ABU DHABI – How It Works Understanding DNS Abuse
Sunday, October 29, 2017 – 10:30 to 12:00 GST
ICANN60 | Abu Dhabi, United Arab Emirates

DAVE PISCITELLO: Good morning. We're going to continue the How it Works, focusing in this particular session on DNS abuse, and DNS abuse has become a very, very large, you know, and challenging topic for the ICANN community. I'm going to talk about a much broader set of abuses than is probably going to discussed at various places in ICANN community this week, but it will be, I think, interesting for some of you -- especially some of the technical people in the room, to see some of the different kinds of attacks, and how we've evolved over time.

Just a little bit of background about myself. I'm the vice president of security and ICT coordination at ICANN. I've been with ICANN for about 12 years. I was employee No. 35, so it gives you an idea, you know, how much we've grown. And unfortunately, DNS abuse has grown much faster than ICANN Corporation. Certainly, much faster than the security team.

Next slide, please? So, we're going to look at four different topics during the course of the next hour -- is that what we have, an hour? Yeah, hour and a half -- oh, even better. Okay. And one is -- we're going to start about just trying to explain what

DNS abuse is and distinguish that from what we call, DNS misuse. Then, we're going to talk about and look at some examples, and we'll walk through some scenarios on how attackers can abuse or misuse the DNS. I'm then going to sort of set you up for what you might expect for the rest of the week at ICANN, in terms of the topics that are going to be most commonly discussed.

And then, I'd like to, kind of -- this actually is not correct here, but I'm going to basically talk about some of the emerging DNS threats and landscape. Next slide, please, Cathy. So, when you ask the question, "What is DNS abuse?" it's very similar to walking in a room, you know, of six blind men and an elephant. You'll get different answers from the six people that you'll ask because everyone has a perspective of what abuse is and that perspective is usually filtered by what they're looking at -- what is affecting their own organization? What is affecting their operation or their people? [CROSSTALK]

So, there are three, you know, three typical variants that we encounter. One is that people associate DNS abuse with cybercrime. One is that people associate DNS abuse with hacking or hacktivism, or some sort of attempt to disrupt service, you know, in the DNS ecosystem. The term that came up in the ICANN community, just before the new TLD program

onset, was malicious conduct; it was people who were the DNS in a manner that it was not intended.

So, that was a very broad definition. If you look at what we'll cover today, we're actually going to look at three kinds of issues, and these issues are actually very important to the community right now, especially with the emergence of data protection and privacy regulations. One is data corruption. The other is denial of service, and the third is, obviously, privacy or the breach of privacy. We, in my team, and, you know, one of the things that my team provides to the ICANN community as part of a public interest remit is training for law enforcement in cyber investigations. We distinguish misuse from abuse.

Okay. Misuse is usually an intentional effort to deceive or connive or unscra- Can we widen this because, you know? Yeah, let's just try to do that. Can you still see in the back? Is that better? Okay, thank you. You know, to try to make use of the DNS and the registration systems in an abusive manner or misuse what was intended to be published in a manner that was not intended. Scroll up, please. I don't think the button is going to work here. All right, that'll work. Is there anyone here who works for Adobe? Good, then I can say this, I hate Adobe products. I have every Adobe product I have worked for in my

entire life. I've been doing this for 45 years; I've never seen a product I didn't hate.

UNKNOWN SPEAKER: Dave, we're on record and recording, so thank you for that.

DAVE PISCITELLO: That is my opinion. It is not necessarily shared by ICANN organization. You know. So, let's keep it a little bit simpler. DNS abuse refers to anything that attacks or abuses the infrastructure, whereas, DNS misuse refers to exploiting the infrastructure for some purpose. One might wonder why is the DNS a target for attacks. It's relatively simple. Everyone uses the DNS, every single day, even when you push a button on an app or when you speak to Siri, or you type something in a search engine; the DNS is in the background because the DNS is resolving user-friendly names to internet protocol addresses. And so, you know, even when you visit one webpage, you're using the DNS multiple times.

As an example, if you were to go to www.cnn.com, you would actually be resolving somewhere between 25-45 domain names, in order to render the page that you see when you land on cnn.com. So, there are many, many, many DNS operations going on in the background that provide you with a way to just

simply use one name and render a page, or you know, use a name in a user mailbox arrangement for your mail -- and it's extremely important.

So, because it's so important, people tend to call it an infrastructure, and in many places, you'll actually hear debates about whether the DNS is a critical infrastructure, and therefore ought to be under the control of governments, or you'll hear that the DNS is a mission-critical infrastructure, and thus, can be, you know, very, very important to private organizations, as well as, public-interest organizations and governments.

So, if you disrupt the DNS, this is like causing a traffic jam. It's like, you know, blockading roads. It's like disrupting electrical grids. You're stopping a service, all right. If you exploit the DNS in some way, you can trick people into doing something that they didn't necessarily think that they were doing. And a lot of criminal activity relies on exploiting the DNS to trick or defraud users. There are several different vectors or directions of attack for, you know, exploiting the DNS. One is to maliciously register domain names. Spammers do this. People who operate botnets do this. And the purpose is to actually control part of the DNS infrastructure for, what is essentially, an e-crime name resolution. You could also steal someone else's existing name service, and we'll show you how you can do this.

And then, you can also corrupt the data that the DNS works on. In the previous session, Steve talked about zone data, modifying zone data, or modifying information that's in resolve or cache is one way to go and corrupt DNS data. I wasn't here, but I suspect that Steve talked about the three operational elements of the DNS, you know, we're going to talk a lot about client or stub resolvers, as the pieces of software that are either in an operating system, or in an application, or app, and any device that is connected to the internet, and these are software that ask questions to the DNS. There are some seats up here if you guys would like to come and sit. Yeah, I can get rid of my junk, and there's two seats here, so -- and you don't have to worry about it -- if you sit upfront, I won't call on you. I promise.

Okay. So, the clients ask questions -- well, who do ask questions to? I'm sure Steve pointed out; they ask questions, mostly, to what are called recursive name resolvers. And the recursive name resolvers go and find answers by asking the authoritative name servers. So, the authoritative name servers are kind of like old-school, telephony, directory assistance. They're the ones that hold the data that is, you know, what publishers intended to make public.

So, think of this as, you know, several hundred million, you know, public libraries -- all of which have, you know, information

about a particular name and the kinds of resources that are in that name. So, now that we know that there are three elements, how would you attack any of these three elements? And this table kind of walks you through all the different attack vectors that people have exploited over the years since we built the DNS. And my guess is that -- I know that we started deploying DNS in 1985 -- my guess is that shortly after we deployed it, we probably experienced our first attack using one of these particular vectors.

So, one way to attack a name server, recursive or stub, is to deplete its access bandwidth. So, if you clog up the access circuit, then no resolution can be performed. You can also attack the network elements that help, you know, connect a DNS name server or resolver to the internet. So, for example, a router or a switch. You can attack the name server at the device level, at the operating-system software level, at the name server software level -- often; these are using vulnerabilities that are exploited. I see people nodding, which is really great. I'm glad you moved up cause now I have a nodded head to focus on. We can attack the cache.

Steve talked earlier about how resolvers, you know, can take a peek at an answer and stuff it in, you know, a local file, so that they actually have information and it can save a lot of queries.

You can attack the application software on a name server. You can also attack the administration or the configuration of a name server or any of those devices. By the way, if you have a question, just raise your hand, and we'll get a mic to you, okay. Please use the mic because we have remote participation. Thank you.

So, that's sort of an overview of what abuse and misuse framework is, and what I'd like to do is walk you through, at least a few examples of, some of the different kinds of attacks. One of the things that you can do in a denial of service attack, which the title or name implies, it's preventing the service from working as it's expected, is to attempt to take a vulnerability and exploit that vulnerability and cause the machine to fail.

Okay. Another is to exploit the machine so that you can take over administrative privileges on that machine and make it do something that you want it to do. I'm certain that everyone here has heard of a distributed denial of service attack, and the distributed denial of service attack is actually the combination of two other kinds of attacks that we'll show. One is called a reflection attack, and the other is called an amplification attack.

So, a distributed amplification and reflection attack is what we typically call, distributed denial of service. We can also look at host resource depletion. We can also look at cache poisoning or

exhaustion attacks. And we can also look at DNS man-in-the-middle attacks. So, we're going to take a look at some of those examples. I actually have a three-hour university seminar on this and at some point in time; I promise that I will try to do that as a webcast.

And Ravi Surya is saying, "Are there plans to share these?" Share what? The slides? We just -- someone just messaged us -- [CROSSTALK] Okay. So, yes, the slides will be made available at, you know, the ICANN60 site, and you can -- they're already there. So, you can get up and leave and go -- you know, and ask me questions later.

UNKNOWN SPEAKER: Dave, just to edify that. They're available through the schedule, and if you drill down onto this session, you'll find the slides available on the specific sessions.

DAVE PISCITELLO: Okay. So, let's look at how you actually perform an exploit to fail denial of service attack. You can either be very clever and go and poke around in using various malformed DNS packets, or you can go and look at the common vulnerabilities and exploits database and choose one and see if some name server has not patched its operating system and software, and try to exploit a

known vulnerability. Remarkably, there are many, many know vulnerabilities that still remain unpatched among the, you know, hundreds of millions, or tens of millions of resolvers and name servers that are out in the internet today.

So, it's actually quite easy, you know, if you have a malicious bone in your body, to go and hurt someone. And what you do is instead of sending a DNS query that is correctly composed, you send the query with some of the bits and bytes set in a way that is known to be malicious and known to cause the target's system to fail. So, what you are going is you're going and you're taking advantage of some software flaw that results in a condition that the host didn't anticipate, and the system crashes or the application crashes. And that causes the name server to simply stop answering. It's much more effective than, if you think about it, than a denial of service attack because, boom, the machine is down.

So, one of the examples, I have here, if you go and you look at the common vulnerabilities and exploits database at cert.cc, you'll be able to go and read the details of what this packet looks like. So, one of the nice things about being a good guy is that we publish all the information that a bad guy needs to go and actually take down our name servers, and it's all in the CVE. An exploit-to-own denial of service attack is very similar to this.

The exploit-to-own forces the target machine to provide root or administrative privileges to the attacker.

So, what you've done is you've gone and used a packet or a sequence of packets that will cause something like a buffer overflow on the target machine. With the buffer overflow, what you can do is inject executable code onto the target machine that would allow you to take over root privileges. And so, at that point, you could then go and connect to the machine and make it do anything you'd like. One of the things you could do is go and change the zone data.

Or you can, you know, then -- some people will take that name server and upload and install a small web server and then, host some sort of proxy for some other attack, or they may put other malicious software there to do some monitoring and collection to find out something else about the network that they've attacked.

A distributed reflection and amplification attack does two things. It begins by having an army of attacking machines. These are usually comprised host or in some cases, they're compromised servers or cloud servers or cloud VMs. And what the attacker does is he uses as his source IP address, the address of the target. So, here, all these machines are using as their source address, 10.0.0.1, and here's the poor name server who's

actually at 10.0.0.1. So, the first attack part is the reflection. What we're doing is we're actually going through this open recursor, okay, and an open recursor is a recursive name server that accepts queries from anyone. And they go to open recursor with a particular kind of DNS query, and they bounce and that recursor says, oh, the actual answer that I want to send and return is something at this targeted host.

And so, through that open recursor, the reflection, you know, bounces to the target. Now, DNS queries are small and most DNS responses are relatively small, so if you want to make your attack something that depletes bandwidth, what you do is try to articulate or send a query that's going to result in a large message, so DNSSEC messages sometimes are large. But you could also use an open recursor that you, the attacker, has put together, and you could put a very, very large text message in your own resolver, and that way, what you would do is for each of these queries of maybe 60-100 bytes, you're actually forcing something like a 4,000-byte response to be sent here.

So, then it just simply becomes sheer numbers, okay. The more thousands of attackers sending the more thousands of packets per second of thousands of bytes -- the faster you're going to deplete some resource that's at the target site. So, we are at the point where 600, 700 gigabits per second kinds of denial of

service attacks are possible. And they're only going to faster and more pernicious because the attackers are now using servers. They're attacking and compromising VMs or other large servers that have much bigger bandwidth than traditional end users. I'm going to kill myself.

UNKNOWN SPEAKER: Dave, excuse me, do you want questions now, or after?

DAVE PISCITELLO: Oh, sure.

UNKNOWN SPEAKER: Okay. We have a question from remote, "If a registry wants to do everything in its powers to prevent DNS abuse all across its TLD space, at the registry level backend; registrar level; reseller level; down to the end-user level; and has ample resources, how would it design or redesign its registry operations?"

DAVE PISCITELLO: That's a topic for like a three-month consulting gig, and, you know, when I retire from ICANN, I'm going to be more than happy to actually offer that -- you're asking a multifaceted question that, honestly, is much broader than looking at any

individual attack here. A holistic approach, where a registry would actually have some mechanism to, not only, enforce policy and protect its own registry operations, but potentially partners or supply-chain registrars, and then all end users and resolvers, is -- yeah, it's sort of a massive undertaking that would require, you know, coordination of global resources that I can't imagine anyone has actually ever achieved.

So, what about what we called a host resource-depletion attack? In this particular case, one way to actually force the machine to get overwhelmed is to deplete its memory, okay, so exhaust the RAM in a machine. One way to exhaust RAM in a machine, whether you're using DNS, or any other application that can run on TCP, is to send lots of requests to open TCP connections. These are called TCP SYN packets, okay.

And here, again, you'd reflect it off an open resolver or you could do it directly, but you just keep sending the TCP SYNs, and then you ignore the ACKs that come back. So, what happens is that the target host gets thousands of requests to open a connection. For each of those requests, he has to reserve a certain amount of memory that's called a transmission control block that is necessary to maintain the state of TCP because TCP has window, it has all sorts of other mechanisms to provide reliable delivery without loss and duplication.

So, if you can send enough TCP SYN-ACKs to the targeted host, and the targeted host is not prepared to have a maximum before it starts to drop connections, then you can exhaust all the RAM. If you have a targeted host that's rather smart and starts to drop incoming connections, so that it doesn't get exhausted, one of the things that you can do is you keep sending them, and then, the ratio of your bad put -- your bad TCP connections, is going to be higher than the ratio of what is normal traffic and so, you're degrading the service because now, some of the people -- some of the TCP SYNs that are coming in are conceivably good, legitimate users whom you're dropping because you're trying to preserve your, you know, system's integrity. Any questions? Yes.

UNKNOWN SPEAKER: [Inaudible].

UNKNOWN SPEAKER: Sorry, can I get you to repeat that into the microphone?

UNKNOWN SPEAKER: Wouldn't the firewall just drop the packets since there's no session from that particular targeted host?

DAVE PISCITELLO: So, the question is, "What if I just drop the packets?" How do you know which packets to drop?

UNKNOWN SPEAKER: Because it'll see that there was no previous session from that particular target host requesting for any connection, so when the response comes with SYN-ACK, it'll just see that oh, I don't have an active session, and wouldn't it just drop? I mean --

DAVE PISCITELLO: Well, [CROSSTALK] it's the SYN packets that are the ones that are actually causing you to create the TCV. Now, the SYN-ACK packets coming back, the attacker is actually ignoring, all right. And so, in this particular case, there are algorithms that people have in DDoS prevention software or in DDoS prevention firewalls, where what they do is they start looking at the accumulation of TCP connection requests from particular IP addresses, all right.

And then, you know, if they start to see patterns of behavior that look anomalous, they will start to drop the TCV packets from that particular pool of IP addresses. Criminals, you know, will do lots of different things to try to defeat the DDoS prevention mechanisms, as well, so it becomes a game of cat-and-mouse. Answer your question? Thank you.

Okay. So, I think -- see, this is why you don't use PDF -- all my animation is gone, and this makes no sense. Okay. Let me see if I walk you through this. Poisoning a cache is a means to exploit a configuration that allows a resolver to incorporate information that comes in what's called the additional section of the DNS response, into it's caching strategy.

So, in this very, very nice animation that no longer works because it's PDF, the attacker launches a spam campaign, and he wants to get the IP address of loseweightfastnow.com -- so, he tempts somebody who's received the email, who decides they want to lose weight fast, to click the URL in the email, and the first thing that's going to happen when you launch the browser, and you try to visit this site is you'll try to resolve loseweightfastnow.com.

So, loseweightfastnow.com is going to be hosted at some name server that the criminal has deployed, okay. Sometimes that could be a registrar's name server because the registrar doesn't know until the attack is perpetrated. Sometimes it's one that he's put on a machine that he's compromised, and the attacker is going to send back a response from his name server that says, "Well loseweightnowfast.com is at the IPv4 address of 192.168.1.1."

In the additional section, he's going to say, "Oh, and by the way, if you want to go to www.ebay.com, it's at 192.168.1.2." Okay. Now, is this guy the authoritative for eBay? Not likely, all right. But the local resolver, if it's misconfigured or it's configured permissively to accept additional records, and this was fairly common practice years ago, says, "Oh, look, I now have loseweightfastnow.com's address. I'll put that in my cache, and this gentleman was so kind as to send me, you know, an update for ebay.com. So, I'm going to put that in my cache too."

So, now the attacker actually has two attack points. One is he can, perhaps, you know, host malware or something else at loseweightfastnow.com, and somebody will go there, but the next time somebody in that network that's being served by the local resolver goes to ebay.com, they're going to the attacker's phishing page. And what does that page look like? What part of eBay? The login page -- yes. Give that man cookies.

Okay. There's also other ways to poison, and in this particular case, you would poison host machines. There was a malware, what -- about three years ago, Greg? DNS changer, 2014, or so? Yeah. And so, DNS changer malware would actually go into the local files of a machine that was compromised. It was mostly Windows XP machines, and so, as a result of either a spam message that the user clicked on, or a drive-by download of the

DNS changer malware, it would alter the configuration of the infected PC. Most PCs have information that points to a resolver that they receive when you first get your IP address from DHCP. They're also some other information that sometimes gets put in what's called the HOST file.

And so, DNS changer makes changes to the HOST file, and will force the requests from that PC, either all or selective ones, to go to a malicious name server that the attacker runs. So, the malicious name server, now, is in control of all the name resolution that this PC attempts, and what the attacker can do is, literally, change the attack surface whenever he wants by simply changing the zone file in the name server that these infected PCs are referring to instead of their legitimate resolver.

There's a lot of little nefarious things you can do once you've actually put this software in. For example, you could selectively identify banks, and instead of changing all name resolution, which would be pretty obvious, you just change -- say, I really want to attack Barclay's, or I really want to attack HSBC, so all the rest of the name resolution goes to pass through and look correct, but if you go to a bank, it's going to go to the criminal's site and then, again, as in the case of ebay.com that we just talked about, you'll go to a fake login. Everybody understands how this works? So, this was a very, very big network.

There was actually an underground marketplace for using DNS changer environments and the criminals would actually sell service and opportunities from their infrastructure. And it was taken down through a massive multi-jurisdictional effort by the FBI, and UK National Crime Agency, and several others in 2014. And there are probably still machines infected with DNS changer because one of things that we can't do in most jurisdictions is go and fix, you know, or change, you know, the software or errors on a particular infected machine.

There are some other ways to actually go and defraud or mislead people using the DNS. One is by using domain-name registration hijacking, or DNS hijacking. Another is to -- and this is kind of interesting -- I wrote a column about this for the last ICANN meeting -- you can use the DNS protocol as a covert infiltration channel or a covert exfiltration channel. And we'll show you some of those.

Then, there's a technique that was first seen somewhere around 2007 and 2008. We got fairly good at detecting it, and now it's back -- and I'm going to talk to you about a recent takedown of a major network operation that uses something called fast flux. So, one of the things that is sort of challenging for domain name registration providers, whether they're the registries that provide registration services directly, or registrars, like ICANN-

accredited registrars that provide registrations, is that the registration systems, you know, is basically a merchant operation, all right.

And Steve sort of talked about what this looks like, but, you know, if you want to a domain name in the gTLD space, what you typically do is go to an ICANN-accredited registrar, and you go to their webpage, and the webpage has a web form, and you type in a label or a string that you'd like to register. So, in my case, years ago, I went to GoDaddy, and I typed in securityskeptic.com, and fortunately, it was available because who wanted to be securityskeptic.com, you know, 15 years ago?

And so, the registrar would send back a message through the web form, saying, "Yes. It's available. Do you want to buy it?" or -- you don't actually buy domains, you register them for a period of time, a minimum of one year, and in most cases, a maximum of 10, all right. So, you get the available message, and you register -- you go and you pay to have the registration performed, and this is actually a really, really important process because once you've paid, four databases that become very important in criminal investigations or in operations or in businesses get populated.

One is the billing detail; this is where your credit card information, your other payment method is stored by the

registrar. The other is the WHOIS record; this is the publicly displayed information about who has registered the domain, where the name server is that hosts the authoritative for that domain, and other information.

The registrar and the registry work with a protocol called EPP, Extensible Provisioning Protocol, to share all the information that the registrant (me) identified as how he wanted his WHOIS to look, where he's going to host his name service, and what records were going to be in his zone file -- and once that's done, the top-level domain registry actually puts A records and name-server records into the top-level domain for, you know, the registered domain.

Okay. There's also some registry-specific information that goes into a registry database. Lots of the information in those four databases becomes relevant to criminal investigations. Some is accessible publicly, some requires a court order, but the information is quite important, in order to be able to quickly remediate, you know, some sort of malicious attack.

One of the things that is very important to most registrars is that the businesses are very commodity-driven business for most commercial registrars. So, you want the process to be highly automated, rapidly provisioned; typically, less than hour for most domains to actually come live, all right. The

correspondence between the registrar and the registrant is largely email, okay, that should immediately, you know, you're warning Will Robinson, you know, issues because anything that can be done with email can usually be defrauded in some form of phishing attack. Inexpensive registrations are plentiful, okay, so people have a large choice of going out and getting registrations that are very, very cheap, and this is very good for growing consumer markets, but it's also very, very good for attackers because they don't have to spend a lot of money to go get lots and lots of domains.

So, why would a criminal want to registry a domain name? Well, we talked earlier about having names like loseweightfastnow.com; the criminal is going to register that domain name because he wants to host content there that's going to be deceptive or going to provide him with an opportunity draw people to his site where he's installed some sort of malware that could be downloaded, and then, someone could unwittingly infect their own machine.

Criminals also use domain names to host mail exchange, so they can actually send spam from lots of domains. In particular, if you look at the illegal pharmaceutical market, you know, the people who run illegal pharmaceutical networks will generate hundreds of domains that have Rx or meds or drugs or some

other word in it to try to get people to think, "Oh, this is prescriptions. This is some place I can go and get my prescriptions for free, or less money than I would normally have to pay."

Same thing with piracy sites -- counterfeit goods, you know, Rolex, spelled with R-zero-l-e-x, is the classic example. So, attackers will, in some cases, register names that are visually similar. In some cases, they'll register names that have the brand somewhere in the name. In some cases, they'll have a subdomain that has the brand. So, for example, you know, somebody could put rolex.example.com, once they've registered example.com, all right. So, there's all sort of different ways that attackers could register names and then use them.

One of the things that attackers normally do, especially spammers, is register lots of domains because they know that there's a triage or a first responder of individuals, and I'm very fortunate to be apart of that group of people, who are out on a 24-hour basis, capturing spam, looking at the domains, determining whether or not they're malicious, and then putting them on what's called a reputation blacklist, okay.

And those reputation blacklists, typically, keep somewhere around 80-90 percent of the 400 billion spam mails, a month, from your desktop. And Greg Aaron and I are publishing an

article this week about reputation blacklists for ICANN60, and I encourage you to read it because it's very, very useful to understand just how ubiquitous registration blacklists are and how important they are to keeping us all from being completely inundated with trash email and threats.

So, the other thing that's important for criminals, in terms of registering domain names, is naming name servers for crime resolution and, also, naming what are called command-control systems for botnets. Everyone familiar with the botnet? Is there anyone who doesn't know what a botnet is? Okay, good. You don't know --?

So, a botnet is, essentially, a network composed of, typically, infected computers, and the infection that is put on a computer is normally called a dropper file, and the dropper file executes -- it self-extracts, and it goes, and it finds a way to run, in stealth, on the infected machine. One of the first things that the dropper file wants executed does is it goes and looks for instructions from a system called the command and control. The command and control is run by the attacker, or sometimes, it's leased to someone else, for payment. And then the command and control can tell the dropper, "Download this file, and put different executable software on that machine."

So, once a machine's infected, it can do all sorts of different things, depending on what the command and control says. It could run a mail server for spam. It could run a web proxy. It could host the DNS service. It can do any number of things, you know, that happen to be beneficial to the attacker or the person that the attacker has leased his botnet to. Good? Okay. And now, all those of you who didn't really admit that you knew what a botnet -- know what a botnet is, all right.

So, okay, there's also hijacking. Now, some people don't even want to pay the 99-cents for the 1,000 domains, or maybe they don't have, you know, credit cards that they've purchased on the dark web to pay, and so, what they'll do is they'll go and they'll try to gain control of a domain. And one of the ways that you gain control of the domain is using social engineering or a phishing attack. And I'll show you some examples on the next slide.

So, the idea is that just as I went and I created that account at GoDaddy with a username and password, and that became where I would manage my domain portfolio, thousands -- you know, millions of people do this, and organizations do this, worldwide, all right. So, this is just yet another way to go and, you know, and conduct a phishing attack, get somebody's login

information, and then, instead of having their eBay account, what you have is a domain name account.

So, if someone were to phish me, and I fell for the phish, and I went to a page that looked like GoDaddy, but the page was actually the criminal, and I typed in my username, and I typed in my password, they could then manage securityskeptic.com's zone file, and they could do whatever they like with it. So, one of the things they could do is they could change the IP address of where securityskeptic.com was hosted, and they could put up an abuse site. Okay.

They could put up a defacement site, mocking me, saying, "Oh, you're the big security expert, and you don't even know how to manage your domain." All right. Much more nefarious, would be putting up an offensive site, having hacked a government webpage, right. That's much more troublesome, and, you know, it terribly embarrassing for the government, all right.

Another thing you can do once you've controlled the zone file is you can add name servers. You don't necessarily have to change the name servers; you could add a MX address, and if somebody's not paying attention, routinely, to what their zone file looks like, the criminals can simply use your name server, name, and their own IP address to host other criminal activity.

So, what you typically get, and I think ICANN's Security and Stability Advisory Committee wrote an advisory about registrar phishing, probably four or five years ago. I was still on SSAC at the time. So, basically, the criminal goes, and he looks for the regular correspondence that would come from a registrar. One of those regular correspondences is called the annual WHOIS accuracy compliance form.

Okay. It's a form that ICANN-accredited registrars must send, annually, to say, "Please make certain that your WHOIS information is correct because if there is an inaccuracy, and somebody complains you may lose your domain." So, what is that playing on? That's playing on people's fear and uncertainty, and a lot of people will go, "Oh, let me go and make certain that my information is correct."

And so, they'll go and they'll login to GoDaddy or to Tucows or whoever it is that, you know, their registrar, and of course, they're going to a phishing page when they do so, and they're giving up their domain portfolio. Another thing that you can sort of consider is what are the consequences of other data breaches on domain name registrations and the opportunities that exist?

So, I think all of you are familiar with the Equifax breaches, and the Yahoo breaches, and the OPM in the United States government, which basically counts all Americans', you know,

email. If any of those accounts use the same email address and the same password for their domain portfolio or their bank, as they used for their mail. And lots of people do that, you know -- if you go and you ask how many people have password as their password; or their mother's maiden name; and 1234; or qwerty, Q-W-E-R-T-Y; 1234; cause that's a strong password, right.

So, you can think that not only are people reusing passwords and putting themselves in jeopardy for their finances, but they're also exposing the DNS, possibly, if you're reusing passwords. So, what's the message? Don't reuse passwords. Never reuse a password, all right? In fact, one of the things that ICANN's SSAC has recommended is that if you're going to manage domain names in any meaningful name, consider having a special email address that's just for that -- distinct from any other email address that you might use.

And, you know, you can even do this with like a Gmail, and it's actually a good way to have your email address not disclose anything personal. So, as an example, I could create a Gmail account that is domainadministratorsecurityskeptic@gmail.com. Nobody knows who that is, but for me that's the account that only I use and is only used for my domain registration, and a lot of large corporations do things like this, so, you'll see

domainadmin@cisco.com, or some other brand in that like. Yes, Steve.

STEVE CONTE: Dave, just to right on a little bit because managing multiple email accounts can get frighteningly complex. You could do that with just an email alias too, and have it all point to one email, but then you could use the two -- or that address has a filter to find out who's using that right?

DAVE PISCITELLO: Yeah. The only problem I have with that particular strategy is that you're using the same password. Yeah. And I'd prefer to have a password manager like 1Password, or KeyPass, and I would simply put, you know, put the accounts and the unique email addresses in that password manager.

STEVE CONTE: Yeah, you're right; I was looking at it from a spam perspective, my apologies.

DAVE PISCITELLO: Yeah. Another other questions? Yes. Mic, please. Hold on. It's coming.

UNKNOWN SPEAKER: Yes, the problem with your suggestion is that you might to forget to login with the new email address, and then, it will be revoked, or canceled by Google, or by Yahoo -- and this is a problem. You need to keep logging every month, at least, or every three months.

DAVE PISCITELLO: So, [CROSSTALK] that hasn't been my experience with Yahoo. I'm revealing way too much information, you know, you're all social-engineering now, yeah. So, I have my, you know, a couple of my domain names and the email addresses at a Yahoo.com address, and I don't visit it -- you know, maybe once every six, eight months, you know, and it doesn't get -- quite honestly, the other thing I find is that if you've only put an email in WHOIS, you never get spam.

I mean, I think this comment about, you know, email addresses being harvested from WHOIS is urban legend -- at least, from my experience because I never get spam in my registration accounts. Setting that aside, if there are policies, you're absolutely right, you have to pay attention. But, often, some people will do this under a second domain delegation.

So, for example, if I register securityskeptic.com, and I was going to building up a portfolio to protect my brand, I would also register securityskeptic.ua, or securityskeptic.net, and then I would use one of those as the email address for, you know, my registration of maybe all three, or I would just distribute so that I had some resiliency. There is an overhead, all right, but it's always a question of risk and reward, you know, the little bit of overhead that you put into managing your email might actually save you from, you know, having your domain account hacked and so, you have to decide, as an organization, or an individual, how important that is -- but good questions.

So, I talked to you earlier about fast flux. And fast flux is actually a very, very interesting abuse of the time to live field. I think Steve probably talked to you about a time to live that can be associated with either all the records in a zone file, or with individual records in a zone file, and it essentially says that you know, when you receive information, or you receive this resource record in a DNS response, it's only going to be valid for the time to live. And if the time to live expires, you know, or passes, you should go and re-query to get the new time to live or the new record.

So, what the criminals do is they will go and they'll put a record in -- an address record in, and they'll have a very short time to

live -- 300 milliseconds -- like five minutes. And after five minutes, they actually go and they change the address record of the particular resource, whether it's a name server or it's a web proxy, or some other piece of their infrastructure, and so, every time the time to live expires, they move their resource to a different IP address that is typically one of their many thousands of infected computers. This is a very, very good way to keep investigators from narrowing down and honing in on an individual address.

Now, it does turn out that what -- as a consequence of doing this, the investigators can typically map out a fair amount of a criminals typology, so there's a cost to the criminal for using fast flux, but there's a also a very, very big benefit because criminals are replenishing their botnets, you know, almost minute-by-minute. I mean, the more malware they can actually get out into the public mailboxes or, you know, by spam, and the more malware they can host somewhere else to draw people to an install, the more botnets they can recruit, and botnets, now, have, you know, hundreds of thousands, and in some cases, millions of infected computers.

So, revealing 50 a day is not a big deal because they've got, you know, several orders of magnitude more than that. So, what you see is often the fast flux begins as an attack, or is used in the

attack, primarily, to do web proxies. And so, the web proxy is actually, where somebody will connect when they go to one of the fake login sites, or of their impersonation sites.

And the proxy, actually, then goes back to the resource that the criminal is actually hosting his content or collecting information from. Double fast flux is actually quite interesting because not only does the criminal flux the web servers or web proxies, he fluxes where the name server is. So, he's altering his infrastructure at two different layers, which makes it very, very challenging. Okay, let's pivot just a little bit and talk about how -- yes, go ahead.

UNKNOWN SPEAKER: [Inaudible].

DAVE PISCITELLO: You need a mic. We have a mic? Here, use my mic.

UNKNOWN SPEAKER: How fast flux attack works in DNS zone or address record -- how the attackers use the flux -- fast flux attack?

DAVE PISCITELLO: So, the criminal goes and, essentially, modifies his zone file -- through an automated manner, to change the address records in the zone file. So, he's going -- let's say he has webproxy.badguy.com, okay, the A record starts as 192.168.142.74. After 300 milliseconds passes, he has an automation that says, "Switch it to arbitrarily one of my other IP addresses." And then, he would switch arbitrarily the next 500 milliseconds.

Now, it turns out that we can't just simply say, "Nobody can use short times to live." And why would you think we can't say that? There's a kind of operation called a content-delivery network, Amazon, Cloudfare, you know, people who want to have the ability to move and locate content, especially locating content that's very regional-specific, or is very topical for a particular part of the country, or globe.

And so, for example, Amazon, may have, you know, a merchant incentive or a sale of products that are actually much more relevant in the UK than in the United States, so what they will do is they'll push content to servers that are more local, so that the latency to get that content is slower, and so using short times to live is actually very, very valuable to them, and they pay companies like Akamai, you know, enormous amounts of money

to actually go and, you know, build these networks and have that very resilient, very agile infrastructure.

So, the criminals actually are emulating one of the highest-end premium services -- unfortunately, they're doing it on our own equipment. Does that answer your question? Good. Good questions. So, one of the things that you can do, if you are cheap and you were smart, and you don't actually want to use the web, but you want to do something like Telnet to a remote host, is you can actually tunnel Telnet or Secure Shell over the DNS, using either 1 byte at a time, in one of the fields, or you know, using some sort of mechanism that you're destination actually knows how to unbundle.

And this is a very common technique for people who want to bypass a pay Wi-Fi portal because DNS is usually open, and the first thing the DNS normally does is the first time you try to resolve something, and go to the page using http, you get redirected to the login page where they say, "Here, come and pay money before you can use the Wi-Fi." Well, what if you didn't have to do that? If you could go and use the DNS protocol to just simply open up a Telnet connection by tunneling Telnet in DNS, you don't have to spend the money.

This is -- and the article I wrote, last year, identifies the sites where you can go, GitHub and elsewhere to buy this little piece

of software if you are that cheap. But it's also a good way for attackers to hide the fact that they're connecting to a compromised machine from your network. One of the other things that you can actually do with a covert exfiltration channel is send information using labels and sublabels as the information.

So, the way that you do this is you, you know, identify a machine that's running a variation of a name server at port 53, but what that machine is doing is it's actually parsing the labels that you have in your query. So, the way that you compose is that the criminal registers example.com -- anything left to left of example.com has a specific meaning for the criminal, okay. The third level might be -- you could actually pull out information from a sequel database.

So, imagine that what I could do is create a label that says dave.piscitello.socialsecuritynumber.streetaddress.city.state.us.postalcode.example.com, okay? That's just a name I'm trying to resolve, and what I'm really doing is I have malware that's using a DNS covert channel to pull information that I've managed to find in a sequel or other database, and upload it covertly -- because who's looking at the labels? All right. So, it's very, very, you know, nefarious.

Now, that's also fairly obvious, right? Because eventually somebody's going to sit there and go, "Well, wait a minute, you know, doesn't this look like, you know, a record?" So, the next thing that the criminal will do is obfuscate all those labels by using either base64 encoding or encryption. So, now, it's just, you know, some random string of letters, and they're even ways to take a random string of letters and turn them into natural language English, so that the labels look plausible.

So, there's a lot of very, very interesting ways to actually go and build the covert exfiltration channel. Suppose what I want to do is not have my infected computers use http for accessing my command and control, but the DNS -- I can also use the DNS to have my dropper file that I talked about before download new executables by asking for text records at my name server.

So, you know, the text record would actually be some chunk of an executable code, right, in, you know, base64 or something else, and my little malware that's sitting there as my dropper, sitting there going, "Okay, I'll just use the DNS. I'll grab this, and I'll keep going until I have the new version of what I want, and then, I'll just install that." Everybody understand that?

There are two examples in the wild, Feederbot and Morto, if you want to go look at them, just do a search on that. You can also

look at the description I have in the article I mentioned. How we doing on time? Half an hour? Okay, good. All right.

Those are all the examples of the horrible things that can happen in the DNS. We should just not run it and go home, right? We're going to try to figure out ways to manage abuse or mitigate abuse and this is a very, very important topic for ICANN and ICANN organization and community. Unfortunately, it's a very, very challenging topic because mitigating abuse costs, you know, it costs operationally; it costs, you know, in terms of manpower; it costs, in terms of changing policy; there can be contractual implications, and as result, you can get, you know, 100 people in a room, all with very, very different and in sometimes violent ideas about how to actually manage this.

The topics that you're going to see this week that are relevant to trying to figure out how to mitigate abuse are at least these four: WHOIS accuracy; the general data protection regulation coming out of the European union; public safety, there's actually a working group for the Government Advisory Committee that focuses on this, and I'll talk about it in a moment; and abuse reporting. Okay.

The Government Advisory Committee actually came forward during the course of the new gTLD-application guidebook, and application-process policymaking, and said that we really want

the new TLDs to focus on, you know, having abuse mitigation in their DNA. And one of the things that we want to make certain is that there are prohibitions against, at least, these security threats: malware, operations of botnets, phishing, piracy, trademark, copyright infringement, fraud, and deceptive practices.

As a consequence of that, there is a specification from the GAC that is currently being implemented in a framework that called Spec 11(3)(b) -- if you hear that conversation, and it's an obligation that the registry operators have to go and report on levels of abuse. If you notice, spam was not in there. Well, spam actually is extremely important and I think I'll talk about that separately, afterwards, but some people who were reading the GAC specification and recommendation were -- said, "Well, spam's not listed, so we don't have to pay attention to it."

In Hyderabad, last year, the GAC said, "No. That was not an exhaustive list that we gave you, that was a list of things that we thought were important. Spam certainly is something that we do is important." And even this is debated by people in the community, as you will see. There are a lot of discussions and debates about whether spam is content, whether domains that are in spam are content -- I'm going to be publishing a blog piece that's a companion to the one that Greg and I have written,

sometime later in the week, that talks about, you know, why spam is a threat. Okay.

So, the groups that will be looking at this, and some of these are private meetings, because they are all law enforcement at a national level, working with the government, but the public safety working group is primarily law enforcement, and prosecutors, and invited subject-matter experts, who are working to try to understand the impact of GDPR, WHOIS accuracy, fast flux, and other abuses on ICANN policy, and they have their own sets of recommendations.

There is also something that is not necessarily domain-name system related, but is a big concern among the law enforcement, and it's called Carrier Grade network address translation -- has anybody heard of this -- CGN? So, let me take a moment to explain what this is. Are you familiar with network address translation?

So, network address translation, just real simply, is a way that was a stop-gap mechanism to deal with the exhaustion of IPv4 addresses, and so a little box like a router in your home, or your access point, assigns private address space to anything that's in your house, and it's all mapped onto one public address. So that many to one saves lots of public addresses from being used.

Carrier Grade NAT takes that principal and pulls it into the carrier's network.

So, the ISP is actually using private addresses to his customers and then, mapping those onto either private or public addresses inside his network to even further save his IPv4 addresses. The problem that law enforcement faces is that whereas the mapping and address translation that they used when the equipment was on the customer's premises was something that they could use to identify location. Once you pull that address translation in the carrier network, the only place where that mapping is seen is in the carrier's logs.

So, you have an entirely different challenge as law enforcement, or an investigator like my team, in trying to find out what address was this criminal really using, all right. And just like NAT, those addresses are dynamic, and so they can change, which means that trying to isolate, you know, who was using it comes down to not only trying to get the data, but getting the timestamp and knowing when the window of attack was.

So, it's a very, very important debate. The solution is actually to, you know, implement IPv6, and lots of organizations, lots of ISPs have sufficient IPv4 space that they're reluctant to go and spend the infrastructure change to do that. So, this is another very

controversial issue. I mentioned fast flux is actually coming back into the conversation. Yes? Do we have a [inaudible].

UNKNOWN SPEAKER: It's just a question on the CGN. If the operators were to not use the CGN, and have every subscriber use a public address or be allocated in a public address, would that not raise up costs?

DAVE PISCITELLO: So, that's a hard question to answer. If you are a carrier, and you migrate to IPv6, or you complement your existing IPv4 network with IPv6, you'll have to go to a regional internet registry and acquire a block of IPv6 addresses. Those blocks come in massively large numbers of addresses, so once you spent that initial outlay, you're going to have hundreds of millions of addresses, so you should have plenty of addresses to manage growth in your own network.

The cost that most of the carriers are trying to avoid is not the addressing cost, it's the cost of retraining their operational staff, retraining their help desk, changing the way that they do routing -- basically, learning IPv6, and learning how to manage it and how to secure it. It's a business decision for the carriers. Some of them have, you know, a very, very, you know, heavily invested infrastructure that is very well-tuned and operational for v4.

They've got addresses that are sufficient, and so, they're plan is to just keep running IPv4 until, you know, that investment they have now has been amortized and, you know, and depreciated or whatever they want to do to make it now cost-effective five years from now to go and do IPv6.

So, it's a stopgap. You can't just run, you know, Carrier grade NAT forever because we're eventually going to have IOT devices and other devices that are going to increase the numbers of, you know, end points on the internet to, 2, 3, 4 orders of magnitude more than what we have today. All right. Does that answer your question? Yeah. Yes, no? You're smiling.

UNKNOWN SPEAKER: I understand that when they migrate to IPv6, this NAT problem will not be there, but say, let's say, they have enough IPv4 addresses, yeah -- in Africa, there still some parts of the continent where you can find enough IPv4 addresses. Yeah. But then, in the IPv4 environment, does it not raise, or does it not save costs to do this [inaudible]?

DAVE PISCITELLO: So, if you -- I guess the question I would have if I were going in, and I were doing, you know, network consulting as I did before I came to ICANN, is why would you invest in Carrier Grade NAT if

you already have enough IPv4 addresses? It's adding another level of complexity because this is not an easy thing to configure and run. You're running routing at two different levels. You're assigning addresses at two different levels.

You've got, you know, a lot of different operational issues that you have to learn, and so, as much as I understand why somebody would want to do this, I actually don't really think that the -- when you finish deploying this, you will have saved any real operational overhead from just going to IPv6. All right. And I'm not a big fan of IPv6. So, I can -- you know, well, the reason why I'm not a big fan of IPv6, quite honestly, is that I wrote a protocol that competed for being IPv6 in 1993, and it wasn't chosen. So, I'm bitter, bitter, bitter, bitter. Yes.

MILTON MUELLER: This is Milton Mueller at Georgia Tech. When you switch to IPv6, you are still running IPv4.

DAVE PISCITELLO: In most cases, that's the recommendation.

MILTON MUELLER: In all cases unless you want to cut yourself off from 96 percent of the internet. So, the reason people use Carrier Grade NAT is not

because they wouldn't like to just switch to v6; it's because they have to maintain the v4 internet at the same time, until 96 percent of the world goes towards v6, and they can turn off the v4, right?

DAVE PISCITELLO: Well, there are 6 to 4 solutions, and 4 to 6 solutions, and it all -- I mean, there are people who argue that you don't have to do dual-stack, all right. And I don't necessarily believe that they're sane, but I don't get to say that. Well, I guess I did. Let's move on from v6 cause there's several other things I want to share with you, and then, if you guys want, we can have a conversation about v6 with people who are probably more sympathetic to that deployment.

Okay. So, the other things that you may hear about, especially when we start to talk about GDPR, and especially our contract considerations -- there are, just to let you know that there are contractual agreements and considerations to DNS abuse in the registry agreements, and in the registrar agreements, and I'm not going to read these because there are some more interesting things to talk about. Some of the places, where they'll see fireworks, and there'll be all sorts of blood-shows and goring, will be the cross-community session on -- towards effective DNS-abuse mitigation.

There's an interesting study that was performed by some people at SIDN on the statistical analysis of DNS abuse in the gTLD registries. This project is very similar to one that I started at ICANN with Greg Aaron and some others, called the domain abuse activity reporting system, the DAAR system, in case you happen to hear that sometime during the week -- there will be the GAC's public safety working group presentation to the plenary. Some of these are closed, some of them are open, but you'll find that there are plenty of sessions where they're talking about DNS abuse.

What I want to do to finish is to talk about three attacks that you may be familiar with, and sort of show you the different abuses that were present to kind of give you a case that Dave's not making all this up. Okay. So, one of the things, we have seen that, to me, is sort of frightening is that not only are botnets getting to be more, they're getting to be bigger, and they're getting to be better, and they're becoming a commodity. We actually have DDoS as a service on the internet.

You can go to a cloud-based operator, and they will DDoS anybody you want to point to for -- anywhere from, you know, $10 to $200 an hour. Fast flux and double flux are actually present today. They, you know, were resurrected in a very, very large spam infrastructure that I'll talk about called the

Avalanche malware. The internet of vulnerable things is another very, very interesting and worrisome attack vector. People -- are you all familiar with MORI? You know, the -- yes, no? Yes.

So, we'll talk a little bit about MORI, and then, recently, WanaCry and WanaCrypt, sort of gave us an insight into how the DNS can be an ignition switch or a kill switch for, you know, launching a malware or another kind of attack. All right. My personal feeling is if you really wanna cry, you have to cry over the fact that DDoS kits are available from Facebook, they're available from GitHub, they're available from the Darknet, and they're called booters or stressors, and if you want to just go find them, all you have to do is type in booter or stressor, and yeah -- it's almost as easy as it is to buy shoes. All right.

There is actually a booter blacklist that is quite interesting, and I want to spend a little bit more time looking at this, but this list tries to identify all the lists of websites that publicly offer denial of service attacks, so you got to think, "Oh, my god, there's already a blacklist -- how many can there be?" Well, there's a whole lot of them. All right.

And as you can see, there are YouTube's, you know, how do you DoS your friend as a joke, right? Well that's, you know, not really a joke; it's actually a criminal activity in many jurisdictions, so I would be very careful doing that, and I love the I am not an

attorney, but this is my disclaimer that should protect me from going to jail statement that a lot of the YouTube or other hosting sites post. I'm just telling you how to do this, I'm not telling you that you should go do it -- doesn't necessarily keep you from incarceration, as far as I understand.

So, let's talk a little bit about Avalanche. Avalanche was probably one of the most successful, large infrastructure takedowns, and it received very little attention because it happened somewhere around the same time window as WanaCry. And so, what Avalanche was, was a criminal malware and DNS hosting infrastructure. Okay, it was a cloud-service. You could go and you could, basically, subscribe to a botnet. You could choose any kind of malware that you wanted to use, and you could, basically, it was like going to a merchant site. You chose what you wanted, you paid your money, and you could launch an attack. All right.

They offer you everything. They offer you bulletproof hosting because they were using fast flux and double flux. Primarily, this was used for financial fraud attacks. They would use ransomware -- everybody, know what ransomware is? Anyone who doesn't know what ransomware is? Okay, good. So, this was a very large ransomware environment, and what you would get with your subscription or your payment was criminal domain

registrations for your command and control, access to a command and control server, access to any choice of the malware that they supported in their infrastructure.

What this network did was generate spam, okay. Hundreds and millions of spam -- spam is sort of the delivery mechanism for the criminal attack, which is why, you know, we think about it very seriously in our domain activity reporting tool.

So, let's talk a little bit about how long Avalanche has been around. Avalanche came on the tail-end of what was called the rockfish group. The rockfish group was a very successful spam group that operated up until 2008, and by that time, we had understood their behavior well enough to suppress most of the spam that they were generating. So, we were putting them out of business. Well, they didn't want to be put out of business. They wanted to make more money.

So, they evolved into this spam infrastructure, this new botnet with a new fast flux and double flux environment, and over the period of time from 2010 to 2012, it became this delivery service that we eventually took down. From 2012 to 2016, the public and private investigators, law enforcement, you know, private investigators from research, academia, and commercial companies, identified operations of Avalanche in 30 countries, 64 top-level domains, 40 backend operators, and it took that

long to get court orders and mutual legal assistance, in order to execute the dismantling action on November 30, 2016. Okay.

That's a pretty long timeline, isn't it? Think of all the money that, you know, was fraudulently taken from people with ransomware attacks over that timeline. Why? All right. Was this the fault of all the registries and all the registrars? Hell, no -- this is the problem of trying to contend with criminal activity that operates at an internet pace, cross-jurisdictionally, in, you know, MeetSpace, okay?

One of the problems -- so, this is a typical attack you can't possibly read this, but often what happens is -- and this is what Avalanche did was, you know, the botnet operators would go register domains, so Avalanche did that for you. They would set up there infrastructure to actually deliver malware and start sending spam, and then at hour zero, they would launch it. In the case of Avalanche, that's when you actually purchased your attack, and you paid for it, and started your attack and collecting revenue.

So, in the first one to 12 hours, typically, consumers are infected. Private sector actors, primarily, are the ones who sees this first. Law enforcement can't possibly be everywhere at once, and private sector have the freedom to look at network traffic from lots of other points that law enforcement can't necessarily see,

for example, inside corporate networks, and so, the private sector started to identify and map out the botnet. Law enforcement eventually gets called into play when victims start making notifications, and a case is opened. Okay.

From the time that the case is opened till court orders and MLATs are all put together, the private sector operators, the registry operators, the DNS name server operators, all get coordinated to actually take down the botnet, takes not just months, but sometimes years. One of the big problems that we have in trying to mitigate cybercrime is that mutual legal assistance was not designed to operate in hours. It's a month-long process. And if you want to put somebody in jail with court orders, you're not going to do it in an hour.

So, we have this big mitigation void that lies somewhere between months and years that we need to reconcile in our legal system. The outcome of Avalanche, I have a longer presentation to this, by the way, that's actually quite interesting -- got some interesting pictures of, you know, the criminals trying to escape off a balcony. There were five arrests, in four countries; 37 searches; there were 39 servers seized in 13 countries; 221 servers were taken offline when we stopped resolution for all the names. One of the things that we had to do, in order to take this botnet down was to suppress the resolution of thousands of

domain names. All right. In fact, 830,000 domain names. So, that's a massive number of names. It's a very, very big -- very challenging operation to coordinate across 63 TLDs.

Okay. I only have a few minutes left. Five minutes. I'm going to go quickly through MORI. I'm only going to talk a little bit about MORI from the perspective of how dumb people are. Okay. One of the problems that we're going to have with the internet of things is that a whole new generation of developers, whose primary purpose was to write software for technology that hitherto had not been connected to the internet, are running out and grabbing Linux builds, and they know nothing about Linux, and so, they grab it, they look at it, and they think, "Oh, okay, this is cool. I have a TCP/IP stack, I'll stick my application on top of this, and I'm done."

And you know what? They have this really nice little protocol called Telnet, and we can just use Telnet, and we can get to our little device no matter where we are, and then we can do some remote management. So, they don't have any idea of the evolution of securing and hardening operating systems that began in the mid-1980s. They're missing 32 years of knowledge.

All right. And this is the problem that we have because they don't know the history. They don't understand the hardening, but they're going into a commodity-competitive market, where

haste to market is driving mistakes. Okay. This is a very, very big problem that we have. God help us, you know, these little devices are going to have lots of personal data, and if you can connect to them with Telnet, then, you know, GDPR is just not going to ever happen.

All right, going to skip that, and that's what I just talked about -- PDF -- What's this? All right, so, let me -- I don't know how I can do this then, cause that's the important picture. Do we have the PowerPoint version? Now, you know why I hate PDF. I know. I'm not blaming you. I appreciate all your help. I'll buy you lunch.

So, people have all heard of WanaCry and WanaCrypt, right? So, let me see if I can begin -- sorry, let me -- WanaCry was rather remarkable for its lack of success, but enormous hype. Okay. It turns out, you know, various people that were part of the global law enforcement investigation, who actually understood a lot about the payout system, that ransomware operators were using, said that they basically didn't -- that they barely made $63,000 out of the WanaCry effort. Now, if you read all the journalists' reports and, "Oh, my god, you know, millions of people were put at risk, and hospitals were taken offline, and nuclear missiles were just seconds away from the launch button."

All that was ridiculously overblown. I see that on my Mac a lot. Yeah. So, there was -- let me see if I can cut to the chase without the presentation, Kathy. Why don't we just --? So [CROSSTALK] it's wheezing over here. Yes. Well, you see what I said in my slide, so I don't lie. Keep going, we're almost there -- god, this is going to be so anticlimactic if you don't think it's interesting -- see all my nice animations that you didn't get to see. Yeah, it's the last -- I think it's the last two slides -- yeah, go back one, and go back one more, and one more, all right.

So, one of the things that was happening when WanaCry first started appearing as infections is that different private researchers were examining the traffic, examining, you know, the malicious payloads -- so they got a copy of the executable, and they started doing what's called reverse engineering. When they were reversing the algorithm for the generation of domain names, they came up with -- why don't you go to the next slide -- they came up with this ridiculously long name that actually didn't -- doesn't appear on this slide because it's on the previous slide, all right, but one malware reverse engineer was looking at it early in the morning, getting tired, and he said, "You know, I wonder what will happen if I just go register this domain name?"

So, he went and he registered the domain name, and the whole botnet stopped. All right. So, he registered one domain name,

and he killed the entire botnet. And he's sitting there going, "Wow! I'm really great. How did I do that?" So, it turns out that the malware author, this is what we suspect that the malware author was trying to make certain that if somebody were to take his malware and try to reverse engineer it in a sandbox, in a place where it couldn't do any harm -- that he would know if he followed this following logic -- if, my ransomware doesn't connect to the C2 system, that means that the domain name isn't registered so it's safe to encrypt the victim system, but if it connects to it, then it must be in a sandbox, so I'm not going to encrypt it; I'm going to exit.

So, by registering the domain name, the malware author -- or the malware reverse engineer, actually, forced all the malware all over the globe, to shutdown. So, they all process exited because he was trying to outwit the reverse engineers, and he outwitted himself. All right, so this is a very "duh" moment, right? But, you know, it's brilliant, and it just goes to show that these guys are not the perfect hackers that we see, you know, in James Bond and other movies. And, in fact, almost anyone who is in law enforcement will tell you that getting lucky and waiting for a criminal to make a mistake trumps really, really good investigations almost every time.

So, I had hope we'd have a few minutes to talk. I'm going to free those of you who would want to go and have lunch, but if you want to sit and talk a little bit about is this the best we can do -- I'm more than happy to do that. I'm delighted to see so many of you are interested in this topic. My team does a lot of outreach on abuse. We do a lot of presentations in association with the Global Stakeholder Engagement's team, all over the globe.

So, if you see one of us, the talks are usually like this. We also do a lot of writing at the ICANN blog, so there's a security awareness series that you might want to read. And I have about 500 or 600 articles at securityskeptic.com, probably a good one-third of those are on DNS. So, as a shameless self-plug, I don't make any money off the blog, but it's certainly nice to have people come and read and say, "Oh, that's cool."

Thank you very much for staying, and I hope you enjoyed the presentation. There will be parting gifts at the door, and I think they're in the form of eating in the lobby. Thank you.

[END OF TRANSCRIPTION]