad thing is then you've completely lost control.

**Steve:** Well, yes.

**Leo:** You know, there's nothing they can do about it.

**Steve:** Yes. And it will be, as we've already seen, it'll be the bad guys that are driven to that platform. And sadly, I mean, there is a level of false positives that occur with this. There are images which someone who's sitting there clicking a button, snapping through images, the human CAPTCHA person is sitting there saying, whoa, what's that? And, you know, go question this person. I mean, it's going to be horrible if that's happening.

**Leo:** Yeah. Yeah. I've always felt like there would come a time when this stuff, this computer technology was too powerful and that governments would want to try to control it and shut it down, and that there would always be a group of us that are called hackers. But there would always be a group of us who said, no, no, we're going to keep it open, we're going to keep it ours, and we're going to keep their prying eyes out.

**Steve:** Like Neo in the Matrix.

**Leo:** Like the Matrix.

**Steve:** Yeah.

**Leo:** Wow. And they're pushing us that way; you know? It's too bad.

**Steve:** Yeah, yeah. Okay. So 30,000 patient records online. This interesting and sobering cyber-hacking news caught my eye and raised an interesting question. Okay. First I'll share the story, and then the question that it brought to mind.

The news was that French authorities have detained a 25-year-old Finnish national who is accused of hacking the Vastaamo Psychotherapy Center. For reasons we'll see, this hack of Vastaamo is considered to be one of the worst in the country's history. Okay. Now, it occurred back in 2018 and 2019, so I guess this kid was, what, 20 years old then, when he allegedly stole the personal medical records of the clinic's patients and attempted to extort the clinic. To put pressure on the company, the hacker leaked extremely sensitive client files on the dark web. When that failed, he sent emails with ransom demands to more than 30,000 of the clinic's patients, asking them each for 200 euros and threatening to publish their medical records if they did not pay up.

**Leo:** Oh, boy.

**Steve:** Uh-huh. Finnish authorities formally identified the hacker in October last year when they issued a European arrest warrant for his arrest, and they detained him last week. Okay, so this is brazen and bad; right? The hacker obtained extremely sensitive personal medical information and chose to use it to extort both the clinic and its past patients, all 30,000 of them. And it was that number of files and patient histories that raised my eyebrows, 30,000. Okay. No matter how large and busy this clinic might be, they cannot be currently treating 30,000 patients. And in fact there are 260 working days a year, five times 52. So if the clinic averaged 10 new patients per day, which seems like a high-side number, 30,000 patient records would be 11.5 years' worth of patient files at the rate of 10 per day.

I'm sure there's some requirement for retaining medical files for some length of time. HIPAA regulations have that here in the U.S. But even so, they certainly don't need to be kept in hot online storage. If it was burdensomely expensive to store all that aging data online, then it would not be stored online because it doesn't need to be. It would be spooled onto some form of offline cold storage. Still indexed and available if needed, but offline and therefore not available to remote online attackers.

This is one of the things that we're going to need to get much better at handling as a society. Excessive data retention is a problem. And it's exacerbated by the reality that storing data costs next to nothing. So why not store it on the off chance that it might be useful for something? It doesn't delete itself unless you actually create some technology so that it does, but no one seems to do that. The problem is, even if all that old data was of no use to the clinic in this instance, it was certainly useful to the hacker, who obtained a far larger pile of extortable victims as a consequence.

So it's unclear how we move past this, where we are stuck now. There needs to be some form of incentive for inducing deletion or at least for the migration of old records into offline archival storage for varying periods of time. And such records should be destroyed once their retention period has lapsed. But "should" was the strongest word I could find. I dug into medical records retention legislation and requirements. I couldn't find any clear requirement under HIPAA for mandatory deletion. It's not there. So if an organization acts irresponsibly, it's not clear whether they would be in any legal jeopardy, at least in the U.S. God help you if you're in the EU. But still. It's clearly, you know, and we've talked about data retention before. It is a real problem.

.DEV, it turns out to my surprise, is always HTTPS.

**Leo:** Hmm?

**Steve:** I know. I encountered something the other day that I didn't realize had happened. I was over at Hover registering spinrite.dev because I thought it might come in handy since I'm planning to be spending the rest of my active coding life on what promises to be a very exciting and worthwhile project. So, as I was checking out, I was presented with a pop-up confirmation notice the likes of which I had never seen. It read, and it was number three of things I had to check off, it said "TLD INFO FOR .DEV." And of course TLD stands for top-level domain.

And it says: "Registration of .dev domains is open to anyone. You should be aware that .dev is an encrypted-by-default TLD by virtue of being inscribed in the HSTS Preload list found in all modern web browsers. Websites hosted on .dev will not load unless they are served over HTTPS, i.e., have a valid SSL certificate installed." And I had to check "I have read and understand the requirements for .dev domains" in order to proceed with the purchase. Isn't that cool?

**Leo:** Yeah.

**Steve:** So \*.dev is permanently preloaded into the HTTP strict transport security, that's HSTS, list for all modern web browsers. Okay. Now before I go any further, let me quickly review HSTS. As I just said, it stands for HTTPS Strict Transport Security. "HSTS" is an HTTP Response header which web servers can send to browsers telling them to treat the site with "Strict Transport Security." This means to only use secure HTTPS TLS connections no matter what. If the browser receives a non-secured HTTP link, the HSTS status instructs the browser to automatically upgrade it without asking anybody else, to HTTPS. The header specifies a "max-age" which tells the browser how long this security upgrade directive is to remain in effect. It's also possible to add an "includeSubDomains" parameter so that everything below that root domain will also be covered.

The first time a site is accessed using HTTPS, and the site returns the Strict-Transport-Security header, the browser records and caches this information so that all future attempts to load that site using HTTP will automatically be promoted to using HTTPS instead. When the expiration time specified by the Strict-Transport-Security header elapses, the next attempt to load the site via HTTP will proceed as normal instead of automatically using HTTPS.

Whenever the Strict-Transport-Security header is delivered to the browser, however, it will update the expiration time for that site, essentially, you know, continually pushing it forward, so sites can refresh this information and prevent the timeout from expiring. Should it be necessary for some reason to disable Strict Transport Security, setting the max-age in that header to zero, over an HTTPS connection of course, will immediately expire the Strict-Transport-Security header, allowing access then via HTTP.

But all this cleverness still leaves us with one problem: What about the very first time a browser visits a site? If that visit were initiated, for example, by following an HTTP link, maybe from a malicious email, the initial connection will be insecure, in plaintext, unauthenticated, and susceptible to interception and on-the-fly modification of the traffic. Even if the web server is sending out HSTS headers, they could be stripped from the insecure connection so that the browser never receives them.

The solution to this first-contact problem is the HSTS preload list. All modern browsers carry a large list of web domains which have previously proven to be HSTS capable by offering HTTPS TLS connections, redirecting any HTTP request over to HTTPS, and

sending an HSTS Response header with an expiration time of at least a year. Those are the requirements in order to qualify for inclusion in the browser's master list.

If all of those criteria are met, the domain qualifies for permanent HSTS registration. At that point, the HSTS Preload site - you can go to [hstspreload.org](https://hstspreload.org) - can be used to submit a domain for inclusion in the global browser HSTS preload list. GRC.com has been on that list since the list's earliest days, when we first discussed this on the podcast many years ago. And once on that list, any attempt to ever connect to port 80 will be redirected by the browser. It'll just ignore that and go to port 443 for the establishment of a TLS connection.

Okay. So with that bit of a refresher, just imagine the number of domains, the dotcoms, like GRC.com is one, how many more that must be on the list with those common top-level domains, dotcoms, you know, and the others. As I said, GRC.com has always been there, but so must be an incredible number of other domains. What's so super-cool about the idea that .dev top-level domain is, by universal agreement, all HTTPS, is that it avoids any need for subdomains of .dev being on the list. Instead of needing to have a list that enumerates all of those domains, like for example Spinrite.dev, there's only one entry on the list, \*.dev.

Down at the bottom of that HSTS Preload page it talks about this. It says, under the heading "TLD Preloading," they say: "Owners of gTLDs (global top-level domains), ccTLDs, or any other public suffix domains are welcome to preload HSTS across all their registerable domains. This ensures robust security for the whole TLD, and is much simpler than preloading each individual domain." They finished: "Please contact us if you're interested or would like to learn more."

So not only is this much simpler, but it is vastly more efficient. Since pretty much now everything needs to be HTTPS these days anyway, it's such a cool idea when a new TLD is created to simply declare the entire thing as HTTPS-only and place that single entry, \*.whatever, onto the global browser preload list. So much better than needing to have every subdomain needing to do that individually. And everybody's protected, even if they don't do the whole HSTS header routine.

Okay. So I thought, what else might be on the list? I posed that question to the gang who hangs out in GRC's Security Now! newsgroup, noting that it would be possible to pull the current list from the open source Chromium repo and run a regular expression on it to extract only top-level domains. One of our very active contributors, Colby Bouma, actually he's the one who got me into GitLab and has been helping incredibly to keep our GitLab instance organized during all this SpinRite work, he stepped up, grabbed, parsed, and filtered the current Chromium HSTS file. And sure enough, the .dev domain has a great deal of company. There are presently 40, four zero, top-level domains in the global browser HSTS list, meaning that any subdomain of any of those top-level domains will only be accessible by web browsers using authenticated and encrypted TLS connections.

Okay. In alphabetical order they are Android - so in every case this is something.android; right? App, azure, bank, bing, boo, channel, chrome, dad, day, dev, eat, esq as in esquire, fly, foo, gle...

**Leo:** Who's going to register steve.foo? Same people register as steve.boo, I guess.

**Steve:** I'll bet it's taken. Gmail, google, hangout, hotmail, ing, insurance, meet, meme, microsoft, mov, new, nexus, office, page, phd, play, prof, rsvp, search, Skype, windows, xbox, YouTube, and zip.

Okay. So .dev is there, along with 39 others. We see that Google and Microsoft, who each own several of their own TLDs, have placed them on that list. And why not? As desirable as it would be to be able to place .com, .org, .net, .edu, .gov, you know, the original bunch, onto this list, or really just to abandon HTTP for user client web browsing altogether, I don't see how we're ever going to get there from here. Doing so would immediately make any HTTP-only sites inaccessible, and that's not something I can ever see happening in our lifetimes. But what I think must be happening, because, come on, foo and gle and dad?

**Leo:** These just are new; right?

**Steve:** Exactly. And that's the point. Any new registration of a TLD is probably automatically saying put us on the global HSTS list for the entire TLD. Why not? That way you're just saying to anybody who wants to set up a web server, great, love to have you. Happy to take your 14.95 per year to maintain registration for you. Oh, and by the way, you can only use - you're going to have to get a certificate. But of course that's free now, too, with Let's Encrypt and the ACME protocol. Or even I think DigiCert is now doing the same thing. So, you know, it's no longer the case that that's a problem. So, yeah, let's make it mandatory. Anyway, I just never knew that. I thought that was very cool.

And Leo, we're next going to talk about the changes Chrome is making in their release schedule. So we were just talking about the idea of staged releases of software updates to minimize the fallout from previously undetected problems. As a matter of fact, given the number of wacky problems I've been encountering with SpinRite, as our early prerelease tests find ever more bizarre machines to torture it with, I've decided that the only sane thing for me to do will be to inform everyone here who's following this podcast when and where it's available in final beta and then in final release. Anxious as I am to inform SpinRite's entire broader user community of what has grown to become a major free upgrade, I'm going to wait a while to see how much a more local larger release goes.

**Leo:** That's smart, yeah. Especially because these are the more sophisticated listeners. They're going to be the great, great people to try it out with and let you know.

**Steve:** Yes. And I can say go to the forum, and we'll be able to get online and communicate and so forth.

**Leo:** That's smart, yeah.

**Steve:** Yeah. And, you know, people have waited 18 years, they can wait another month or two. So, yeah. And apparently Google has decided to do the same with Chrome. Back a few days before Christmas, they posted the news "Change in release schedule from Chrome 110," with the subhead "From Chrome 110 an early stable version will be released to a small percentage of users." And of course, as I just said, I can relate to that.

Chrome is just about at 110. Yesterday, the Chrome beta channel was updated to 110. There are four channels which stage the progressive rollout of each new major release. The most bleeding edge is the Canary channel, followed by the Dev channel, then the Beta channel, and then finally the main release channel. So 110, where they're going to

start staggering, staging the release, just went into beta yesterday. Its next move then will be to release. And that's where the timing will be changing a bit. What Google is now explaining is that 110 will be appearing more slowly in the release channel than before.

They wrote: "We are making a change to the release schedule for Chrome. From Chrome 110, the initial release date to stable will be one week earlier. This early stable version will be released to a small percentage of users, with the majority of people getting the release a week later at the normal scheduled date. This will also be the date the new version is available from the Chrome download page. By releasing stable to a small percentage of early users, we get a chance to monitor the release before it rolls out to all of our users. If any showstopping issue is discovered, it can be addressed while the impact is relatively small." So again, if you think about the number of Chrome users there are, it's just an unimaginable number. So, yeah, I think that makes absolute sense not to have everybody having the same problem all at once in the world.

We've been tracking the gradual increase in accountability for cyber intrusions and data breaches, with more recently IT employees even increasingly being held accountable. In another bit of just surfaced news, we learn that Russia is moving forward with its own legislation to impose major fines and even prison sentences for IT administrators and their managers following major data breaches. Yes, nothing encourages the quick and full public disclosure of data breaches more than the prospect of some prison time at the other end. The idea first surfaced last May in Russia. And once this legislation is passed, the Russian government will be able to fine individuals anywhere from 300,000 to 2 million rubles. Now, of course, 300,000 rubles won't buy you very much, maybe a Russian car. That's \$4,200 equivalent, up to 2 million rubles, which is \$28,000. And/or or imprison them for up to 10 years if their companies get hacked, and user data is stolen.

Now, okay. That's brutal. I'm all for accountability. But this could well devolve into shooting the messenger rather than the source of the message. Sure, there could be misconfiguration that IT should have known better and done more to secure. But there are also plenty of zero-day vulnerabilities that no one should be held to account for, more than the original source of the vulnerability, which is where the zero-day came from in the first place. I'm not going to dwell upon this further now because this week's primary topic winds up posing some serious questions about accountability, in this case the VMware ESXi issue. But this additional news demonstrates that we are continuing to see, and not surprisingly, mounting pressure to hold someone accountable for cybersecurity incidents. And this isn't over by a long shot.

I had to shake my head at this little piece. There's a new scam that's growing in popularity in the cyber underground where there are templates for carrying it out. Generically they're known as "crypto drainers." They're custom phishing pages that entice victims into connecting their crypto wallets with an offer to mint NFTs on their behalf. And of course this is where we all collectively chant in unison, "What could possibly go wrong?" To no one's surprise, other than the hapless victims, as soon as victims attempt to mint NFTs, the crypto drainer page siphons both a user's cryptocurrency and the desired NFT into an attacker's wallet.

**Leo:** I think the name is kind of a giveaway, the Crypto Drainer.

**Steve:** Crypto drainer. Yeah, I want to sign up for the crypto drainer page.

**Leo:** Yeah. What could possibly go wrong?

**Steve:** What could possibly go wrong? According to Recorded Future, there are several crypto drainer templates currently being advertised on underground cybercrime forums, and they are growing in popularity, of course.

Okay, now, apparently it's the Bible's Proverbs 21:20 which is the original source of the expression "A fool and his money are soon parted."

**Leo:** Now, Steve, I didn't know you were so up on the Bible.

**Steve:** Oh, honey, I'll tell you, there's nothing you can't find on Google.

**Leo:** Ah, yeah, good.

**Steve:** I didn't even @chatgpt. Now, that proverb, however, speaks of wealth being capriciously spent. In this case, of course, the outcome is the same. And you've really got to wonder that there are people willing to connect their wallets to some random page on the Internet which states, you know, "We'll mint NFTs for you and auto-deposit your profits into your wallet."

**Leo:** Sure.

**Steve:** Because, you know, you can trust us and our broken English. Oh, god.

Okay. Unfortunately - and Leo, remember, Proverbs 21:20.

**Leo:** 21:20. I'll keep that in mind, yes.

**Steve:** The Taiwanese NAS (Network Attached Storage) vendor QNAP is back in the news. And, you know, with them the news is never pretty. This time, QNAP has recently patched a SQL injection vulnerability tracked as CVE-2022-27596. That's the end of the good news of this story. A week later, Censys, that's that newer IoT search engine group, Censys says that roughly 98% of the 30,000 QNAP NAS devices it currently tracks remain unpatched.

**Leo:** What?

**Steve:** Yes. Nobody patches their QNAP NASes.

**Leo:** Guess not. It's just sitting in a closet and...

**Steve:** Yeah, exactly. So 98% of 30,000 are unpatched. And, it turns out, because it's trivial to exploit, and the exploitation process does not require any authentication, Censys expects the vulnerability to be quickly abused by ransomware gangs, as has happened many times previously, like all the many times we've talked about this before. And the number of vulnerable devices could possibly be much higher since Censys said

that there are another 37,000 QNAP systems online for which it could not obtain a version number but which are also likely vulnerable as well. So maybe 98% of 67,000 QNAP devices.

Okay. And speaking of NASes, I just wanted to give a shout-out to Synology. I own one, and I've just ordered another. They're backordered right now, and I'm not surprised because, damn, they are amazing.

**Leo:** Oh, I'm glad to hear you say that, yeah.

**Steve:** I am so impressed. I had been running a pair of colocated Drobo's which were running just fine. But the oldest one of the pair, which is now more than 10 years old, started acting a little flaky, and it finally went belly-up. Since the company's, Drobo's, future is a bit uncertain, I decided to switch to Synology which I kept hearing about. And oh my god, what a fabulous experience. What I got are the DS418s. It only has four bays as opposed to the Drobo's five, but my storage needs are not excessive, and the management experience is so good. Since I have two work locations, I plan to use their integrated Synology synchronization system to have the two boxes mirror each other. And then I'll be keeping my local work synchronized locally.

Anyway, I just wanted to say, for what it's worth, just one user's experience of Synology. It's been 100% positive. And these guys, they should have the market because they've done it right. And I know you feel the same way.

**Leo:** Oh, yeah. I have three of them. I love them.

**Steve:** Yeah. To no one's surprise, after the vulnerability intelligence company VulnCheck analyzed more than 25,000 entries from the NIST vulnerability database that contained CVSS ratings from both NIST and the product vendor, VulnCheck discovered that more than half of those analyzed, 14,000 of the 25,000 vulnerabilities, had conflicting scores where the vendors and NIST had assigned different ratings for the vulnerability's severity. Imagine that. VulnCheck says that despite the large number of entries, most of these came from 39 vendors, whom they did not name, suggesting that some companies are intentionally downgrading the severity of their own vulnerabilities.

And the trouble with this is not just public relations, which of course is why they're trying to, you know, that's what's driving them to falsely claim things are less serious than they are. At the high level, the vulnerability ratings are actually being used to set patching priorities. You know, if you can't patch everything, patch the bad things. So it's natural to patch the most important problems first. So intentional vulnerability downgrading messes with the ability to do any of that correctly. And now we have some numbers, 58% of the 25,000 where there are private listings and public listings, like official listings, the 39 of the companies who are doing this are saying, eh, we don't think it's as bad as everybody else.

Okay. As a consequence I think of the fact that we've been talking about passwords a lot in the last - actually all year so far, all of the interesting questions that I ended up finding in my mailbag were about that. I have four. Simon Lock tweeted. He said: "Dear Steve. What OTP Auth" - I'm sorry. What OTP app - I already gave away the answer. "What OTP app can you recommend, or what do you use? Mostly I think for iOS, but if it also does Android, that would be nice. Cheers and thank you for a lot of great hours listening to Security Now!"

Okay. So the one I've chosen after poking around with them a bit is the iOS app OTP Auth. For those who have settled upon something else, the fact that you have settled upon anything and are therefore using one-time passcodes is far better news than which one you've settled on. I'm not saying that it matters much at all. So I'm in no way suggesting that OTP Auth, my choice, is superior to XYZ Auth. It's just the one I like. Its interface is clean, it synchronizes among all of my iDevices through iCloud, I can unlock it with my face or touch, it pastes the code to the clipboard, which makes transcribing it simpler, and I like that fact that it has a customizable widget that allows me to have a subset of the passcodes I most use appear on the iPhone's notification center for even easier access. But it's definitely iOS only, so it won't do the cross-platform deal over to Android. Oh, and it also allows encrypted backup to a documented file format. It's published by some German guy, and he feels German. I'm impressed with the app's author.

**Leo:** That's who you want to document a format, to be honest.

**Steve:** That's right.

**Leo:** It is going to be like this.

**Steve:** That's right. It's like a no-nonsense solution. It's beautiful.

**Leo:** OTP Auth. I'll have to check it out.

**Steve:** OTP Auth. I really like it. Via a DM I received: "Hi, Steve. I've been following your podcast for more than two years, and I love it. Even though I'm not a cybersecurity or even an IT professional, I have learned a lot." I think he's maybe an ophthalmologist. Anyway, based on his Twitter DM. He said: "I have a question regarding your favorite two-factor app, OTP Auth. Would you be able to explain how does the syncing via cloud work for it? I'm syncing it via iCloud, but don't necessarily see a file there. If theoretically my iCloud was compromised, would someone be able to get hold of my OTP Auth tokens and get access to all my two-factor authentication codes? Thanks in advance."

Okay. So app data stored and linked through iCloud is not like iCloud Drive with Desktop, Documents, Downloads, et cetera. iCloud Drive is an app that deliberately exposes those shared resources. By comparison, app data is registered by the app and is never seen by the user. You only get to see like how much an app is using of your iCloud space, if you go in and analyze the way memory is consumed. Essentially, apps are able to use iCloud as their own secure synchronization service, which is private within that app. And Apple does not have the keys to that app data. They only exist in the users' devices.

So I'd say that it's as unlikely as possible for iCloud app data to be compromised. But if you were really worried about it, you can flip that switch off. As you said, and I agreed, Leo, this German guy, he said, well, maybe they don't want iCloud sync, fine. Turn it off. And I'll bet you dollars to doughnuts that he deletes it from the cloud as part of that.

Mark Jones tweeted: "Steve, you continually reinforce time-based authentication and discredit the now exceedingly common SMS message as a second factor." Amen. "You've never touched an option that I'm seeing more and more. Frequently I now have services asking me to validate via their app on my mobile device. Google just asked me to check my Google app on my phone before letting me log on a Windows machine. That is after

I've set up Google Authenticator as my second factor. Apple does it, too. I've never seen an analysis of the security of this new model. What are your thoughts?"

Okay. If a giant company like Google or Apple has the luxury of requiring you to run their app on another device and to respond to its authentication prompts, then I think that's nearly as secure as a time-varying passcode. And it certainly beats the crap out of SMS, because everything does. I say that it's "nearly as secure" because really the only way to improve upon our current six-digit standard would be to increase the number of digits, and that's not necessary since the right answer changes every 30 seconds.

The seductive beauty of the time-varying code, which only requires that both ends agree on the time of day and date, is that nothing is sent to your authentication device. The system is open loop. The authenticator can be offline and without any radio, like remember those original LCD footballs that we had back before smartphones when we first, when OTP, you know, this notion of a six-digit varying code first appeared. That time-varying code, which is driven by a shared secret cryptographic key, is really the perfect solution.

The one downside with a vendor's authentication app - oh, except I should give myself a caveat there. I didn't think of it last night. And that is interception. We are seeing that second-factor authentication of this kind is being intercepted because the channel back to the server is through the web browser. So if you're not actually where you think you are, and you could be at a spoof site, the spoof site was just asked for a two-factor code. It forwards that request to you on your browser. You go to your app, give it the six-digit code, the spoof site gets it and logs in, and is doing this behind your back.

So that, you know, that is a problem with our six-digit time-varying codes, which the Apple and Google and whomever standalone authentication app doesn't have because they are talking to their app on your phone, which they've established a relationship with, and when your phone lights up, then you know it's from them. Except one downside with the vendor's authentication app which can push notification requests is notification fatigue, which we've talked about before. Attackers are refining this now to a science, timing their spoofed authentication requests for the time of day when its user would be expected to be logging into their remote services. Or sometimes just using more brute force approaches, fatiguing the user by prompting the user over and over and over until they just give up and accept the authentication request and allow the bad guys in.

So, yes, specific vendor closed-loop authentication beats SMS, as I said, because everything does. And as long as you are giving your six-digit code to the proper site, the site you think you are, and not a spoofed phishing site, then nothing beats the open loopness of a one-time passcode.

And finally, Dan Stevens. He tweeted: "Hi, Steve. In the last Security Now! Episode 908, you and Leo discuss extensively the rules for creating secure passwords in a way that can be 'reconstructed' from memory. How complicated. What if you forget what the rules are? Maybe you've said this before, but my advice would be use a password manager with a completely random master password at good length, and write it on a slip of paper and keep it somewhere accessible and safe. Refer to the slip of paper whenever you log into the password manager. And eventually, for most people, the random password will stick in muscle memory, at which point you can destroy the slip of paper for extra security. Is this not a whole lot simpler?"

**Leo:** Not simpler, but definitely better.

**Steve:** I agree completely. But there are places where a password manager cannot reach. When I'm logging into my servers, or even into my Windows desktop, I don't have access to a password manager. It's true that I could open the manager on my phone and carefully transcribe a long and complex password. But the threat model for local login to my desktop or remote login to a network service that no one at any other IP than mine can even see is different from logging into random Internet websites.

So Leo uses an approach that he likes. And I have the phonetic made-up word approach that I like. The important thing to appreciate, I think, is that there is no one right answer nor a best answer. Anyone who's been listening to this podcast will have been exposed by now to the fundamental theory of password cracking and password entropy. And we've tossed around many different systems and schemes for creating passwords. So the right answer is any answer. The key is that you've given this some thought and will arrive at an answer. And you will hopefully have arrived at a system that creates strong passwords that are also workable for you, depending upon who you are and what your goals are. And I think we are closing this topic.

**Leo:** I mean, his way is definitely better, I mean, a truly random password. The problem is I can't be getting a slip of paper out every single time. I log into my password manager all the time. I mean, it's just part of the deal. And I'm going to keep this in my wallet?

**Steve:** And you do? Because I hardly ever log into my password manager.

**Leo:** Oh, all the time. Constantly. For a variety of reasons. I mean, I'm using it on a lot of different systems.

**Steve:** Ah, okay.

**Leo:** Your password - you're using Bitwarden; right? It doesn't time out? I've set it to time out. So if I'm not using it after a period of time, it times out. As it should.

**Steve:** Yeah. And so I'm in a locked environment. No one else has access to my machine. And the machine itself has a very strong authentication system which is protecting its access.

**Leo:** They can't even get into the machine; right.

**Steve:** And the hard drive is encrypted, blah blah blah.

**Leo:** Yeah, I could probably do that with the machines. The mobile devices use biometrics, so I don't often have to enter it. But I still from time to time will have to enter it. To me it's not practical to carry a slip of paper around with my master password. I don't think it's secure either, by the way.

**Steve:** And I often have my wallet at the other side of the house, too. So, you know.

---

**Leo:** Right, yeah. So, you know, look. I have a long password. Maybe 30-some characters of completely random stuff would eventually be memorized. But in the meantime it's a pain in the butt. I feel like I've come up with a system that generates as close to a random password as you can get. I mean, it's not truly random because it's based on a phrase. But that's pretty random.

**Steve:** Yeah.

**Leo:** I'm not too worried about it.

**Steve:** Yup. Again, I think to each their own. The important thing is think about this. Well, certainly everybody listening to this podcast is not only tired of thinking about it, they're tired of hearing about it. So we're done now.

**Leo:** Enough. Now I'm trying to figure out how I can get my secret keys out of Authy so I can move them over to your choice, which I like, by the way. I just downloaded, I've been playing with it a little bit. It's good. I think you're right. I think it's a nice one. The problem with Authy, the reason I like Authy is because it backs up my secret keys to the Authy server so I can put it on multiple phones. I don't want to, you know, used to be you'd have to reset up Google Authenticator from scratch every time.

**Steve:** Yeah.

**Leo:** But your solution is a perfect intermediate. OTP Auth lets you back it up to a file in some secure place that's encrypted. And then I can download it, and unencrypt it, and then import it, and I'd be set. So I think this is - I prefer this to trusting Twilio with it. So I think I'll probably, if I can figure out how to get those OTP seeds out, I think there are ways. But we shall see. All right, Steve. Let's get to this VMware exploit here.

**Steve:** Yeah. Today's sad story involves VMware's ESXi. ESXi is VMware's hypervisor technology that allows organizations to host several virtualized computers running multiple operating systems on a single physical server. The solution's grown very popular among cloud-hosting infrastructure providers because it's one of the good ones.

If by any chance you are two years behind in patching with a publicly exposed instance of ESXi, please, we've told you about canaries. Stop listening to this podcast right now. Go patch. If you're using a cloud-hosting provider instance, you should immediately perform a proactive version check. In fact, you could use GRC's ShieldsUP! service to make sure that your port 427 is closed to the public. And if you want to watch what's sure to become a honeypot feeding frenzy, place an instance of the OpenSLP service on port 427, stand back, and get ready.

What's going on is that over this past weekend, just two days ago, a new ransomware strain being tracked as "ESXiArgs," and we'll explain the name in a minute, swept through and encrypted several thousand unpatched VMware ESXi servers. And here's the heartbreaking bit. The entry point to all of these systems was an unpatched vulnerability more than two years old, well known, long since having been identified, being tracked as

CVE-2021-21974, for which, as we'll see, there is also a publicly available proof of concept which made it easy for the bad guys to hack these VMware ESXi servers.

Okay. So we'll get back to this weekend's attack in a minute. Let's first get some perspective on all this by turning back the clock to the fall of 2020. Back on March 2nd, 2021, Lucas Leong, a researcher with Trend Micro's Zero-Day Initiative, authored a blog posting titled "Pre-Auth Remote Code Execution in VMware ESXi." And this was in March, once he was finally able to talk about this publicly, which was about six months after he first informed VMware of what he had found.

So in his posting March 2nd, Lucas wrote: "Last fall, I reported two critical-rated, pre-authentication remote code execution vulnerabilities in the VMware ESXi platform. Both of them reside within the same component, the Service Location Protocol (SLP) service. In October, VMware released a patch to address one of the vulnerabilities, but it was incomplete and could be bypassed. VMware released a second patch in November, completely addressing the use-after-free portion of these bugs. The use-after-free vulnerability was assigned CVE-2020-3992.

"After that, VMware released a third patch in February completely addressing the heap overflow portion of these bugs. The heap overflow was assigned CVE-2021-21974." That's the one that is the trouble. "This blog," he says, "takes a look at both bugs and how the heap overflow could be used for code execution. Here is a quick video demonstrating the exploit in action."

Okay. So that was what he posted March 2nd, 2021, nearly two years ago. And then his blog post proceeds to demonstrate and provide descriptions, details, and pseudocode of the critical portions of the homegrown OpenSLP server that VMware had running in their ESXi server. While continuing to be responsible, Lucas disclosed all of the juicy details a month after the trouble was finally patched. So they finally patched it in February 2021, VMware did. Lucas waited a month and then he did his blog posting. Didn't do a proof of concept publicly, but did reveal what he had found. When Lucas is describing the heap overflow bug in question - this is the one ending in 21974 - he notes, he says: "Like the previous bug, this bug exists only in VMware's implementation of SLP."

As I noted, the balance of his posting provides pseudo code of VMware's code and walks the reader step by step through a theoretical exploitation process. Lucas implemented it as shown in the video; but being responsible, he deliberately stopped short of providing a working proof of concept.

At the end of his step-by-step explainer he notes: "If everything goes fine, you can now execute arbitrary code with root permission on the target ESXi system." He says: "In ESXi 7, a new feature called DaemonSandboxing was prepared for SLP. It uses an AppArmor-like sandbox to isolate the SLP daemon. However, I find that this is disabled by default in my environment." And as this week's news demonstrates all too clearly, Lucas was not alone in finding that sandboxing was not present or enabled.

He concludes with: "VMware ESXi is a popular infrastructure for cloud service providers and many others. Because of its popularity, these bugs may be exploited in the wild at some point. To defend against this vulnerability, you can either apply the relevant patches or implement the workaround. You should consider applying both to ensure your systems are adequately protected. Additionally, VMware now recommends disabling the OpenSLP service in ESXi, if it is not used." So, yes, adding insult to injury, we also have the old security bugaboo of a service, which turns out to be readily exploitable, which is running by default, unbidden, even if there is no need for it in any given deployment. Yet there it is, not even a back door. This is a front door.

Now, being a responsible researcher, as I said, Lucas's job was now done. He found a problem, privately and responsibly notified its publisher, in this case discovered that it hadn't been fixed once, or twice, but finally the third attempted patch worked. So Lucas doubtless moved on to examine and improve the security of other software which would benefit from his scrutiny. But, of course, other people have other interests. Nearly three months after Lucas's posting, on May 24th, 2021, a hacker by the name of Johnny Yu extended Lucas's work, essentially pushing it across the finish line.

Johnny wrote: "During a recent engagement, I discovered a machine that's running VMware ESXi 6.7.0. Upon inspecting any known vulnerabilities associated with this version of the software, I identified it may be vulnerable to ESXi OpenSLP heap-overflow CVE-2021-21974. Through googling, I found a blog post by Lucas Leong of Trend Micro's Zero Day Initiative, the security researcher who found this bug. Lucas wrote a brief overview on how to exploit the vulnerability, but shared no reference to a proof of concept. Since I couldn't find any existing proof of concept on the Internet, I thought it would be neat to develop an exploit based on Lucas's approach. Before proceeding, I highly encourage fellow readers to review Lucas's blog to get an overview of the bug and exploitation strategy from the discoverer's perspective."

So here we have a textbook example of the way we get from "Something doesn't look right here" to "Here's how to exploit this if you ever encounter a server with it unpatched." The two-year-old vulnerability allows threat actors to execute remote commands on any unpatched ESXi server through VMware's own implementation of the OpenSLP service on port 427. What's OpenSLP? The project has its own website which describes this as: "Service Location Protocol is an Internet Engineering Task Force (IETF) standards track protocol that provides a framework to allow networking applications to discover the existence, location, and configuration of networked services in enterprise networks.

"The OpenSLP project is an effort to develop an open-source implementation of the IETF Service Location Protocol suitable for commercial and non-commercial application. While other service advertising and location methods have been invented and even widely consumed, no other system thus far has provided a feature set as complete and as important to mission-critical enterprise applications as SLP." So I've never looked at it closely. I don't know about it. It looks like, well, for some reason VMware decided they wanted to add it. They apparently rolled their own, and it had some problems.

And not only is it often unused and unneeded, but it's running by default. So until and unless patched, it offers a way for criminals. How many criminals so far? Is everybody sitting down? More than 3,200 VMware - 3,200 individual VMware ESXi servers were hacked over the weekend.

**Leo:** What? Just over the weekend?

**Steve:** Yes.

**Leo:** Oh, god.

**Steve:** First reports came in on Friday, and then they increased. 3,200. Okay. This is this ESXiArgs ransomware campaign. France is the most affected country because they have a hosting provider who unfortunately seems to really like to have old versions of ESXi for their customers. France, followed by the U.S., Germany, Canada, and the UK in declining numbers. And we have the ransom note.

The home page of the web server that ESXi publishes will say after the attack, "How to Restore Your Files," in looks like Heading H1 in HTML. "Security Alert!!! We hacked your company successfully. All files have been stolen and encrypted by us. If you want to restore files or avoid file leaks, please send 2.034413 bitcoins to the wallet" - and then a bitcoin address. "If money is received, encryption key will be available on TOX\_ID." And then they provide a public key we'll talk about in a second.

And then "Attention!!! Send money within three days, otherwise we will expose some data and raise the price. Don't try to decrypt important files. It may damage your files. Don't trust who can decrypt. They are liars. No one can decrypt without key file. If you don't send bitcoins, we will notify your customers of the data breach by email and text message, and sell your data to your opponents or criminals. Data may be made release. Note: SSH is turned on. Firewall is disabled."

So that's not a note that you want to receive coming from your server. And more than 3,200 VMware servers are now, or were, broadcasting that note. That 2.034413 bitcoins is, you know, bitcoin value fluctuates; right? It looks like at the time this happened it was about \$50,000. So they're asking for about \$50,000 per instance. The logic must be that since it was a collection of hosted servers running inside the VMware hypervisor that was taken down, not an entire enterprise, this isn't worthy of hundreds of thousands of dollars in ransom payment. And since the attackers have left more than 3,200 of these ransomware notes, they presumably expect to receive many smaller payments rather than one big score.

In the U.S., cybersecurity officials at CISA have confirmed that they're investigating the ESXiArgs campaign. A CISA spokesperson was reported saying that "CISA is working with our public and private sector partners to assess the impacts of these reported incidents and providing assistance where needed. Any organization experiencing a cybersecurity incident should immediately report it to CISA or the FBI."

Now, the standing advice, of course, is always do not pay. And in this instance that seems a little extra warranted because it turns out that the bitcoin wallet addresses appearing in the ransom demands are not 100% individualized. Wallet reuse has been detected. But still, there are a great many of them. Since the ransom note is left behind on a public-facing web server, and it always follows the same pattern, researchers have been scanning the 'Net for infected machines - that's how we have a count - and compiling lists of the bitcoin wallet addresses appearing in the ransom demands.

I have a link in the show notes to a GitHub page that's maintaining a growing list of detected addresses. And I think there were like 700-some last time I saw, but it wasn't super current. And somebody did do a sort by the address and was seeing doubling of the use. So it looks like the bad guys didn't want to create an individual bitcoin wallet for every single one of these 3,200, I mean, there's only so much time. They're so busy infecting and taking over all these VMware ESXi servers.

**Leo:** But if you pay, and they've shared the wallet, how do they know you paid?

**Steve:** Ah, precisely. There is a way, although I don't know how unique - well, there is a way. I'll explain in a second. The ransom note refers to a Tox ID. And Leo, this kind of comes back also to our conversation at the beginning about the EU's surveillance intentions. The Tox ID is shown in the demand and provides a very long hex string. Tox is an interesting, open source, end-to-end encrypted, peer-to-peer instant messaging system that uses no centralized servers. So it boasts that it cannot be shut down.

I have not examined it closely, so I can't say whether or not it could be blocked. But it's a perfect example of the trouble that the EU or any other bureaucracy is going to have when they attempt to tighten the screws on the legal and illegal use of encrypted communications. As we've always said, the math has already escaped. There are an infinite number of ways to communicate with unbreakable encryption. It's true that stomping on the mass market solutions will catch those who are unaware, but history also shows that awareness follows very quickly.

Anyway, a Tox ID is used to identify peers on the network, and the system is simplicity itself. The Tox ID is simply the 256-bit, thus 32-byte, static public key of the other peer on the network to which you wish to communicate. This means that a packet of communications can be encrypted with a random nonce. That nonce can then be encrypted using the recipient's Tox ID, that is, the recipient's Tox ID public key, and it can then be sent on its way. Only the party with the matching private key will be able to decrypt the nonce and then use that decrypted nonce to decrypt the message payload.

So a victim sends: "Hey creeps. I just paid you your \$50,000 in bitcoin. It went to the following wallet at this time of day. Please send me the decryption instructions and destroy our unencrypted virtual machines that you stole." And then of course they kneel down to pray because, you know, who knows if they're ever going to see...

**Leo:** Well, we know.

**Steve:** ...the decryption key that they think they bought.

**Leo:** You know, bad guys are honorable and can be counted on to keep their word.

**Steve:** That's right. That's right. So by far the most impacted are the customers of hosting provider OVHcloud, based in France. While it's tempting to blame them for the misery that their customers are suffering, it appears that all that the cloud service is providing are bare metal servers onto which the VMware ESXi hypervisor is installed. It's difficult to understand why such an outsized proportion, I think it's like 44% of all of the compromises is this one provider. So it's hard to understand why such an outsized proportion of impacted ESXi servers are within OVH's cloud.

It might be that OVH offers initial setup services, and that over the course of many years they set up their ESXi servers on behalf of their customers which were never then patched or upgraded. And who knows how recently? Maybe even OVH didn't bother updating beyond the 6.5, 6.7 server that has the problem. I don't have any experience with the ESXi upgrade process, but I did note that VMware's page describing the process of upgrading ESXi was last updated yesterday. So it appears that there's a sudden demand for information about how to get away from the old and buggy version 6's and the early version 7's. Patches to an existing system appear to be far more easily applied, and that would have solved the problem two years ago. But many thousands of ESXi admins never bothered.

In a statement to TechCrunch, a VMware spokesperson said the company was aware of reports - you think? - that a ransomware variant dubbed ESXiArgs, this is the spokesperson, "appears to be leveraging the vulnerability identified as CVE-2021-21974" and said that patches for the vulnerability "were made available to customers two years ago in VMware's security advisory of February 23, 2021." She goes on to add that "Security hygiene is a key component of preventing ransomware attacks, and

organizations who are running versions of ESXi impacted by CVE-2021-21974, and have not yet applied the patch, should take action as directed in the advisory."

Okay. So as we know, mistakes happen. This is all complicated stuff which we haven't yet figured out how to create securely. But as much as I have infinite understanding for mistakes, I'm unforgiving about deliberate policy decisions. Someone, somewhere, made the policy decision at VMware to have this homegrown OpenSLP server that apparently few people actually need running by default, opening port 427, then listening for and accepting incoming unsolicited connections from the public Internet. And all that, as I said, while the service was typically unneeded, unwanted, and unused. Minimizing a system's attack surface should be taught, and probably is, during Cybersecurity 101. Yet that basic lesson was ignored here with catastrophic results.

Okay. However, the good news is it appears this policy was changed for the better several years ago, though only after all of the servers being attacked had been deployed. In a blog posting yesterday, VMware's Edward Hawkins, whose title is High-Profile Product Incident Response Manager - and yes, Edward, this would qualify as a high-profile product incident - he wrote: "We wanted to address the recently reported ESXiArgs ransomware attacks, as well as provide some guidance on actions concerned customers should take to protect themselves.

"VMware has not found evidence that suggests an unknown vulnerability, a zero-day, is being used to propagate the ransomware used in these recent attacks. Most reports state that End of General Support" - which they call EOGS - "and/or significantly out-of-date products are being targeted with known vulnerabilities which were previously addressed and disclosed in VMware Security Advisories." Those are VMSAs. "You can sign up for email and RSS alerts when an advisory is published or significantly modified on our main VMSA page.

"With this in mind," he finishes, "we are advising customers to upgrade to the latest available supported releases of vSphere components to address currently known vulnerabilities. In addition, VMware has recommended disabling the OpenSLP service in ESXi. In 2021, ESXi 7.0 U2c and ESXi 8.0 GA began shipping with the service disabled by default." What is that about horses having left the barn? But still, this was clearly the correct policy change. In OVH's first posting last Friday the 3rd, they observed, they said: "The attack is primarily targeting ESXi servers in versions before 7.0 U3i, apparently through the OpenSLP port 427." Right. So the moment VMware changed their policy, turned off that unneeded service, and closed that port, their systems were no longer vulnerable.

Now, there's some confusion about what files are encrypted. The encryption code has been found now and analyzed, so we know that it targets all files with the extensions .vmdk, which is the mother lode, as well as .vmx, .vmxf, .vmsd, .vmsn, .vswp, .vmss, .nvram, and .vmem. We know that the encryption appears to use a variant of the cipher used by the Babuk ransomware whose source code was leaked and became public, thus allowing it to be, you know, offshoots to be created, and this appears to be one.

And we know that the encryption was done right. There is no easy decryption path without obtaining the key. In that regard the ransomware note was correct. The ransomware obtained its name "ESXiArgs" because, for every file that it encrypts - and it doesn't need to do many because this is a virtual machine, right, it just needs to encrypt the container. For every file that it encrypts with those extensions that I mentioned, it leaves behind that encrypted file .args, which is containing the specific per-encryption data that is needed to direct the file's eventual restoration.

There was some initial news that the big master virtual machine image, the big .vmdk file, was not being encrypted, which would have allowed for the reconstitution of the system without paying the ransom. All of the other little pointer files could have been fixed, apparently. But everything we're seeing suggests that maybe that was a one-off or a low-probability incident.

In another bit of good news, it may be that the claim of exfiltration and subsequent public exposure is an empty threat. One victim, posting on BleepingComputer's forum about their own post-attack forensic analysis, wrote: "Our investigation has determined that data has not been exfiltrated. In our case, the attacked machine had over 500GB of data, but typical daily usage of only 2 Mbps. We reviewed traffic stats for the last 90 days and found no evidence of outbound data transfer." Of course that's not definitive for everyone, of course, but another interesting data point.

Okay. With all this said, I was left with one other thought: Why were the bad guys allowed to find and exploit this? This problem had been waiting for discovery for two years, while VMware knew that they had a serious remotely exploitable remote code execution vulnerability. We know they knew that this was a critical remote code execution vulnerability affecting all of their ESXi servers at the time. ZDI's Lucas would certainly have shared his own private proof-of-concept exploitation demo with them, though he never released it publicly. And as we know, they proactively changed their policy to no longer have their OpenSLP service running and exposed by default. So there's proof of awareness.

Big, slow, lumbering bureaucratic national governments are now proactively scanning their own nation's networks checking the version of the systems that are publicly exposed. Why isn't a leading high-tech Silicon Valley superstar like VMware, who produces highly sophisticated public-facing Internet servers, proactively scanning their own customers to protect them from the known, potentially catastrophic consequences of using the software they publish and sell? I was unimpressed by VMware's spokesperson blaming their customers for not patching, when VMware is entirely able to know who has patched what and when. VMware is certainly capable of scanning the Internet looking for and checking the security of their own server technology.

One of this podcast's ongoing questions and explorations is about the post-sales responsibility of massively profitable private enterprises whose license agreements state that they're going to take your money, and plenty of it, to support their growth. But what you get in return is whatever they feel like providing, and they're not going to be in any way responsible for what might happen to you afterward as a result of your use of their products for which you paid good money, regardless of what happens. Can you imagine the chaos that would ensue if automobile makers were able to sell their multi-ton vehicles under these terms? Or how about Boeing? "Sure, buy one of our big new shiny passenger jets. We had a bunch of very enthusiastic summer interns design the avionics for it, and they mostly seem to work now."

Cyberthreats are real and growing, but the software industry's perverse and unique utter lack of accountability for its own failings removes the only incentive for improvement that's been shown to work. VMware never bothered to protect their own customers because it's been established that it's their customers' fault for not proactively patching the buggy software that VMware sold them in the first place. That famous definition of insanity is continuing to do the same thing and expecting a different outcome. Well, things are going to keep getting worse unless we make them get better. So far, there's not even a hint of anything like that happening.

I'll finish on a happier note. We have the ChatGPT Astonishing Reply of the Week, courtesy of one of our listeners. And this is really becoming interesting. Somebody said to ChatGPT: "Please analyze and provide a description of the function of code that follows

this statement." Okay, so this is a person who, like, I don't want to say they're lazy. Maybe they're going to turn out to be smart. And we see here in this quoted question the code. So it starts out script source, and it's got a URL to googleapis, ajax/libs/jquery/3.6.1/jquery, you know, min.js. So it's a minified JavaScript. So it loads that. Then the actual script in question, which you can - there's a couple of arguments in the clear, but most of it is gibberish. It's all been scrambled and encrypted, you know, minified and obscured so nobody can see what's in there. And you see little bits. There's a return statement. There's a parse integer and another one and a few. So anyway, it's gobbledygook; right?

Astonishingly, this guy says: "Please analyze and provide a description of the function of code that follows this statement." ChatGPT says, and I kid you not: "The code appears to be obfuscated JavaScript code that makes an AJAX request to the URL, and then it gives the URL, which it has deobfuscated. The payload of the request contains an email address that is passed as a parameter. The response of the request is written to the document. The purpose of the code is unclear, but it may contain malicious content, such as a code that performs unauthorized actions on the client side, or collects and exfiltrates sensitive information.

"I," says ChatGPT, speaking in the first person, "I would advise against running this code as it may pose a security risk to your system and data." Leo, I don't know.

**Leo:** I think it's smart.

**Steve:** I am astonished. Once again, I am astonished. And I do see something happening. I've seen some tweets which are a little concerning because people are starting to ask ChatGPT for help. I mean, they're like, one of the things that needs to be reinforced about this is it is often wrong.

**Leo:** Yeah.

**Steve:** It is not - maybe it's like asking Uncle Benny, who's got a little bit of the, you know, we're not sure about him. This thing, you know, it always sounds authoritative. And so it's like it's selling its own answers. But sometimes it's just like way off. So, you know, we should just remind people, yes, you can use it as Rob did to create a template for some code he would have never written if he had to do it himself. He had to go in and fix it, though; you know? It was broken in a bunch of places. And I'm not letting it get near SpinRite. But for what it's worth, it's worth something. And, boy, I do think it's found its home in search engines, Leo. The idea that this thing could, I mean, it is a search engine, essentially. But stick that on the front end, and we might really see search take on a whole new form.

**Leo:** Yeah. Microsoft announced today they're going to use it with Bing in their Edge browser. And Google's got something they're going to announce, I think tomorrow. So we shall see. It's exploding right now.

**Steve:** It is just astonishing. I know. As I said last week, we're on the brink of something. I don't know what. Nobody knows what. I think this is, you know, still early days. But we need to be careful not to think that it actually has right answers to everything, even though every answer sounds amazing.

**Leo:** Right. Well, we live in interesting times, as they say. And you make it much more interesting, and we thank you for that. Steve joins us every Tuesday to do this show about 1:30 Pacific, 4:30 Eastern, 21:30 UTC.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/>