HYDERABAD – Tech Day (Part 2)
Saturday, November 05, 2016 – 13:45 to 15:00 IST
ICANN57 | Hyderabad, India

EBERHARD LISSE: Before we start, can I ask the host to come to the podium already so that we have got the speakers?

Welcome, everybody, to the postprandial session. Of course, I must always punish the latecomers, and since they're not there, I'll punish the ones that are here.

The next presentation will be what I usually call the host presentation. It has been a tradition that we always invite the host for a presentation of their choice. Without further ado…

UNIDENTIFIED MALE: Good afternoon to all of you. Thank you for this opportunity for this host presentation. I'd like to mention C-DAC, the organization which I came from. C-DAC stands for Centre for Development of Advanced Computing. It's the premier research institution under the Ministry of Electronics and Information Technology of the Government of India.

We are fairly spread across the country at 11 locations. We work on entire gamut of IT technologies.

We are coming specifically from C-DAC-Bangalore. It's a small team. We work on [inaudible] cybersecurity research, cloud and HPC, language, computing, and ICT.

I specifically [inaudible] and Internet Engineering Group. We work on parameter security solutions, information security solutions, and Internet standards.

Here the Government of India, through us, has been trying to promote the development of Internet standards to increase the contribution of Internet standards in the country.

We have recently started an effort called Indian Internet Research and Engineering Forum, iiref.in. People can visit that. We award fellowships and try to improve the participation of the Indian community within the standard development process. So that's one of the activities that we do.

Here of specific interest, we have been associated with ICANN since 2013. We have members from ICANN contributing and trying to bring out the knowledge on DNS security to various participants.

These are the two important things which made us come here. With that brief background, we thought we can share some of our work which is being done.

Our agenda is basically to draw the synergies between TLS and TLSA. In India, we have the hierarchical trust model, where, if you want a digital certificate, whether it's an SSL certificate or a signature certificate or whatever – e-mail certificate – you need to approach a certifying authority. The certifying authority has to be licensed to be a root-certifying authority of the country, with is called a CCA.

If that is the case, then your certificate is legally valid. If there is any issue or attack happening, you can approach the court and you have some sort of remedial measure. But if you don't follow this part and you try to take a certificate from some other institution or organization which is not authorized with the root-certifying authority, then it's up to you to do it. This is the scenario which is prevailing in India.

But there are issues, as you know, that applications maintain their own trust store. For example, Mozilla has its own trust store. And applications like Adobe PDF have their own trust stores.

The thing is that whatever the root-certifying authority of India's, as well as their CA (certifying authority), should be loaded into the trust stores. When you consider users who are not experts or [inaudible] to these aspects, and if they visit a website which is having a certificate, which is actually licensed

by the root-certifying authority of India but is not loaded into the trust stores, it becomes an issue. The user may see a message from the browser saying that this particular website is not to be trusted. So he backs off.

So in such kind of scenarios, organizations move to other well-recognized certifying authorities, which can have a widespread acceptance. But legally, of course, they don't have any sanctity.

So this is a scenario which is prevailing here. Of course, I want to show the Board the positives and negatives of what is happening the process of TLS certificates and, of course, DNSSEC/DANE, which is TLSA.

TLS. Just a brief introduction because most of you would be aware of it. The structure of TLS is a very complex one. There is a handshake protocol which establishes the connection. It shares the keys, etc. There is a [record] which actually carries [inaudible]. And there is quite a big length of cypher-suites which are there.

Of course, TLS 1.3 is in the discussions in India. So it's in the TLS community, and the IETF is making it up. The objective is to clean up the latest version of TLS [on BIND 2], increase the security, and improve the performances.

But the main thing here is the certificates. Certificates are the key. Certificates which you obtain from certifying authority actually stands as the key for this.

But, again, of course the validation process of a certificate is quite a complex thing, including browsers, as well as applications do that.

Maybe if I want to distill the process as a [inaudible], this is here. Basically, you need check for the validity, the time, the CRL certificate [inaudible] certificate, etc. And you need to validate the signature which is present in the certificate. Please note there's a recursive algorithm, but we have just put up the basic steps here. So they issue us a certificate. If it's not self-signed certificate, then you need to continue. Again, you need to go back. So the entire chain of certificates need to be downloaded and it needs to be verified.

This is what typically happens in a certificate validation. Of course, you have to treat the root-certifying authority, which is generally a self-signed certificate, as a special case. That is step #4.

So that's the entire process which we have briefed up.

Of course, applications manage their own trust stores and come up with a set of pre-loaded certificates. In countries like India,

we have a separate trust chain. Those certificates don't come up with the applications that are pre-loaded.

But the user may be using such applications. For example, if I send you a digitally signed PDF document which has been issued by a licensed Indian certifying authority, it may be saying that it is not able to trust it. So now it is up to you to figure out which certificate is missing, and you need to add the certificate and see if you can trust it.

This is an issue which skips going between the application developers and the certifying authorities, but that is one side of the equation.

But of course, if you trust the certificate and a citizen of this country, you can explicitly add certificates of this domain, which you want to trust.

So this is what happens in the trust stores. Of course, this is a very brief overview. Now I'm jumping into the next topic of DNSSEC, which everyone, I think, in this room should be aware of. With DNSSEC, you have a secure connection the DNS server also.

Of course, DNSSEC [inaudible]. Every zone has two crypto key-pairs. This makes it a little complex, but of course I'll just run

through the steps. Operational keys: you have the zone-signing key as well as the key-signing key and delegation signers.

Of course, DNSSEC uses public key cryptography and digital signatures to provide data origin authentication and data integrity. It provides protection against spoofing of DNS data. That is, it tries to avoid corruption of the content. But it does not provide you the secrecy or confidentiality. It does not protect you against the DDoS attacks.

This is actually another slide which explains the processes involved in DNSSEC. This animation is available and in [inaudible] format, but unfortunately I can't show it here. It's in [inaudible] you can have the complete animation which is being there. So that's the thing.

Coming back to the next step, that is DANE, which completely based on DNSSEC. It allows the pinning of the TLS certificates into the DNSSEC zone, TLSA.

But there is one issue with a TLSA record. TLSA is basically one of the record types that is available. It is introduced and allows the pinning of the certificates. It allows a self-signed certificate or any arbitrary self-signed certificate, and there is an opinion floating in the community that it also provides the required security and one need not approach a hierarchical trust model.

But then there is a risk in it. That risk has been recently identified in a draft, which we discovered it on October 9th, 2016, which actually talks about an unknown key-share attack, where an attacker is having a secure connection to a server but none of them know that they are actually going through a man-in-the-middle. So that is a simulation which is being described in this RFC draft.

Of course, the main issues is that, in the TLSA, a validation step [must] be there, validating the server certificate, which it has received from the DNS.

But now we produce a self-signed certificate. There are four cases which are there. The last case is basically the risky one. That is, if you are using a self-signed certificate – that is, especially when the code is 03. There is one self-signed certificate which is shown there in the listing we have set up. So basically, the risk is, if you are using a self-signed certificate which has not been authorized by any CA, then you run the risk of attacks.

Right now, I think the attacks are in a discovery phase. One of attacks has been documented. Of course, some mitigation strategies also have been mentioned in this particular draft.

So these are the thing which have been coming up. We would like to conclude basically by saying that both TLS and TLSA are

required to establish the current level of communications. Most likely, if you have a hierarchical trust model, it will help you to have that safety feature.

Of course, there are complexities. There are issues in the validation of certificates, and there are attacks which happen, but that can be resolved. But this is one simple thing which can be done.

So though TLSA can stand independently, we suggest that it should avoid the use of self-signed certificates, but rely on certificates which are issued by a valid certifying authority.

So that's the part of it. Thank you. There are some references that are given here which can be looked into. Thank you.

EBERHARD LISSE: Thank you very much. Any questions?

All right. Give him a hand, please. Rick is the next presenter, so he can ask his question now or later.

RICK LAMB: Oh, okay. Excellent presentation. I'm just trying to make sure that I understood the driving force behind it. It sounds like you're saying that, for some of the trust models that you have here in India, these are not automatically included in Internet

Explorer, in Microsoft, and – is that true? So it's not there by default?

UNIDENTIFIED MALE: Yes. It's not there by default.

RICK LAMB: Okay. That makes this very interesting to me. This is an opportunity. I'm obviously a DNSSEC geek. I'm very excited about this. This makes it an excellent example where DANE and DNSSEC can actually maybe help expand this trust model.

So that's what I was just trying to understand. To me, that was subtle, but that was really cool. Thank you for that.

EBERHARD LISSE: Okay. If there's no other questions, thank you very much. Now we move straight to Rick Lamb, and then the other two presenters can start moving towards the podium.

RICK LAMB: Okay. All right. That was a perfect segue into what I'm going to talk about. I'm going to try to do a demo. Demos never go well, so I have some backup slides. If you pull down these slides, you will see screenshots of how it should work.

Let's first start. I'm assuming everyone understands DNSSEC in here. We had an excellent introduction already from the previous speaker. One of the things that we, as DNSSEC geeks, continue to look for is some sort of driving application for DNSSEC to help it get adopted more widely.

Of course, DANE is one of those, but what I often find is, "Okay. DANE is the right answer, but it's not running on my Windows machine." I know that's probably sacrilege to say I have a Windows machine, but there's still a very large percent of the population that uses Outlook and Windows.

So that's all this presentation is. If I can get through the demo, all I'm going to demo here is how TLS/TLSA certificate and SMIMEA certificates inside DNSSEC will allow me to seamlessly send encrypted e-mail by pulling down certificates from other users without a pre-exchange of certificates.

Right now, if anyone in here has played with S/MIME on Outlook or some of the other packages, typically you need to exchange a certificate first between someone, or exchange a piece of e-mail first – maybe signed e-mail – to get a copy of the other person's certificate before you can send an encrypted message to them.

For me, this has always been the Holy Grail. Forget about Snowden. Forget about NSA and all this stuff. I've always wanted end-to-end security. I've always wanted to be able to do that,

regardless of what I'm trying to protect against. So hopefully, with this demo, I'll be able to show this.

Now, I'm not the first one to try to do this. Dan Kaminsky in 2009 at a Defcon conference hacked something together that used DNSSEC and pushed some across semblance of certificates or fingerprints or something across the net. But I have yet to see an application or this built into Microsoft or something like that. And that is my goal.

The key here is that second-to-last bullet. The way I do this is, since Outlook already talks to something called LDAP, a directory service – an address book service; it's a standard protocol – I had somebody help me write an LDAP-to-DNSSEC convertor, a validating convertor. So it validates the lookups from DNS, and it converts it in and out of the DER.1-formatted LDAP messages, and all this on a Windows machine.

All right. Next. Pray. All right? Let's see. I'll try to share. How do I share? Oh, okay.

EBERHARD LISSE: It will be shared for you.

RICK LAMB: Shared for me. Okay. All right. This is all running under Linux, by the way – two Windows 10 VMs running under Linux. I'm only going to be able to show one VM at a time – one Windows thing for a time.

So I'm just going to try to send some e-mail. I'm going to start with the sending side. That's Outlook. I've already gotten a certificate because I didn't want to waste everyone's time. So I already have a certificate that I pulled down from Comodo, a free e-mail certificate. I have my crib notes here just to make sure I do things exactly the right way.

So I will try to create an e-mail. Okay. I can see there. All right. And I'm going to try to send it to another account that I have, Detesto1. Detesto3 is the source e-mail address. I'm not very creative. I apologize. Okay. Options: I'm going to make sure I want it to be encrypted e-mail. All right. And I'll just type something here, "Test encrypt e-mail." Go to the next line: "1, 2, 3. Can you read?"

Okay. It's very small. Does anyone know how to make this stuff bigger? Let me try control – this thing – no. So it's very small to read. If it's really that small, I can go to –

EBERHARD LISSE: Rick, carry on. We all have got Acrobat. We can all access Acrobat on the screen and see it on the laptops.

RICK LAMB: Okay. Because I can try to do the pre-canned presentation. Anyway, I'm going to try to send it and it's going to fail. Duh, right? I don't have the certificate for the destination e-mail address. So I cancel out of this. Then what I do is I say, "Okay. Let's look at the Outlook address book." So I'll switch the address book and I will try to add an address book, LDAP. The convertor application is actually just going to sit on the same machine, 1.27.00.390. 390 is the port number. I got to type that in. 390. Okay.

All right. Done. It's telling me I have to reboot, so I'm going to finish or restart Outlook. So I'll do that. All right. Close Outlook. But while I do that I have to also install this little application that's going to do the translation between LDAP and making DNS lookups.

Let's see if I can find that. All right. I'm not a Windows programmer, but there it is. It's a little program. It actually sits there and runs in the systems tray. All right.

Just to show you things are working, I will also kick up – and this I can make bigger – the log file, or look at the tail of the log file.

All right. Let's see if I can make that any bigger here. Yes. Control plus, plus, plus. All right. This is just to prove to you that I don't have any cards up my hand and I'm not trying to fool you.

All right. Now I will try to execute Outlook again, and I'll try to do the same exercise. Let's see. I'll pick this one e-mail account that I have here. Pray, once again, to the demo gods because this is just ridiculous that I'm hoping this is going to work.

All right. "Encrypt Detesto1." Test e-mail. "Can you read this?" All right. Then I'm going to try to send it and cross my fingers for a while.

It seems to have gone out. Now we need to look at the other side to see if this worked. Okay. To do that, I am going to stop sharing this copy of Windows, go to another copy of Windows I have running, and start sharing that screen. This is my ICANN e-mail account, so please don't read all the angry e-mails I get from people. I'm not popular there.

Okay. And there it is. Yeah, there's one [below], but there's the date for that. "Test e-mail." This is what I typed. You may not be able to see it, but "Test e-mail." There's a lock there that says it's encrypted. All right. So it did work. So I was able to send an encrypted e-mail across the net without any user intervention.

If this was set up at a site – for example, the U.S. Department of Defense has 3.2 million employees that have a smart card. Every one of them is trained to use S/MIME and Outlook. I worked in a government, too. There's some very, very large segments of the population that are all ready to go. S/MIME is not new. It's been around for a while. Outlook is not new. It's been around for a while.

But with this sort of a plug-in, if you want to call it that – I don't want to call it a plug-in – this little executable running, and DANE and DNSSEC, they will be able to exchange e-mail with anybody.

Let's look at the certificate. What's really in the DNS? We're networking guys. We want to see DNS. This was some Microsoft mumbo-jumbo, so let me exit out of that and stop sharing that.

Thank you, Kim. [This actually] is working. I am just really amazed. I really didn't think I could switch between two different Windows sessions like this.

All right. For this, I should be able to get a much bigger screen. Anyway, there's a log file. You can see just some coding there that's trying to break down the DER ASN.1 structures and doing validation on that and actually coming back and providing responses. So we're done with that.

Okay. What does the record for this thing look like? I obviously already have it pulled up here. Let' see. I'll try to make that as big as I can because I just know that that is just impossible to figure out. Type 53 – I think that's – what is that? Is that TLSA type or the SMIMEA type? TLSA. All right. So TLSA type. And this is, according to the current draft, [is] still a draft RFC for the S/MIME-DANE-DNSSEC protocol.

If I look that up on Google, full certificate. I store the full certificate. So here's the question. Let's look at the first part of the answer. Here's the answer. For you super-geeks in this space that follow the TLSA stuff, you could see this is Type 03. I think that's Type 03.

Anyway, there's the whole response. So that's what's in there. If you want to try something like this, I was able to convince the kind IT staff at ICANN to set me up – so richard.lamb@ICANN.org actually has this in there – so you can send me directly encrypted e-mail if you wanted to.

So that's the first part of my demo. I have another really short part, must shorter than this, that I'll describe.

Following me so far? This is really simple. The idea has to be simple. My goal in this thing is I want Microsoft or somebody just to take this and make it part of their operating system. Then it just goes away. It's just there.

One of the LDAP address books – this would be [an option]. "You want this special address book?" Click yes. "You want the DNSSEC address book?" Click it. Something like that.

Let's try one other thing here. I do apologize for not being visible, Robert. It never occurred to me that this is so small. It is, and I hate when I'm sitting there and I can't read this stuff. It just –

UNDIENTIFIED MALE: [inaudible]

RICK LAMB: Yeah. And you're blind, too, like me. Right? Anyway. All right. I also have this little test setup that I have. Please don't smash on it right away, as soon as I do this. Give me a chance to get my request in first. If you send e-mail to an address – and these are all in the end, in the reference section. Smimea@zx.com. That's one of my domains. I just send a test to that. "Test message. Test. 1, 2, 3." I don't sign it. It's not encrypted or signed or anything. Just a test message. "1, 2, 3." Carriage return.

I send it. I wait. See if something comes back. Come on. Okay. Stuff's coming back. All right. So two messages come back if you send a message. Anybody can do this. It's just script that I put at the other end, my e-mail listener. What you get back from this,

actually – ah, I can't make that bigger, either. Anyway, so what you get back from that is a response that tells you the state of your SMIMEA setup. In this case, I sent it from an e-mail account that actually has SMIMEA records in it. It comes back and says, "Here's what the SMIMEA record looks like and what type it is." It gives you some other information – serial number, the certificate, the fingerprint, and a few other things.

If, for example, you have this information already in the DNS, your certificate, it tries to send you an encrypted e-mail. And it does, and it says yes. "If you can read this, congratulations. Your setup is working."

So that's it. I have here now in my notes, "Contemplate response." Okay, fine. I'll stop sharing. Can I go back to my presentation?

Okay. That actually worked. I'm amazed. So if you get this presentation, you'll have all this in a much more visible form. That's all what these slides are. It goes through every one of these steps because I wanted belt and suspenders here, just in case something failed.

Anyway, what happened? Ugh – don't get old. I'm going to switch glasses. So what happened? Outlook asked the address book, the LDAP, for the certificate for Detesto1. It did this by connecting to port 127, the local loopback address of port 390 – I

just picked that out of a hat – and then this program, LVDT.EXE, which you can download – it's free; it's in beta but it's free – is this really minimal, completely-from-scratch LDAP server that converts LDAP things into DNS lookups.

The response is, of course, a converted back – and, of course, validated. Why is this important? LDAP to DNS is nothing new. LDAP to DNSSEC means that I am now validating at my end point what the response is. So it validates the response, converts it back to an LDAP response, sends it to Outlook, and voila. Everything works.

Outlook is still checking the certificate that comes back to see if it is valid in their root certificate store. You can't just pick a self-signed certificate here. There's some tricks you can do, but that's for another class.

So that's it. There's the resources for it. Lvdt.dc.org actually has that executable, if you want to download it. If you want me to verify the hash for it, I will for you. The other one – smimea@zx.com – is that test place where you can send e-mail.

Of course, I couldn't do anything of this without the draft that Jakob Schlyter and Paul Hoffman wrote – the draft at [IETF-DANE-SMIME.] I don't know what state that's in yet. I don't know if that turned into an RFC yet or not. Because there were some discussions for a while.

Anyway – oh. That's it. Any questions?

EBERHARD LISSE: Warren –

RICK LAMB: Tomatoes?

EBERHARD LISSE: Warren?

WARREN KUMARI: No. Warren Kumari, Google. Just speaking to your last point, no. Unfortunately, that's not an RFC yet. To some extent, that's my fault and Oliver's fault. We did the working group last call. We didn't get enough comments. We should have pushed harder to get more comments. We're going to be starting another IETF working group last call as soon as the IETF starts and asking people to please comment. So it sounds like you care. Please comment on the draft when the working group last call is open again.

RICK LAMB: Okay. Thank you. I love it. Any other comments? Am I smoking dope? To me, it seems okay. Have I missed anything here? Okay. All right – oh, no. Good, good. Because I feel like this is a "duh."

WES HARDAKER: No. Rick –

EBERHARD LISSE: Sorry. Identify yourself for the remote audience.

WES HARDARKER: Thank you. Wes Hardaker, USC ISI. Rick, you know I love this stuff, right?

RICK LAMB: Oh, I know.

WES HARDAKER: This is the magic bullet-type stuff.

RICK LAMB: Yeah, but do you love me?

WES HARDAKER: I do. I do love you. The coolest thing about this is the first demo shows there's some complexity and we saw some UI issues, but the fact that you got it working so easily with those hacks is just awesome.

I actually loved your e-mail address even better. That's something you can advertise to people. "Here's how it can help you out. Just send an e-mail to this and you can get it started." If you chain that to show people, "Okay. It looks like you're using this mail reader. Go create an S/SMIME certificate this way," and then walk then through the steps –

RICK LAMB: I could parse the header, yeah.

WES HARDAKER: Yeah. And then figure how to tell them to go add their record to their zone. That's going to be the hardest step, right? Go to your registrar and put in a new Type 53 code.

RICK LAMB: Yeah. But we have tools, right? You've written tools. I think Russ has.

WES HARDAKER: Right. Some people edit their own zone files, but I'm weird and a geek, right? Right. So this is a great step. Thank you for doing this.

RICK LAMB: Okay. I was waiting for the "but." All right. Okay.

WARREN KUMARI: Sorry. Warren Kumari again, Google. And, yeah, I also love this stuff, but, no, I don't love you. Sorry.

I'm not sure if you mentioned all of them, but NIST currently has an open comment period on this thing. It's 1800-6 on basically doing this. They would like comments on if people think it's a good idea. Basically, they've suggested it; please provide feedback.

RICK LAMB: Okay.

WARREN KUMARI: The German Federal Information Processing Group or something –

RICK LAMB: [VSI]?

WARREN KUMARI: Yeah. BD – yeah.

RICK LAMB: Something like that.

WARREN KUMARI: Something like that. They are actually recommending that people do DANE-encrypted mail, and I hear recently that so is the equivalent thing in the Netherlands that's also suggesting it. So a lot of governments are actually now saying it's a good idea, so maybe people do it.

RICK LAMB: Thank you. All right. Well, thank you very much, guys.

EBERHARD LISSE: Can I ask a very dumb question?

RICK LAMB: Yeah.

EBERHARD LISSE: How do I do this with Apple Mail?

RICK LAMB: How do you do what?

EBERHARD LISSE: How do I do this with Apple Mail?

RICK LAMB: Well, you know, I'm just not a Mac user. I don't use Mac because I need to feel the pain. If I don't feel the pain of the masses, then I will not find the solution.

EBERHARD LISSE: My understanding is that ICANN staff will get Macs if there is a need for it. Do we have a need now?

RICK LAMB: Yeah, I guess so. Yeah, all right. But more to your answer, Eberhard – I'm sorry to be a little flippant about it – I'm trying to make this work right now with Outlook under a Mac. The next step will be to try to make it work with Apple Mail because I understand the LDAP interface for address books is not unique to Outlook.

But, yeah, I need to set up a Mac and have a test lab for this. I will do that. That's my next to-do. Thank you.

EBERHARD LISSE: Okay. As I said, at Copenhagen you can tell us if you have anything done by then.

Diego?

LUIS ESPINOZA: Quick question. Why are you using your own LDAP server? This can be implemented in a [center] – an open LDAP, for example?

RICK LAMB: Implemented in what?

LUIS ESPINOZA: Open LDAP or something like that –

RICK LAMB: No, I wrote it from scratch. I write everything from scratch. I wrote my own ASN compilers and decompilers.

LUIS ESPINOZA: Why?

RICK LAMB: Why?

LUIS ESPINOZA: Yeah.

RICK LAMB: Because it's much easier for me. It's much easier for me to do my own stuff than understand someone else's stuff.

No, no. I'm just telling you. That's just the way I code. And it's the way these other guys code – my Russian friends.

LUIS ESPINOZA: My question is, could it be implemented in a standard LDAP server?

RICK LAMB: I'm sure this could be done all like that as well. Windows is strange, right? When you write something in Windows and it has to run quickly, it's this multi-threaded IO completion-based – it's a very different architecture. It just didn't lend itself to that.

Anyway, yeah. It's a good point.

EBERHARD LISSE: Okie-doke. Can we all give him a hand? Thank you very much.

Now, Diego Espinoza is going to take us through the source code of Mirai. As we all know, we were having a bit of an issue two or three weeks ago, which even made the press. Dyn got seriously under a DDoS attack. We asked them to present. They didn't have somebody scheduled to come to Hyderabad, and it was a bit of a short notice to convince them that we would have a budget for this.

It's also short notice, so if we do it in Copenhagen, which we are confident they will do, it gives them more time to do deeper analysis. That has also led us to think about whether we should not have a session focused on DDoS with a roundtable and everything. So if anybody has any interesting projects or presentation in mind for Copenhagen, please stay in teach.

Now, Diego, please.

LUIS ESPINOZA: Hello. My name is Luis Espinoza. I'm working for Akamai, but this little work is more for fun or as a hobby than from Akamai's work.

Who in the audience knows what is Mirai? Okay. Mirai is not a [publisher]. It's a program. Just in case, my wife is here and I need to convince her it's not [inaudible].

Okay. Sorry for the size of the font, but you can access this list of user names and passwords used by Mirai. Mirai is a software. It's a botnet created to attack last week the [.in] server. But after analyzing the software, they could do many other things with this.

They use a list of hardcoded user names and passwords from devices. Let me show you something. Well, I will talk a little bit more about Mirai.

This time happened something very particular. In some way, the hacker or somebody else published the source code of the botnet and the controller on GitHub. You can go there and download the source code and analyze the source code and see what it does. On the GitHub, you can read a disclaimer, "This is for academic purpose." But if you checked GitHub you will see a lot of forks for this code. Then we can expect this code as a base for more nasty things in the next weeks.

What is special about this software? This software infects devices like security DVRs, like cameras, like routers, small devices like [inaudible] sitting in laps. These are small devices. It can affect anything that runs BusyBox. BusyBox is a special Linux version used in these devices.

The strategy was infect these kind of devices – not a computer – and use these devices as attackers, creating a huge botnet.

Within the code, there's some interesting lines. I will show you here and talk about these. But this is not near a deep analysis of the software. It's just some highlights of things I found interesting.

In this [inaudible] it is included with these lines. This line set runs a comment, like /bin/busybox with ECCHI. It depends on the answer of the device. The hacker knows the device could be vulnerable to the attack or not. If the device responds with, "Applet not found," then it's not vulnerable. But if the device responds with a help menu, the device could be vulnerable to this attack.

In other parts of the loader – the loader is one of the programs – this is very nice. This program tries to use Wget to download the software. It tries to use, I think, http, but if none of those things work, it'll use a very simple thing. It's creating a binary using and echo comment. Then these [barriers] will install the botnet inside this little box, using the echo comment of the ASCII or hexadecimal codes, and create the program inside the device.

Once the program is infected, this is the [competence] of the bot by itself. The bot is the subordinate program that will work as a command from the hacker. This is a very simple program. This is not too long to read. This is a very small program.

Some of the codes we'll check here. This is the DLR. DLR is the payload for the particular processor which is running in the box. This small box doesn't use Intel or Pentium or something like that. They are using very cheap and small processors, like arm or m68k or mips or something like that. The program has one payload for one of each of these processors. In this way, it can be loaded to the device instantly.

In the source code, these DLRs are binary. I didn't take my chance to try to debug these binaries. It's not that easy to do. But it's a payload for that particular process to infect that device.

The other interesting thing here is – this is very bad; the size of the font is too small – in these lines of this resolv.c, there is a CNC domain and is the name of the controller for the botnet.

We'll roll back a little bit. A botnet works like this. You have several nodes. Each node has a small program that it reports itself with a controller. Every node of this will report to the controller, and the controller will send comments to these devices to generate attacks in a massive way.

When this program is [installed] to the node, the node must know how to get to the controller. At this part, the programmer leaves the space. What he's saying in this first line is,

"cnc.changeme.com." This means you need to put there the DNS name of the controller for this botnet.

The real value of this name is obfuscated. It's not encrypted, but it's obfuscated. But the end? You can reproduce that using some [inaudible] comments or something like that.

The next line says the part number of who is – this botnet will communicate will communicate with the controller. In this case, we'll use port 23. Port 23 is very well-known as a Telnet port.

The second part is a kind of reporter. Once the botnet is installed in the device, it will scan the network to try to infect other devices, too. At the moment when it is used for attack, the results of the attack will be reported to the CNC, the controller, using this other port. In this case, it's something like 48101, the port.

The other thing hardcoded in this botnet is very interesting. It is using the DNS server 8.8.8.8. I'm not so sure, but probably if you [just block] the access to 8.8.8.8, you probably cannot resolve the name of CNC, and then the botnet is useless. But I'm not sure. It could be.

Another interesting part here is the main.c part of the program. It prevents some components of the operating system [in] the bot – or in device – can kill the barriers itself. Then it would

prevent the watchdog. They reboot the system to prevent the barriers getting inside. In that way, it will make sure that the device will not reboot it and it will keep inside the device.

This is very bad. I'm sorry for the font size. I didn't figure out that the font was too small. Well, any of you can access the Mirai code and GitHub and see these lines by yourself, but the thing here is, as a reference, this hacker is using a very well-known user name and password of devices.

Okay, maybe you can see some of them here. You can see the fifth line – this group. The password is 8888888. Maybe some of you or many of you have been using some of these Chinese DVRs, security cameras. This is the default password for these DVRs, 8888. It's very easy.

UNIDENTIFIED MALE: [inaudible] China.

LUIS ESPINOZA: Yeah. There's a lot of these devices installed in many places because it's very cheap. These are very cheap DVRs. They are using these passwords for the IP cameras, too. You can see something like admin/admin or root/admin and root/123456. These are all passwords for these devices.

There's a list of these. It's very common that you find many people don't change these default passwords, and you can access easily, with these passwords, these devices.

Okay. Once the botnet is installed in the device, the hacker can send a comment to the botnet and generate an attack to a specific IP set of addresses, but it has an exception – yes, thank you. If the botnet is about to attack these IPs, some of these IPs are loopback, for example, or an invalid address, or an internal network, but some of them are real networks; for example, the Hewlett-Packard company or the U.S. Postal Service network, or the [IANA net] [inaudible]. But these are exceptions in the IP addresses that the hacker can use to attack the system. This is interesting.

Another part of the software, Mirai, is the CNC. The CNC is command and control. This is the main system that is controlling each one of the devices hacked and ready for the bot. These botnets are a kind of service within the hacker community that you can rent or use. It depends on the size of this botnet. If the botnet has something like, I don't know, maybe 5,000 devices or 100,000 devices, you pay more or less money to use that. The system is available for use with different objectives.

The CNC is written in go language. It's very similar to C, but it's not C. It's more simple. It has some very clear code, very well-documented code. You can see a few things it does.

This is the main message of the control system. It can report if the connection was hijacked from the device, or it can report if the masking connection is hiding from netstat. It has this kind of very simple message, letting the hacker know how it's working, this botnet system.

I think this will read a little bit better – yeah. Okay. Thank you. Attack.go is a part of this component. It receives a parameter, the name of the attack. You can see the name of the attack. In this case, it could be udp, vse, dns, syn, ack, stomp, greip IP, greeth Internet, udpplain, or http.

This botnet system allows all these kinds of attacks. In the case of last week, it was DNS only. But this botnet can be used for any of these attacks, too.

The other part I'm [inaudible] this program is the list of the targets. The list of the targets is in the code is like that, with examples. It can be an IP address. It can be a net block, or it can be a range of addresses.

The other thing you must type as a parameter for this botnet is the duration of the attack. It could be from 0 to 3,600 seconds, something like one hour. It's the maximum. It's limited to that.

There's some flags. These flags are for the packets that will be used during an attack. The lens of the packet, the rand, the tos. These kind of headers of the packet could be modified during the attack by the hacker.

Okay. The main.go. The CNC uses MySQL database to store all the botnets reported to the system and to store the status of the system and to store the status of the attack, too. This is the username and password for the database. It's very simple: root/password. The name of the table is Mirai. It is stored locally within the control system.

It will listen on port 23. Do you remember a few slides ago I showed you that the bot will try to connect using port 23 to the CNC? Well, here the CNC will listen on port 23 and will listen on port 101 too. But in port 101 there'll be the api, not the system by itself – the api, sorry.

With this piece of code, it's easy to read how the system will work during the attack, how the bots will report to the controller system, and which ports will be listened to. And of course, the database. If you found some control system running on some computer, you can check the database easily with this username

and password. The hacker implementing the system could change these resource parameters.

The other part is there is a scan report system, listed on port 48101. This port is for [side] reports to avoid the collision of many reporters at the same time.

The thing here is there's a game-changer, as somebody said yesterday. This attack is an example, and there'll be more attacks like this. The game-changer here is they are no longer using computers infected with botnets. They have started using things connected to the Internet.

According to Intel and some researchers, by 2020 there'll be around 26 smart objects by every human. That is a lot of devices. Then we can expect that this kind of massive attacks will increase for sure because we are not talking about 100,000 computers or 200,000 computers. We are talking about millions of devices. These devices could be the router in your home, the security camera in your home, the refrigerator.

This morning, I read from Indian Times that there's a flaw in this Philips LED smart lights. Hackers can easily control the lights using this BG protocol. It's a wireless protocol used by Philips to control the lights when you change the color or dim the light. Apparently, it's easy to hack the bulbs.

You can think, “Well, why is it useful to hack a bulb?” Well, in each of these LED bulb lights, there is a small computer. These small computers can be used for attacks, too. Each of these small computers doesn’t have too much power, but if you talk about millions of these devices, then it’s serious.

This research about these lights, LED bulbs, they use a car with software running in the car. They run it from the street, and they start hacking bulbs in the houses. They can hack something like 15,000 bulbs easily. They could change the lights or annoy people with the bulbs from other places, but eventually they can use these bulbs to attack systems, too.

I think what we saw last week with the Dyn attack will get worse. We are talking about, I think, exponential dangers in these times because we are talking about devices connected to the Internet, not only computers.

That’s it. Comments? Not questions. Comments.

EBERHARD LISSE: Thank you very much. I think this is a fascinating problem. Any comments?

WES HARDAKER: Thanks very much because I've been meaning to dive into that code and look at it, and you've just saved me hours of time by explaining it all to me very quickly. Thumbs up.

One real quick question. Do you have any idea why all the strings are obfuscated, why they went through the process of converting them to hex as opposed to putting the host name directly in it? That seems like a strange thing to do.

LUIS ESPINOZA: No idea, really.

WES HARDAKER: That's what I expected, but I just thought it was strange. If it's not encoded, why?

LUIS ESPINOZA: Yeah. But what I've seen in the past is that many of things done by hackers are more funny things. It could be a joke between them in some way. Yeah, no idea. Yes, it's weird.

The simple thing is they published the source code. It's weird to me. It's very weird.

Any other comments?

EBERHARD LISSE: Okay. Thank you very much. Let's give him a big hand.

So before the coffee break, Jaromir Talir will speak about DNSSEC automation in FRED.

JAROMIR TALIR: Good afternoon, everybody. My name is Jaromir Talir. I work for CZ.NIC, which is a ccTLD registry for .cz, a TLD for the Czech Republic, which is an awful small country in Europe. [Christian] mentioned you can locate the Netherlands as a small country next to Germany. The Czech Republic is another small country on the other side of Germany. So Germany is quite a good location point in Europe.

We have a lot in common with SIDN, that we are also doing a lot of development. One of the systems that we develop is an open-source registry called FRED. Today I would like to talk about how far we got when we were trying to integrate some sort of DNSSEC automation in FRED.

To be honest, we didn't get so far, or a far as I expected. So currently we don't have any service running for the registrants and .cz. We only have some sort of prototype that we have to work with a little bit more. However, we still have some experiences that we got that I could possibly share.

In my presentation, I will quickly mention what I mean by DNSSEC automation and how these things can be implemented in FRED and where the obstacles will be faced are that we will have to solve in the future and what we all do.

So we've already talked a lot about DNSSEC. We know that, with this technology, we got quite a good security mechanism. However, it was not for free. It brought some more effort for the sysadmins because they now need to take care about some particular things that happened with the zone file. Particularly, it is a necessity of resigning the zone file because of expiring signatures, and also, at some point in time, to roll the keys, which is a good security practice.

For both of these issues, we already have quite good tools that we can use. You may know OpenDNSSEC or BIND, which have integral signing, automated signing. Our [inaudible] DNS server, KnotDNS, also already has automated signing.

However, there is still some manual effort when you are, let's say, a DNS operator or the person that wants to have complete control over your system and you are signing your zone file locally. When you do KSK rollover, you have to typically log into your registrar system, fill some sort of form, where you will fill your key material. And after some time, when you will change

the key in your signing system, you have to again log into your registrar and remove the old key.

To solve this issue, [Goran] here wrote an RFC two years ago which is trying to solve this by introducing two new resource records, CDS and CDNSKEY, which contains a new key that you would like to publish. It is expected that some entity – registry, registrar – will [poll] for these resource records and they will take the information from DNS – verified by DNSSEC, of course – and change this key material in the registry themselves.

However, if you look around how this RFC is supported in tools, it's not as good as it should be or as it could be. As far as I know, for OpenDNSSEC, there exists a branch that is not maintained right now. I don't think this branch has been merged into new versions. So hopefully maybe this will change with the new OpenDNSSEC 2.0 branch or a new version, that they will integrate it.

With our signing tool, KnotDNS, we already started to work on that. Hopefully during the next year, we will have it published for the public.

The most promising development that has happened was with BIND. Right about a month ago, ISC released a new version, 9.11, where they introduced a new automation tool called DNSSEC Key Manager. This is something which is similar to OpenDNSSEC,

that you specify a DNSSEC policy, what algorithms you will use, how often the key will rotate, and things like this.

It was supposed to have a full CDS/CDNSKEY support. However, at the end, it's supported only partially. It's not possible to specify when the new CDS and CDNSKEY resource records are introduced. But it's supported in the DNSSEC settime utility that can be used to schedule when, for these keys, these new records will be introduced into the zone file.

With the combination of these tools, you can play a little bit with that, write some script, and then you have the tool that will properly insert these new resource records into the zone file.

I wonder if maybe any of you know about any other tool that is using these new resource records. I would like to know that. But for now, at least we have something that people probably in the future will start to use. So now it's probably time to look at it and try to support this in the registries or the registrar systems.

Going back to the registry, I have already mentioned that FRED is our open-source registry that we are developing since 2006, so we have now a ten-year celebration. It's already used by many other countries. There are eight countries that are already using this software.

In this year, there are new deployments in Malawi and Argentina. In Argentina, they have about half million domains, so this is the second biggest registry after .cz that is using this system, which is for us a good reference.

The other news for this year is that we have implemented RDAP. We also run this RDAP server for .cz, surprisingly. The RDAP server for .cz is still the only RDAP registry in the IANA registry.

So are now working with the other countries that already deployed FRED to upgrade to the most recent versions. Hopefully, maybe during next year, there'll be eight more RDAP servers in the IANA registry.

Back to DNSSEC, we have quite a different approach to DNSSEC than in a regular EPP registry. We have tried to implement the concept of key sharing in the registry, similar to how we did with the name servers sharing. For name servers, we have a new object called Name Server Set, which holds a group of name servers. This object can be shared among multiple domains.

So we use the same concept for DNSSEC, and we introduced a new object called KeySet as a collection of DNSKEY resource records. It's a first-class object, so a registrar can register this object. There's a dedicated registrar responsible for that object. Via EPP, registrars [inaudible] is the content of the objects.

It also means that the registrars don't upload DS records but DNSKEY records. The DS records are counted automatically. There's a possibly to configure the registry if you want to use the [Shell1] or [Shell2] algorithms for DS record generation. So this is quite specific for FRED.

How did we add the automation? We created a prototype script written in Python because our reference FRED EPP client is written in Python and it's also a Python library. Together with DNS Python, we created a script that will get CDNSKEY information from the [inaudible] DNS via DNSSEC validating resolver and call update_keyset via EPP.

Now we got to the interesting point. How do we deploy this? We actually found out that there are three scenarios for how we can deploy this. Those scenarios are either the registry, as us, will modify the objects that the registrars [found] the keysets of the registrants. The second is that we will convince the registrars that they will deploy this service and they will be responsible for their keysets. The slightly modified version or the combination, that we will become the registrar for keysets. We will allow the keyset [server] registry. We will be a registrar for keysets.

Going through all of these three scenarios more deeply, in our registration rules, we have a condition that only the registrar of that particular object can modify the data of that object. So the

first deployment scenario actually means that we need to change the registration rules, and we have to actively touch the data of the registrars, which is something that some people don't like. But why not? The registration rules definitely may be changed, but it's quite a long-time process.

For the second deployment scenario, it's definitely easier for registrars to do that. They could have even done that already if they knew it was possible. Definitely, if we all prepare some tool for them that they will just put into cron, maybe it's easier for them. We have the regular meeting with the registrars, so we will reschedule this topic on the next regular meeting to discuss this with them.

Within informal discussions, some of them stated that, yeah, maybe they could deploy this. Maybe they see it as a sort of competitive advantage. They are trying to search for services that they could offer to registrants.

The third solution is that we could potentially allow the transfer of keysets to us, to registries; that we would come the registrar for those objects for automatically managed keysets, which is possible because the architecture allows the transfer of the keysets from one registrar to the other.

We have experience with that because we are running some identity service called mojeID, which is nothing more than a

registrar for validated contacts. We are, for some contacts that have the process of validation, we are the registrar. So we have experience to be a registrar. The cost for us can be minimized if these keysets would be automated. So we don't expect that people will reach us very often.

There's a small bootstrap issue for all these rules. If the keyset is going to be assigned to the domain, only the registrar of the domain can do that. So this is something that we will have probably to solve.

So from those three scenarios, we will have to select one of them or a subset of them.

Here's just a recapitulation of some obstacles that we have found out during this process of analyzing. I mentioned that we support the sharing of keysets, which for some things is a good thing. But for this, it brought us to the issue of, "What do we do if there is 1,000 domains sharing the keyset and we see the CDNSKEY in one of those domains?" The general answer is, "Let's wait until all domains have the same CDNSKEY."

The other option that we thought about is that maybe there will be some signaling domain that will be in the keyset, and when this domain signals there is a CDNSKEY record, we will rotate the key.

The other obstacle that we have seen is that there is a bootstrap issue right now. It's probably not feasible to check all the million domains that we have to look periodically for these records. So we expect that people create the keysets manually.

The other issue is the timing, that we are actually generating the zone file every half an hour. So when we will manage to check all the automated keysets in one half an hour, it's okay. But maybe when this will be more popular, we will have to find some other solution for how to trigger this key rotation.

Just as a recapitulation, where we are working now is that we have three scenarios. All of them look viable. Maybe after discussion with registrars, we will reduce this number of scenarios. We have the internal prototype that is almost finished that we are running for some domains that are trying to change to generate, let's say, every day a new key.

Some of the issues that we have touched can be resolved by implementation of the draft that Jacques Latour will talk about, maybe in about two hours. He will probably explain it in a little more detail. But it's actually possible to trigger this change, not via [polling] the DNS but by calling some URL. After calling that URL, the process will start automatically.

So these are the issues that we have. The current state brought us into a lot of questions. I hope that this not the end, that

maybe during the end of the year or at the beginning of the next year we will continue and will bring some service for our registrants.

Thank you. If you have some questions, feel free to ask.

EBERHARD LISSE: Thank you very much. I'm not really going to entertain questions, not because I don't want to, but I've been informed that we may lose power during the break. So maybe we go and have a coffee now. If they lose us the power, they can do it while we're not doing it.

Thank you very much.

Okay. I've just been informed that they will just switch from generator power to house power, and that means they have to shut the lights down. So don't get surprised.

[END OF TRANSCRIPTION]