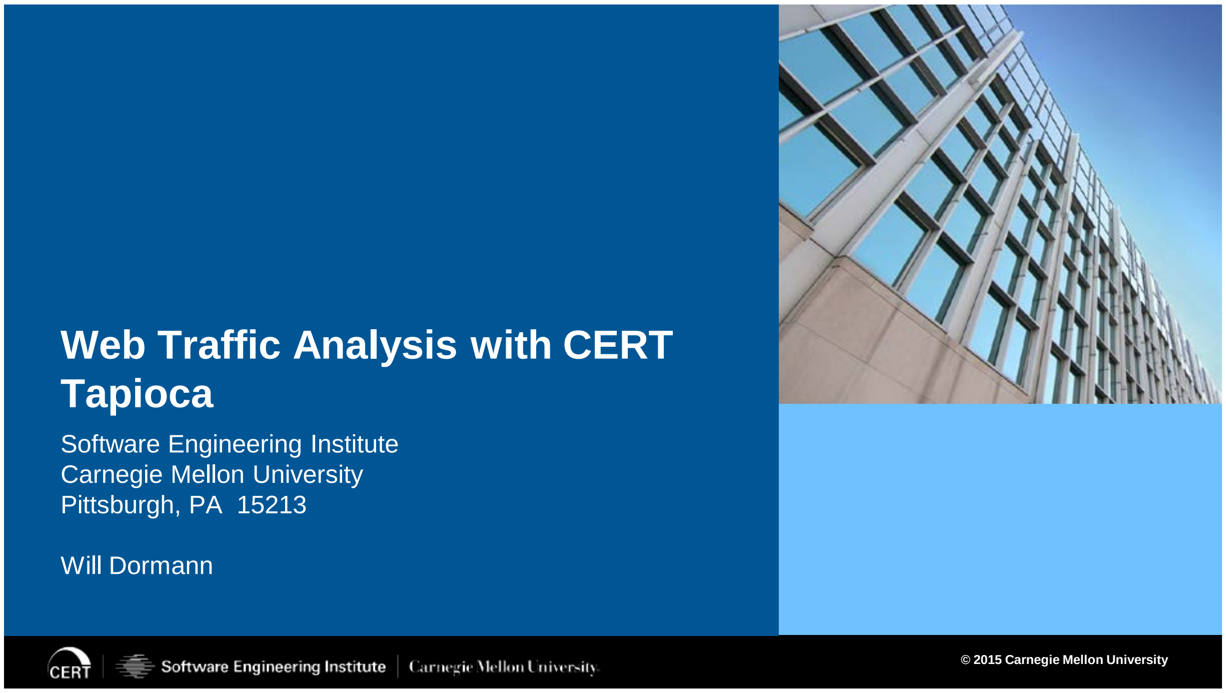
Web Traffic Observation with CERT Tapioca
Will Dormann

Table of Contents



Web Traffic Analysis with CERT Tapioca

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

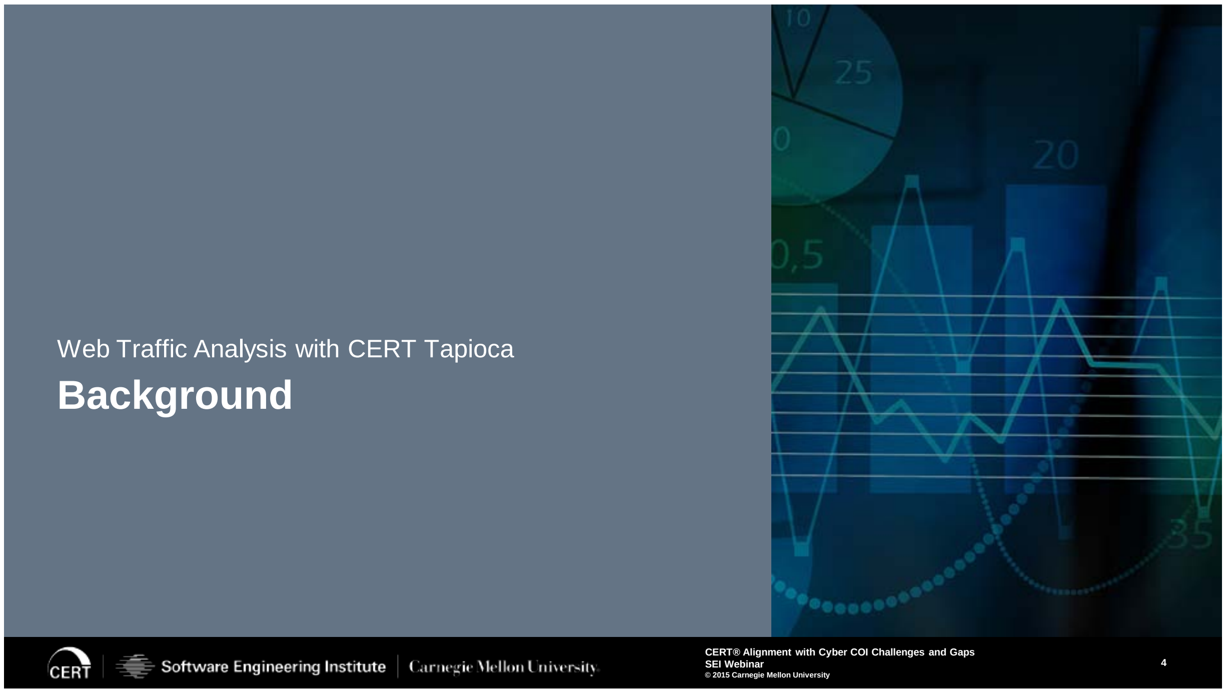
Will Dormann

**003 Shane: So we're going to welcome in Will Dormann, and Will's topic is going to be web traffic observation with CERT Tapioca. Will Dormann has been a software vulnerability analyst with the CERT Coordination Center, or CERT CC, since 2004. His focus areas include web browser technologies, ActiveX, and fuzzing. Will has discovered thousands of vulnerabilities using a variety of tools and techniques.

And before I turn it over to Will, I just want to remind everybody that's logging in and out throughout the day, we have a files tab that's available in your console. You can download the presentation slides along with other CERT work in the area of cybersecurity that you can walk away with today. If you're only enjoying us for one or two presentations, you'll see a survey tab. We request that you fill that out as well, as your feedback is always greatly appreciated. So we have Will queued up. We're going to turn it over to Will. Will, welcome. All yours.

Will Dormann: Thanks, Shane. I just wanted to talk a little bit about a project that I've been working on and a tool that I've released called CERT Tapioca, and basically this is an appliance that can be used to analyze web traffic.

Web Traffic Analysis with CERT Tapioca



Web Traffic Analysis with CERT Tapioca

Background

Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

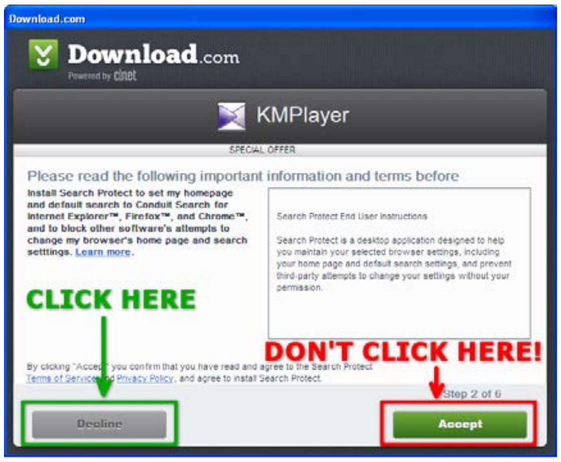
4

**004 But before I get into the details about the tool itself, I just wanted to give a little bit of detail as to how I came across the need for this particular type of tool.

History

History

Download.com



Download.com

Download.com
Powered by cnet

KMPlayer

SPECIAL OFFER

Please read the following important information and terms before

Install Search Protect to set my homepage and default search to Conduit Search for Internet Explorer™, Firefox™, and Chrome™, and to block other software's attempts to change my browser's home page and search setttings. Learn more.

Search Protect End User Instructions

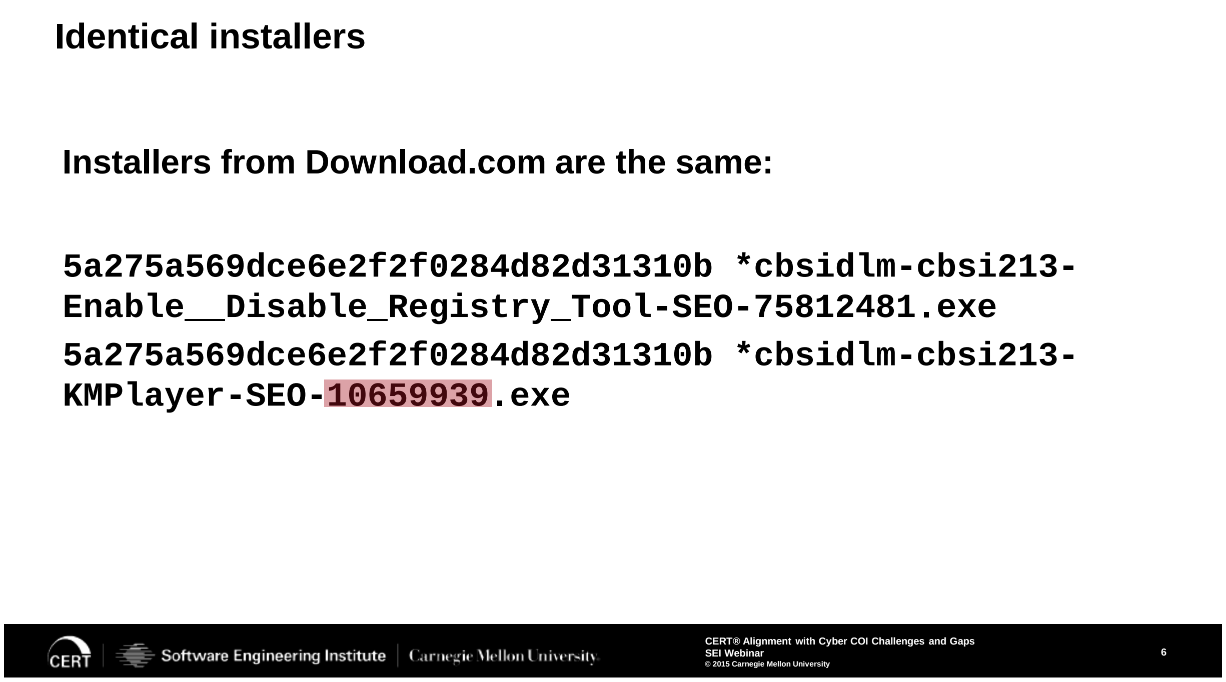
Search Protect is a desktop application designed to help you maintain your selected browser settings, including your home page and default search settings, and prevent third-party attempts to change your settings without your permission.

By clicking "Accept" you confirm that you have read and agree to the Search Protect Terms of Service and Privacy Policy, and agree to install Search Protect.

http://www.cert.org/blogs/certcc/post.cfm?EntryID=199

**005 So it was a year or so ago. I was looking into a particular technology. Basically it's the concept of bundled software, and the way that I started looking into that is I was working with a virtual machine. I needed to download some software, and I just did a quick internet search for something, and it turned out it didn't actually get me to the site that had the software, but it was actually another site that was providing it for me. And I started-- I actually did a blog post talking about the risks of bundled software. In this particular case, I've got a screenshot of something from a site called Download.com. There's a number of different sites that do that same sort of thing, but in this particular case, I started to look into what happens when you run this particular installer.

Identical installers



Identical installers

Installers from Download.com are the same:

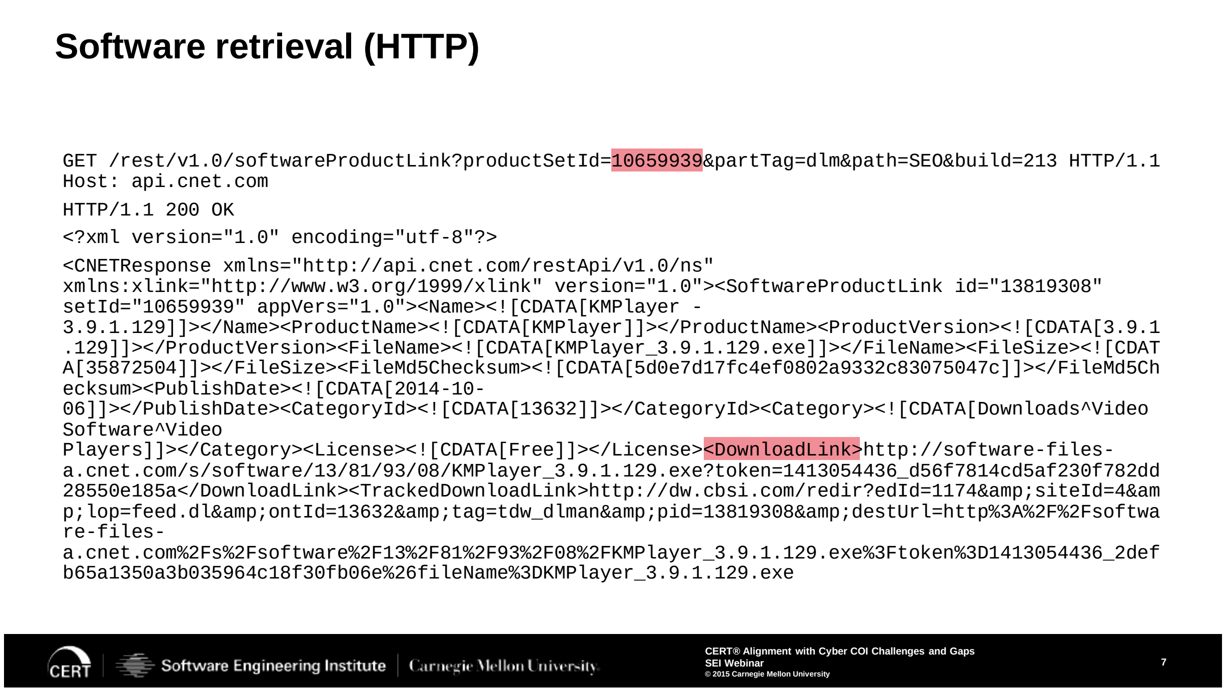
```
5a275a569dce6e2f2f0284d82d31310b *cbsidlm-cbsi213-
Enable__Disable_Registry_Tool-SEO-75812481.exe
5a275a569dce6e2f2f0284d82d31310b *cbsidlm-cbsi213-
KMPlayer-SEO-10659939.exe
```

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

6

**006 But as I started digging into it, I noticed-- the first application that I was looking at was called KMPlayer. There happened to be another tool called Enable/Disable Registry Tool. But what I noticed is as I had both of these files on my system, they had the same MD5. So the installers were absolutely identical. So when you run this installer, how does it actually know what to get from the internet? And if you look closely at the filename, there's actually a number as part of the filename itself. So what happens is when you run this installer, it uses that number as part of the file system, or as part of the filename, to retrieve and execute the installer from the internet.

Software retrieval (HTTP)



Software retrieval (HTTP)

```
GET /rest/v1.0/softwareProductLink?productSetId=10659939&partTag=dlm&path=SEO&build=213 HTTP/1.1
Host: api.cnet.com
HTTP/1.1 200 OK
<?xml version="1.0" encoding="utf-8"?>
<CNETResponse xmlns="http://api.cnet.com/restApi/v1.0/ns"
xmlns:xlink="http://www.w3.org/1999/xlink" version="1.0"><SoftwareProductLink id="13819308"
setId="10659939" appVers="1.0"><Name><![CDATA[KMPlayer -
3.9.1.129]]></Name><ProductName><![CDATA[KMPlayer]]></ProductName><ProductVersion><![CDATA[3.9.1
.129]]></ProductVersion><FileName><![CDATA[KMPlayer_3.9.1.129.exe]]></FileName><FileSize><![CDAT
A[35872504]]></FileSize><FileMd5Checksum><![CDATA[5d0e7d17fc4ef0802a9332c83075047c]]></FileMd5Ch
ecksum><PublishDate><![CDATA[2014-10-
06]]></PublishDate><CategoryId><![CDATA[13632]]></CategoryId><Category><![CDATA[Downloads^Video
Software^Video
Players]]></Category><License><![CDATA[Free]]></License><DownloadLink>http://software-files-
a.cnet.com/s/software/13/81/93/08/KMPlayer_3.9.1.129.exe?token=1413054436_d56f7814cd5af230f782dd
28550e185a</DownloadLink><TrackedDownloadLink>http://dw.cbsi.com/redir?edId=1174&amp;siteId=4&am
p;lop=feed.dl&amp;ontId=13632&amp;tag=tdw_dlman&amp;pid=13819308&amp;destUrl=http%3A%2F%2Fsoftwa
re-files-
a.cnet.com%2Fs%2Fsoftware%2F13%2F81%2F93%2F08%2FKMPlayer_3.9.1.129.exe%3Ftoken%3D1413054436_2def
b65a1350a3b035964c18f30fb06e%26fileName%3DKMPlayer_3.9.1.129.exe
```

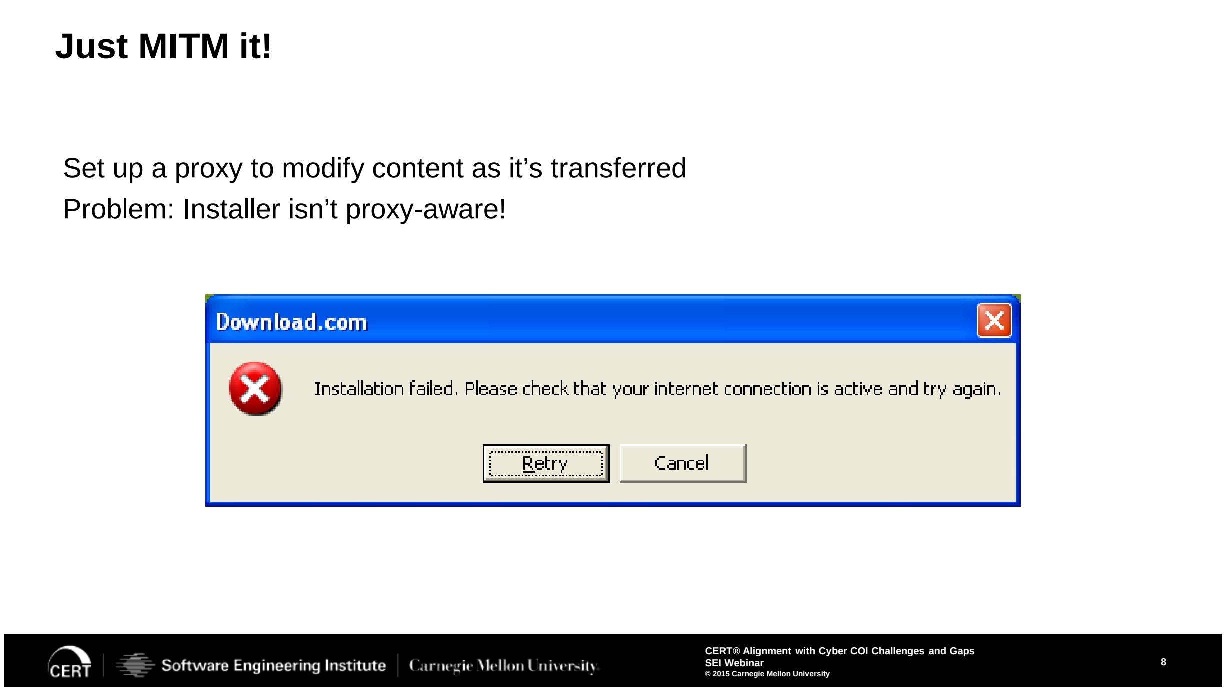
CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

7

**007 So I started to look into, "Well, how does it actually get the software from the internet?" You want to obviously do that securely, and it turns out this particular application was using HTTP, which is not encrypted-- it's completely in the clear-- and if you look you can see that exact same number that was part of the filename is part of the GET request, and then there's a particular section here called "DownloadLink," and that's actually the download location for the actual application that I'm getting. So not only is this particular request in the clear in HTTP, but the download link itself is in HTTP. So I figured, "Well, I should be able to just man-in-the-middle it."

Just MITM it!



Just MITM it!

Set up a proxy to modify content as it’s transferred

Problem: Installer isn’t proxy-aware!

Download.com

Installation failed. Please check that your internet connection is active and try again.

**008 So I should be able to inject myself in between the application and just see would an attacker be able to subvert this particular application, and the common way of doing that is you just configure a system to use a proxy server. At that point, all of your traffic, rather than going directly to the internet, is going to go through this man-in-the-middle system.

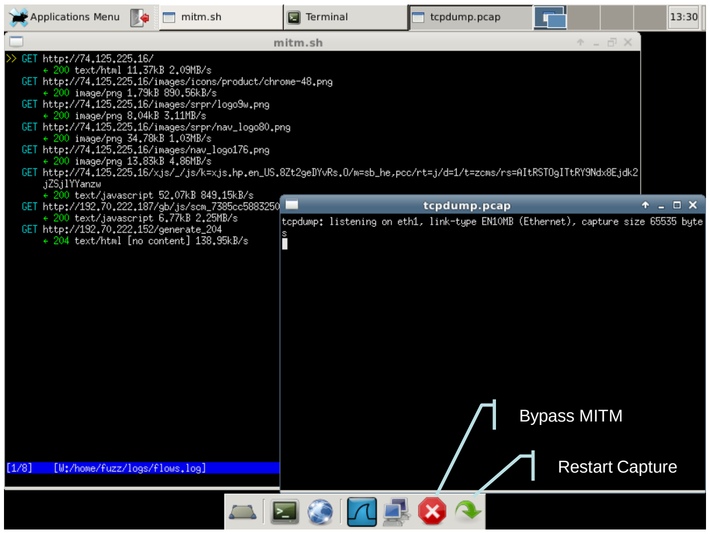
The problem I ran into is this particular installer, it's not aware of your proxy. So you can configure your Windows system to say, "This is the proxy I want to use," but when I go ahead and try to use that installer on a system that only has access through that proxy I see, "Well, you don't have access to the internet." So I was thinking, "Well, how can I dig into this a little bit deeper?" I want something that doesn't operate with proxy settings, but I want something that happens on the network layer.

Solution: CERT Tapioca

Solution: CERT Tapioca

Transparent Proxy Capture Appliance

UbuFuzz + iptables + mitmproxy



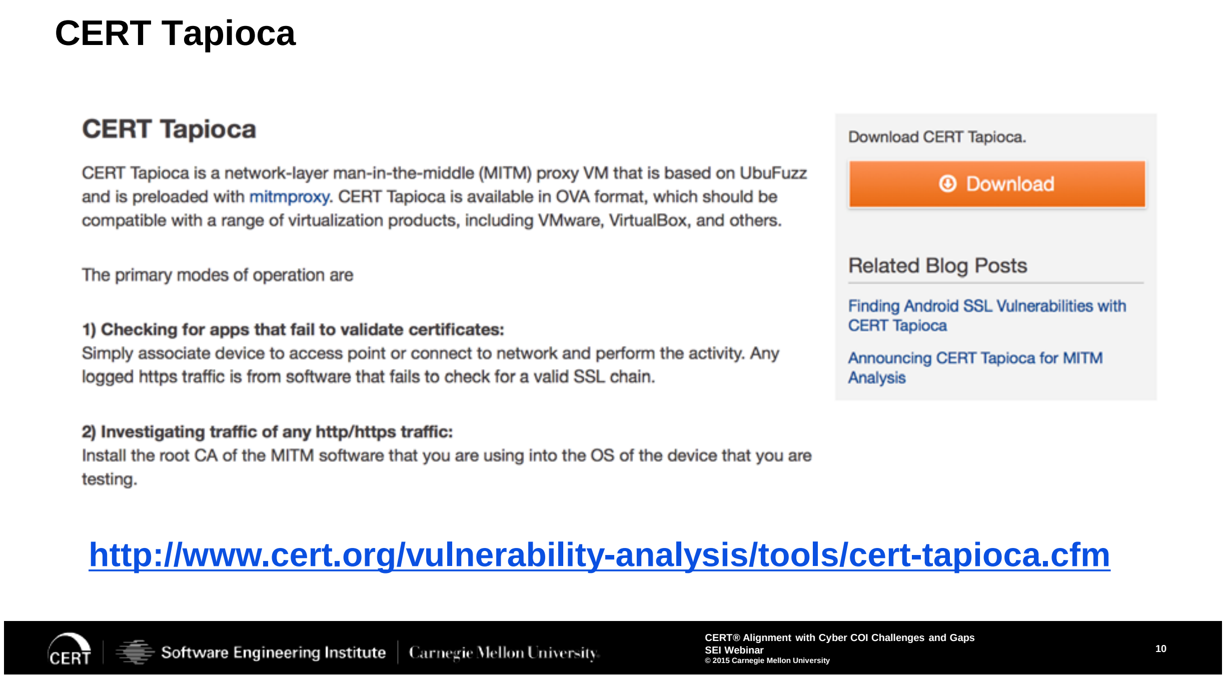
```
mitm.sh
>> GET http://74.125.225.16/
       ← 200 text/html 11.37kB 2.09MB/s
   GET http://74.125.225.16/images/icons/product/chrome-48.png
       ← 200 image/png 1.79kB 890.56kB/s
   GET http://74.125.225.16/images/srpr/logo9w.png
       ← 200 image/png 8.04kB 3.11MB/s
   GET http://74.125.225.16/images/srpr/nav_logo80.png
       ← 200 image/png 34.78kB 1.03MB/s
   GET http://74.125.225.16/images/nav_logo176.png
       ← 200 image/png 13.83kB 4.86MB/s
   GET http://74.125.225.16/xjs/_/js/k=xjs.hp.en_US.8Zt2geDYvRs.O/m=sb_he,pcc/rt=j/d=1/t=zcms/rs=AItRSTOgITtRY9Ndx8Ejdk2
       jZSjlYYanzw
       ← 200 text/javascript 52.07kB 849.15kB/s
   GET http://192.70.222.187/gb/js/scm_7385cc5883250
       ← 200 text/javascript 6.77kB 2.25MB/s
   GET http://192.70.222.152/generate_204
       ← 204 text/html [no content] 138.95kB/s

[1/8]    [W:/home/fuzz/logs/flows.log]
```

```
tcpdump.pcap
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 65535 byte
s
```

**009 So I really just used-- I had a virtual machine that we had actually released as part of our fuzzing framework called BFF. There's a particular machine called UbuFuzz, which is really just a stripped down version of Ubuntu-- no password or anything, because we just want it to be pretty simple-- and we used an application called mitmproxy, and the real magic of where this happens is iptables is used to transparently redirect network traffic to this MITM Proxy software. So what happens is when I run an application that's talking using web protocols, it's transparently being interpreted by this particular virtual machine, and basically it's a virtual machine with two network adapters; one of them connects to the internet, the other one connects to the system that you want to analyze. So it's relatively straightforward, but just providing it in an appliance form helps.

CERT Tapioca


CERT Tapioca

CERT Tapioca

CERT Tapioca is a network-layer man-in-the-middle (MITM) proxy VM that is based on UbuFuzz and is preloaded with mitmproxy. CERT Tapioca is available in OVA format, which should be compatible with a range of virtualization products, including VMware, VirtualBox, and others.

The primary modes of operation are

1) Checking for apps that fail to validate certificates:
Simply associate device to access point or connect to network and perform the activity. Any logged https traffic is from software that fails to check for a valid SSL chain.

2) Investigating traffic of any http/https traffic:
Install the root CA of the MITM software that you are using into the OS of the device that you are testing.

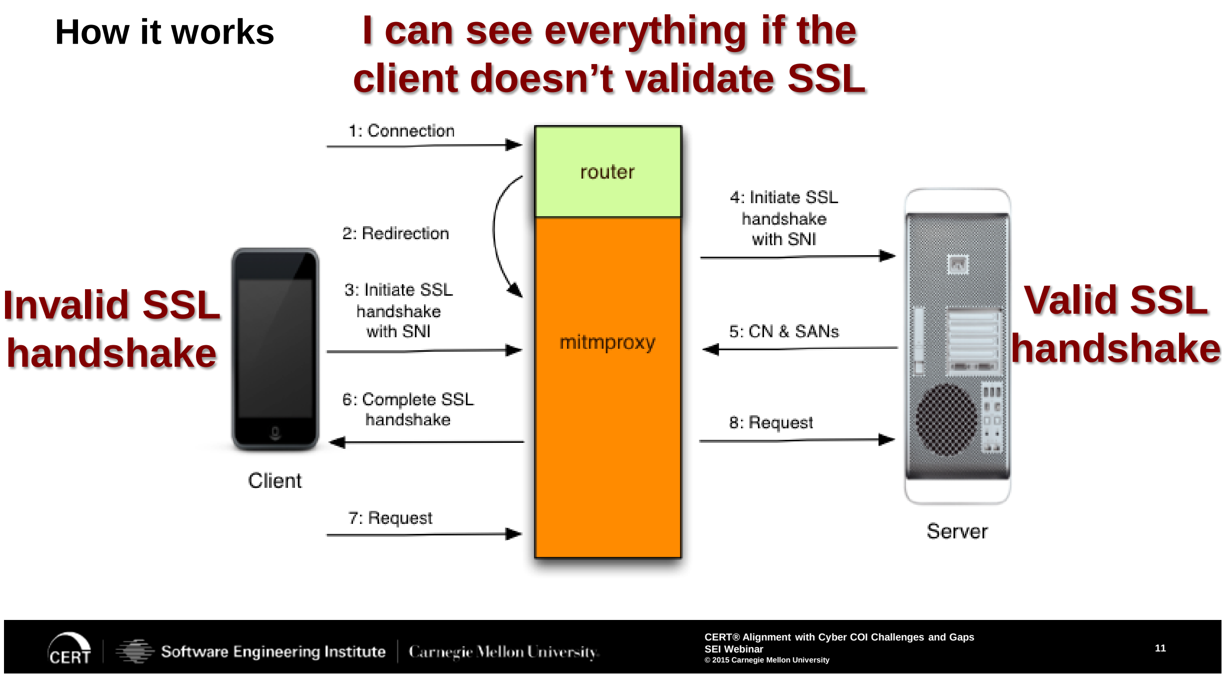
Download CERT Tapioca.
Download
Related Blog Posts
Finding Android SSL Vulnerabilities with CERT Tapioca
Announcing CERT Tapioca for MITM Analysis

http://www.cert.org/vulnerability-analysis/tools/cert-tapioca.cfm

CERT | Software Engineering Institute | Carnegie Mellon University
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University
10

**010 And we made this publicly available. You can download this particular appliance, and it's pretty much compatible with most of the virtualization products that you might use.

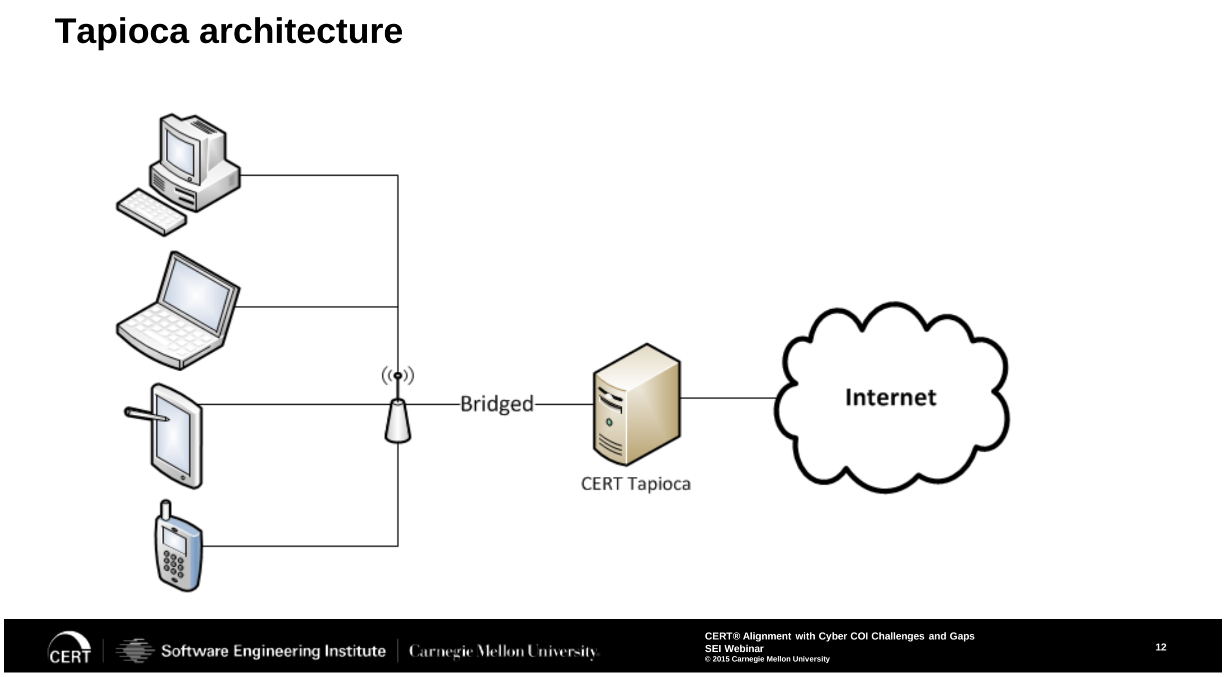
I can see everything if the client doesn't validate SSL


How it works
I can see everything if the client doesn't validate SSL
Invalid SSL handshake
Valid SSL handshake
1: Connection
router
2: Redirection
3: Initiate SSL handshake with SNI
mitmproxy
4: Initiate SSL handshake with SNI
5: CN & SANs
6: Complete SSL handshake
8: Request
Client
7: Request
Server
CERT
Software Engineering Institute
Carnegie Mellon University
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University
11

**011 So just a little bit of the detail about how MITM Proxy, and subsequently the Tapioca virtual appliance works. It's important to look at what happens when you-- now mostly this is focusing on secure communications, so through SSL, and I'll get into a little bit of the details later on-- but essentially what happens is you have a client-- and that might be a phone, it might be a Windows system, Linux, it might be a smart light bulb-- but basically what happens is when the client communicates with this proxy, the proxy makes a valid SSL handshake with the server. What also happens is the client makes its own SSL handshake with the MITM Proxy. So it's kind of a two-stage sort of thing. If the client is not making sure that it's a valid SSL handshake--

This particular component
in the middle can see everything. So
it's a very straightforward way of
observing traffic of applications that
are not validating SSL certificates.

Tapioca architecture

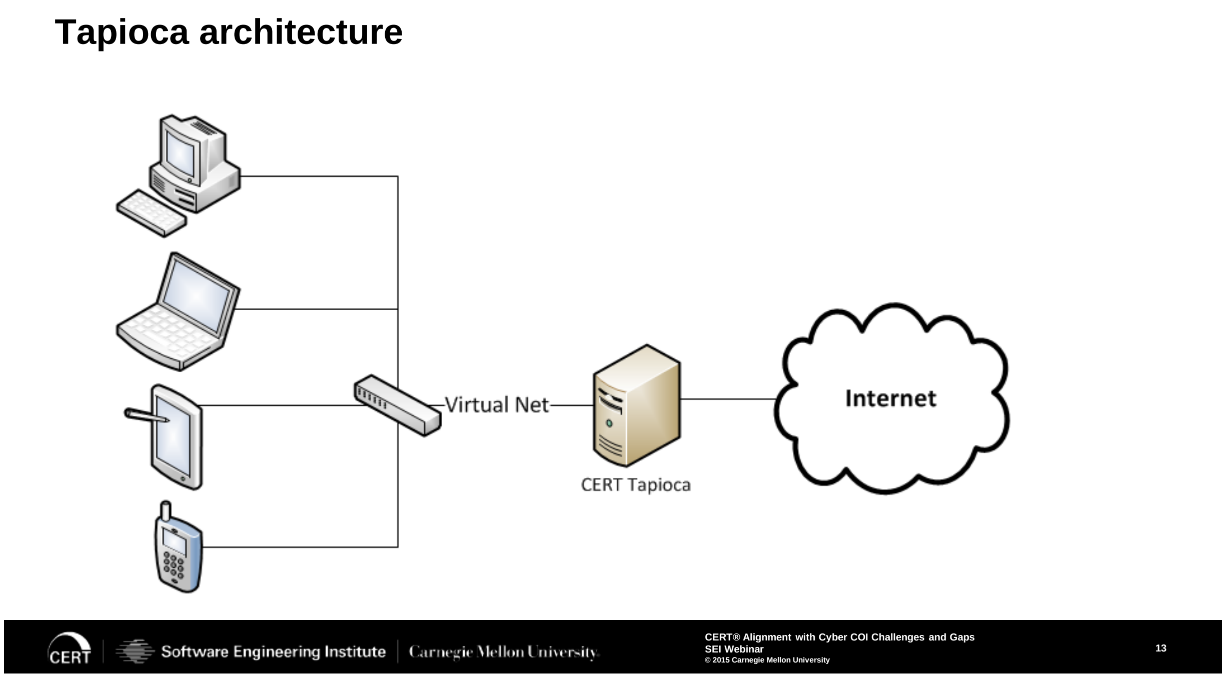


Tapioca architecture

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University
12

**012 And the architecture of
Tapioca, like I mentioned before, is
it's a virtual appliance. You could
actually make it into a physical
machine if you had an application
that needed that. But basically you
have the Tapioca machine that has
two network connections, one of
them is an uplink to the internet, and
then the other side is something that
is connected to something that you
want to test, and whatever you want
to test can be anything that talks
TCP/IP-- it's talking with web traffic.
It could be a phone, it could be a
virtual machine.

Tapioca architecture


Tapioca architecture

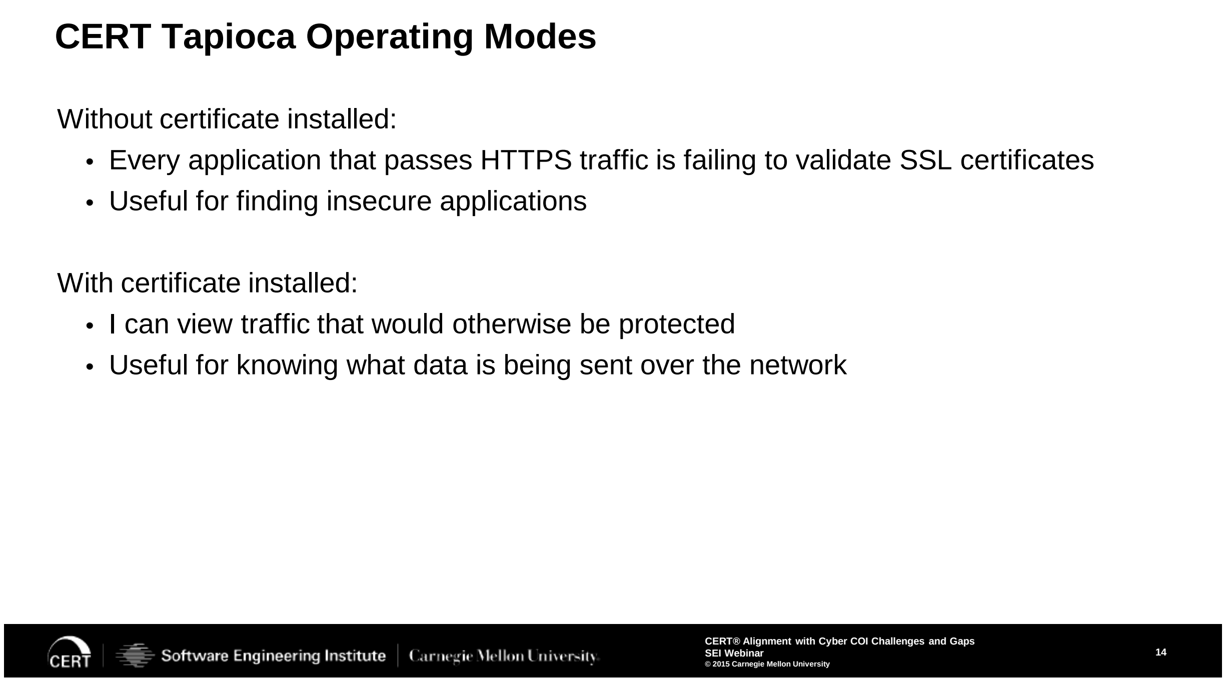
CERT Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

13

**013 In this particular
case, you can have a virtual network.
So it could be actually physical
devices that I might have in my
pocket, or this could be all virtual
machines, and Tapioca is able to
observe the traffic that happens
between the client and the internet.

CERT Tapioca Operating Modes



CERT Tapioca Operating Modes

Without certificate installed:

- Every application that passes HTTPS traffic is failing to validate SSL certificates
- Useful for finding insecure applications

With certificate installed:

- I can view traffic that would otherwise be protected
- Useful for knowing what data is being sent over the network

CERT | Software Engineering Institute | Carnegie Mellon University

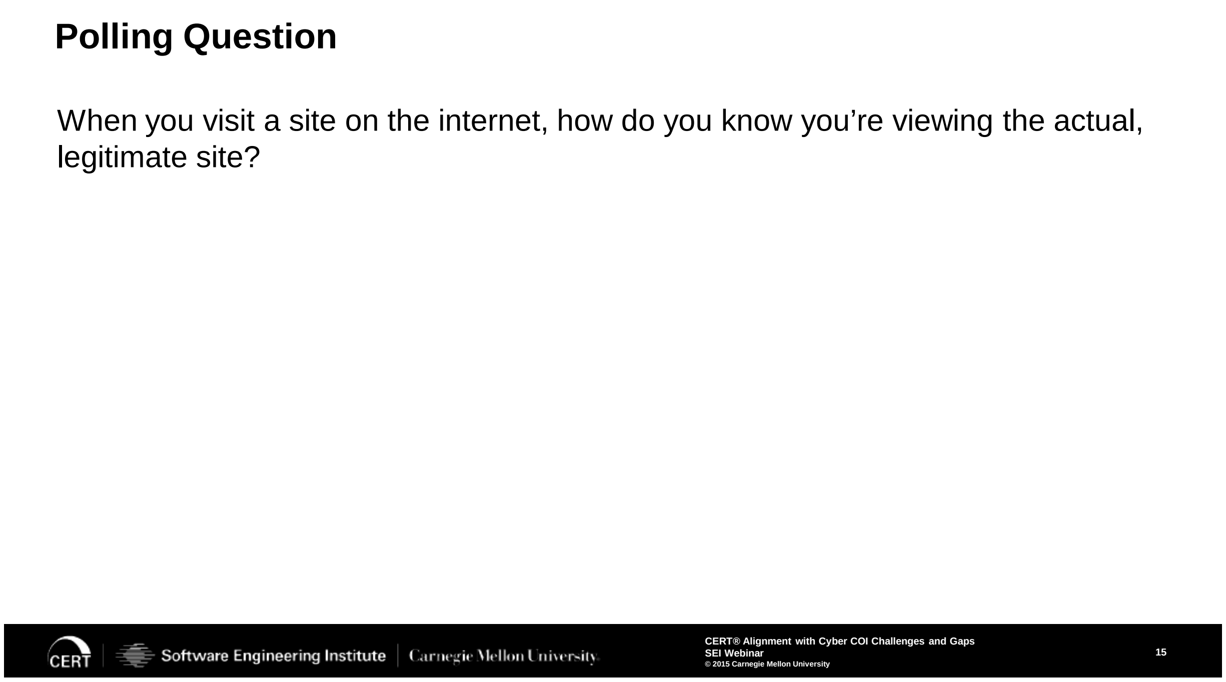
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

14

**014 And Tapioca really has two primary operating modes. I had mentioned the first one earlier. The other mode that you can use Tapioca is-- or, excuse me. The first mode that I mentioned earlier is if I have a system where I don't have the certificate installed for the SSL traffic-- and that's basically just by default, if I take my cell phone or my smartphone and I associate it with an access point that is uplinked to Tapioca-- what that means is that any client system is going to receive an invalid SSL handshake. So what that means is you can look at the CERT Tapioca traffic log, and if you ever see SSL or HTTPS traffic passing through Tapioca, that means that the client has ignored the validity of the certificate. And so what you can do is you can look for applications that are communicating insecurely on the internet. If you're not validating an SSL certificate, that means that you actually could be talking to something other than you expect.

The other mode that you can operate Tapioca in is if you do install the certificate in the client system that you're analyzing, what that might allow you to do is if there's a particular application that I want to look at, a particular-- let's say there's one particular app that I think is interesting and I want to know-- when I just look at the network traffic of that particular system with Wireshark, for example, you don't actually see what is transmitting over the network. You see an SSL handshake and then at that point everything is encrypted. If I have an application that I'm interested to see, "Is this actually perhaps sending sensitive information over the network? Is it sending or retrieving something that I don't want it to be sending?"-- you can actually observe that traffic if you have the certificate installed, and that assumes that you have the ability to install a certificate through the platform that you're looking at. Certain Internet of Things type of devices, like a smart light bulb, you might not be able to do that. But for the type of platforms that support certificate installation, you can look at that traffic.

Polling Question



Polling Question

When you visit a site on the internet, how do you know you're viewing the actual, legitimate site?

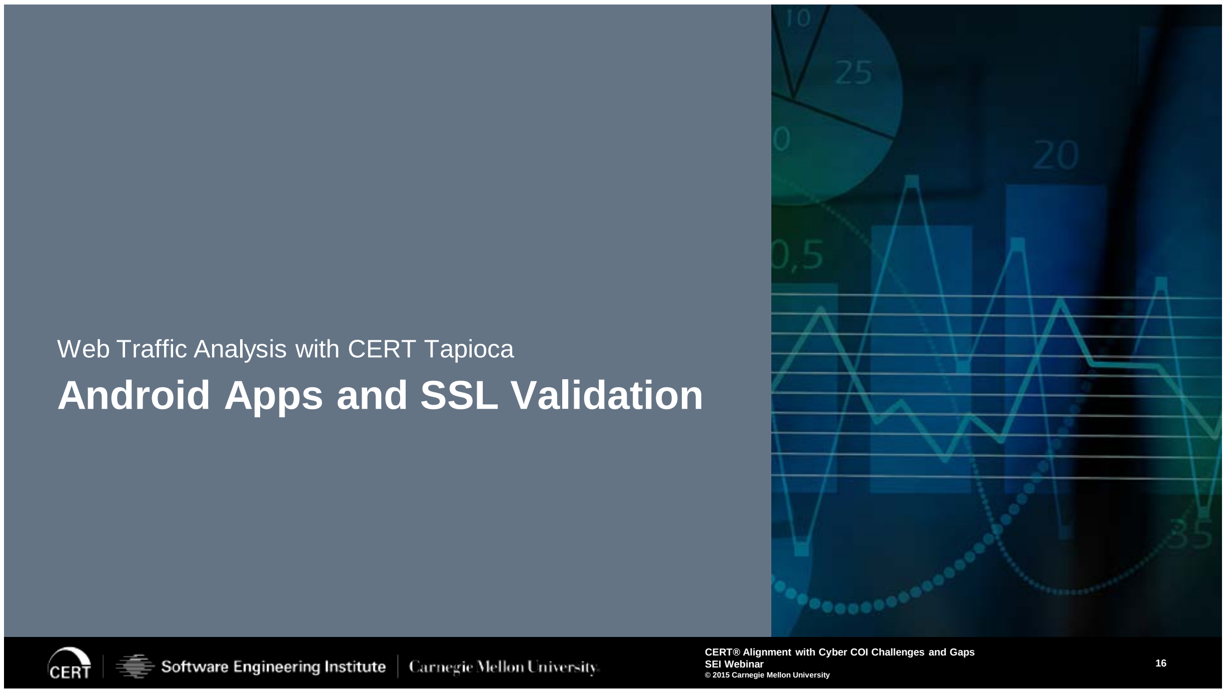
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

15

**015 Shane: So we're going to launch a quick polling question folks for you here. When you visit a site on the internet, how do you know you are viewing an actual legitimate site? Since there's multiple answers to this, we ask that you just type a response into the Q&A window and then we'll get back to Will with some of the responses as they flow in.

Will Dormann: Absolutely. So it seems like a very straightforward question, but there are actually some idiosyncrasies to that. There are some aspects that might surprise you.

Web Traffic Analysis with CERT Tapioca



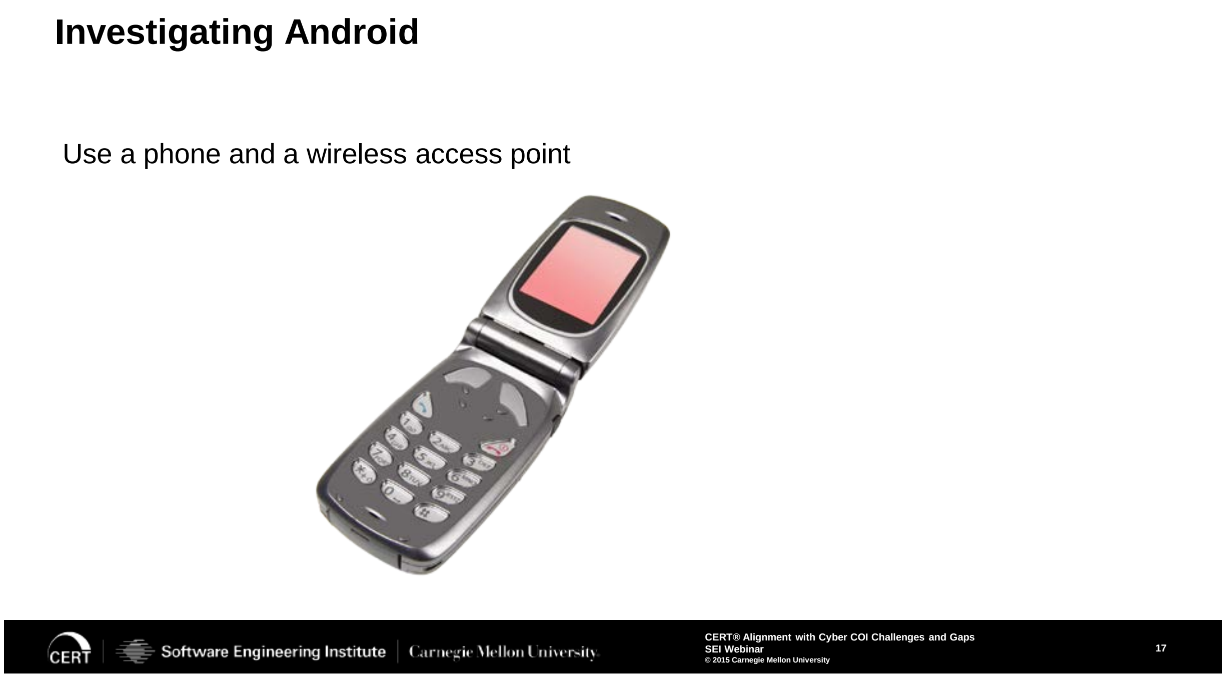
Web Traffic Analysis with CERT Tapioca

Android Apps and SSL Validation

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

16

**016 So one of the first things that I used as an application of the CERT Tapioca appliance is I started look at Android applications-- and this is the first operating mode of Tapioca. I want to look at apps that are not validating SSL certificates.



Investigating Android

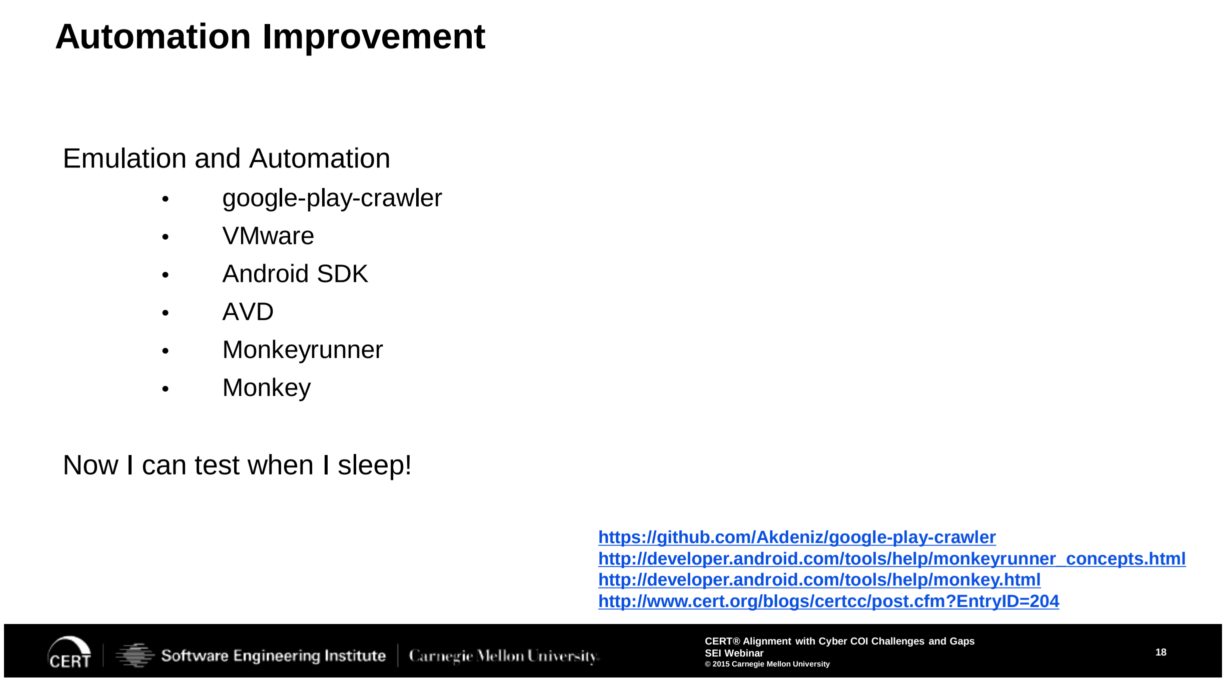
Use a phone and a wireless access point

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

17

**017 The first thing that I ended up doing is-- you start out with-- any sort of test that you're doing, you start out with the simplest sort of situation, and I happened to have an Android phone, and I just associated it with an access point that was connected to Tapioca, and you can try an app manually and see, "Is there any HTTP traffic that is getting sent over the network?"

Automation Improvement



Automation Improvement

Emulation and Automation

- google-play-crawler
- VMware
- Android SDK
- AVD
- Monkeyrunner
- Monkey

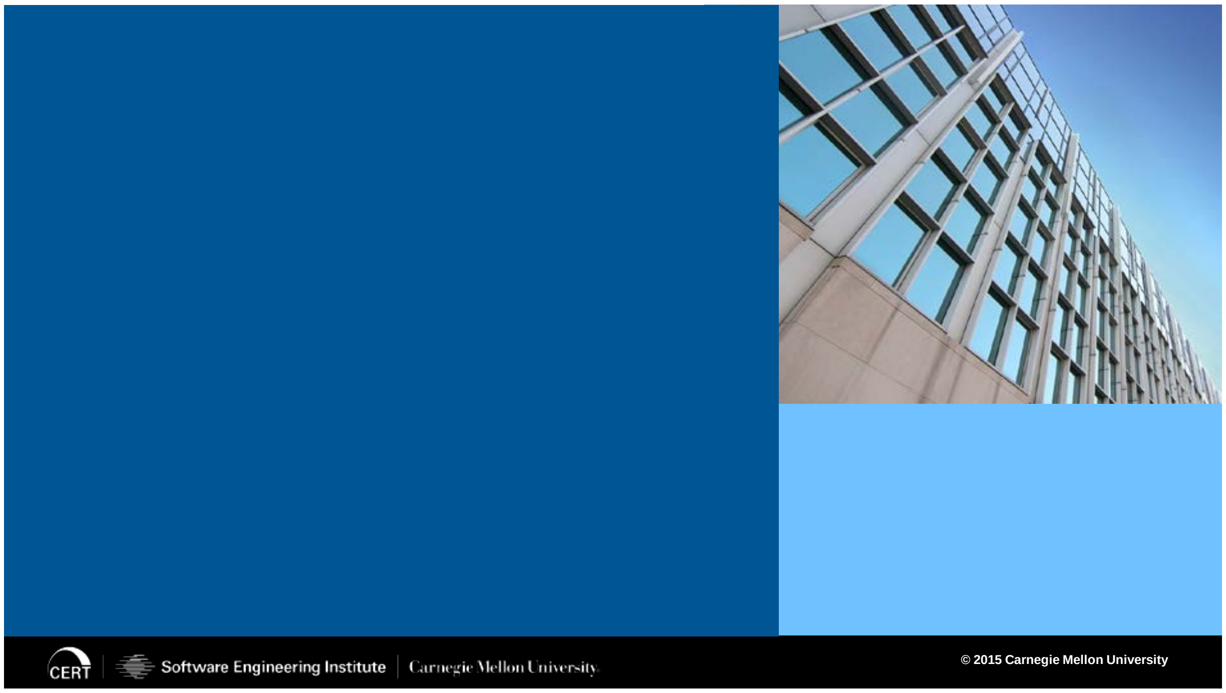
Now I can test when I sleep!

https://github.com/Akdeniz/google-play-crawler
http://developer.android.com/tools/help/monkeyrunner_concepts.html
http://developer.android.com/tools/help/monkey.html
http://www.cert.org/blogs/certcc/post.cfm?EntryID=204

CERT | Software Engineering Institute | Carnegie Mellon University
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University
18

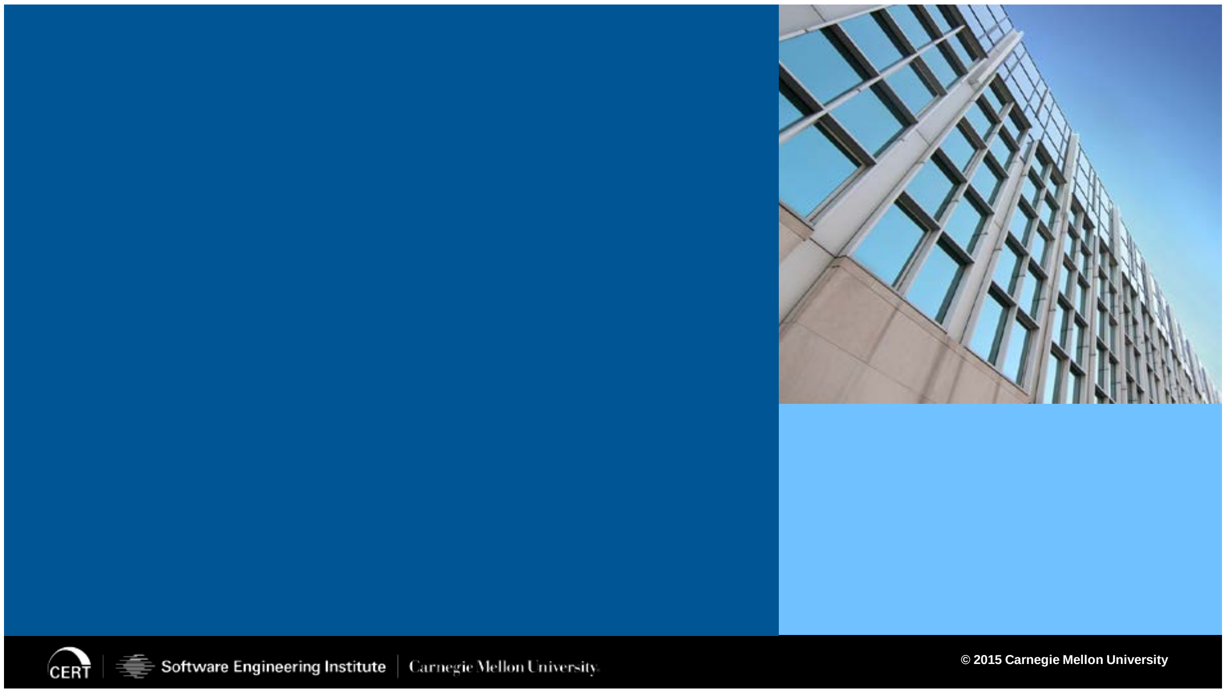
**018 But obviously that doesn't scale very well. There's over a million apps, a million free apps in the Google Play store, and I got bored very quickly. But as we do with a lot of different things, when you're analyzing things in bulk, is you look into ways that you can automate the technology, and there are a number of different components that we used. Google Play Crawler is a component used for obtaining apps from the Play Store. VMware. We started out using the Android SDK, which comes with an emulator, or an Android virtual machine, and then Monkey and monkeyrunner are both components that are associated with the SDK that allow you to automate an application. So if I'm firing up an Android application, I want to actually exercise as much of that code as I can. So basically I'm emulating keystrokes and that sort of thing. So the nice thing about that is now I can test when I can sleep. I can go home from work and it's still testing, and I come back and there's more things waiting for me.

Emulator video


CERT
Software Engineering Institute
Carnegie Mellon University
© 2015 Carnegie Mellon University

**019 So here's an actual video of the emulator, and as you can see-- it doesn't make a lot of sense-- but you can see it's actually typing things into the Android application, and when you type things into a form or type things into the application, quite often-- in this particular example, you've got a username, you've got a password, and I've got a button that says Log In. If my automation is working properly, it'll type in the username, it'll type in something for the password, and ideally it presses that Log In button. So if at that point I see traffic that goes over the network over HTTPS, I know that I now have a vulnerable application.

Tapioca side of things

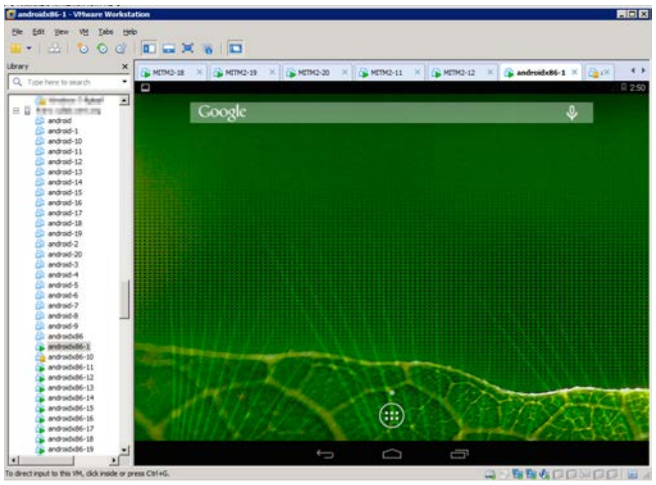

CERT
Software Engineering Institute
Carnegie Mellon University
© 2015 Carnegie Mellon University

**020 So this is actually the Tapioca side of things, and it kind of goes along with you seeing the Android application being automated. But basically what's happening-- and this is accelerated, because at the time that I took this video, it was through the emulator as opposed to a more fast virtualized version of Android. But basically what's happening is I'm seeing all of the web requests that are occurring, but in this particular case I'm actually seeing HTTPS. So what that means is that this particular application has sent data insecurely over the network. So I can log this as a vulnerable application.

Virtualization

Virtualization

http://www.android-x86.org/



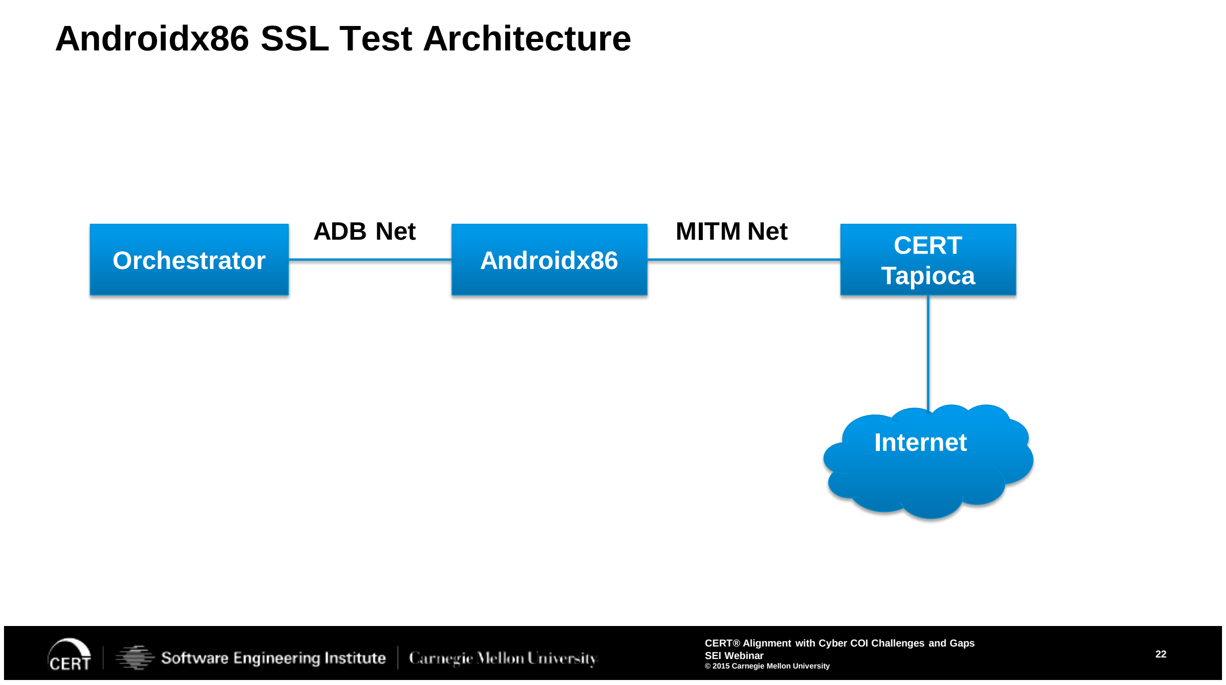
CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

21

**021 So that was through the emulator. Because emulation is much slower that virtualization, because the emulator is emulating the ARM architecture, it turns out that you can actually get an x86 version of Android, and you can run that directly in your virtualization software, and the advantage to that is because I'm not emulating and I'm virtualizing it, you get a tremendous speed increase. So just by transitioning to the x86 architecture, we were able to get at least a tenfold increase in performance, and as you can see in this particular screenshot here, I've got 20 different Android virtual machines. So as I'm testing these applications, not only am I testing them more quickly for each particular application, but I'm doing it twentyfold at a time. And if we wanted to ramp it up a little bit higher, we probably could have used some sort of cloud technology to really do it at scale.

Androidx86 SSL Test Architecture



Androidx86 SSL Test Architecture

CERT | Software Engineering Institute | Carnegie Mellon University

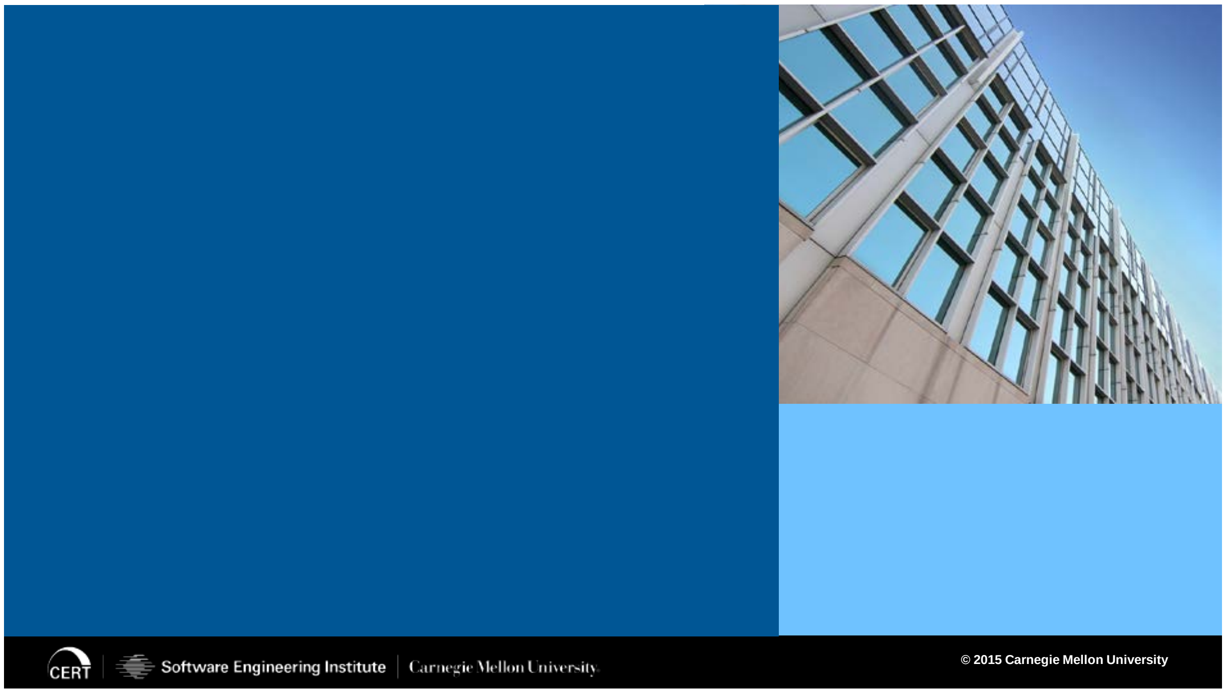
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

22

**022 So this is the architecture for the testing of Android x86. It's slightly different than using the emulator in that when you use the Android emulator, you can run the ADB command, which is the Android Debugging Bridge, and I can just tell this application to do this, and it happens automatically. With Android x86, because it's an actual live operating system, you don't have that automatic debug bridge, so really what you need is just a second network. So you can see on the left side, or in the middle of this particular screen, I've got the Android x86 virtual machine. That has two network adapters. One of them is just the orchestration, telling the

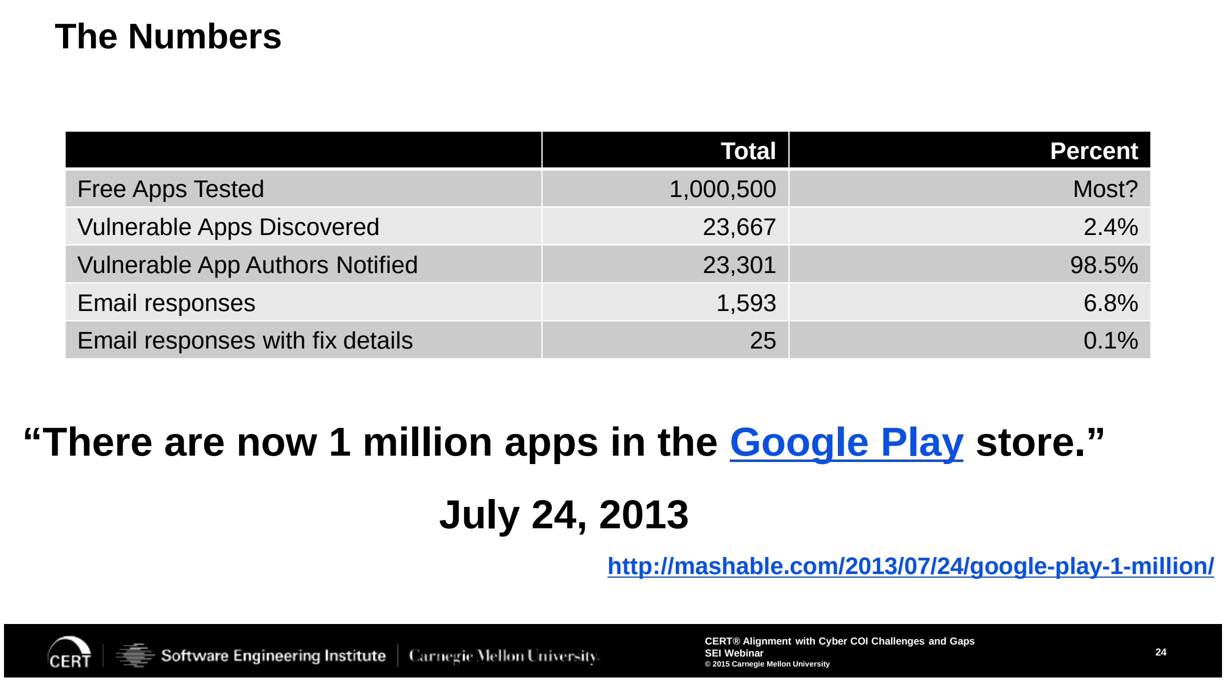
Android virtual machine what to do, and then on the right side I've got CERT Tapioca with a standard operation of one side goes to the internet, the other side goes to the application that I would like to test.

Particular screen of automation


CERT
Software Engineering Institute
Carnegie Mellon University
© 2015 Carnegie Mellon University

**023 And here's a particular screen of automation. It's really just-- it's a fixed screenshot at this point, but what's happening is basically there are different terminals that are sending commands to the different Android applications that are being tested.

The Numbers



The Numbers

	Total	Percent
Free Apps Tested	1,000,500	Most?
Vulnerable Apps Discovered	23,667	2.4%
Vulnerable App Authors Notified	23,301	98.5%
Email responses	1,593	6.8%
Email responses with fix details	25	0.1%

"There are now 1 million apps in the Google Play store."

July 24, 2013

http://mashable.com/2013/07/24/google-play-1-million/

CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

24

**024 So this is actually a test that I had performed over the period of several months, really just downloading one application at a time, and you can see we got just past the point of one million applications. We looked at free applications, assuming that free applications are representative of the entire set of Android applications. It's simplest to download the free applications.

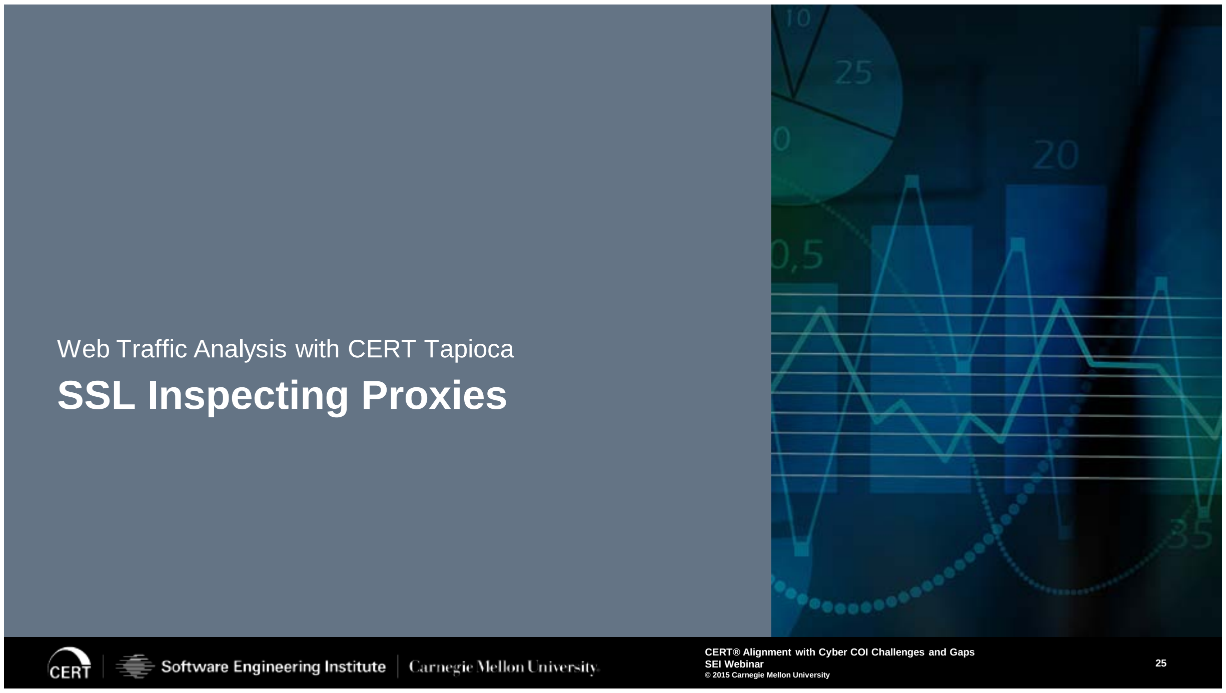
In 2014 there was an article that mentioned that they now reached one million apps, so at the time that I started this particular test in 2014, I assumed that we got probably most of the Android applications. So after testing just over one million Android applications, we were able to find close to 24 thousand vulnerable applications. Some of these applications are very important. Some of them are banking applications. Some of them are credit card applications. These are things where I am now perhaps logging into my Google account, or some sort of social media account. Now, some of them are not important at all. It could be that I have an app where it's really just a game and it retrieves and advertisement over HTTPS but it does it insecurely. So it's really-- you can't really lump the severity of all of these all into one, but just realize that there's an entire spectrum of importance in the apps that we're looking at.

Most of the apps had email addresses where we could notify them, some of them did not. So really we were able to notify over 23 thousand application developers, and this was an interesting case of automating the whole coordination aspect of vulnerabilities, and that is not only just finding the vulnerabilities but actually notifying the vendor that needs to find out.

The part where it gets a little bit depressing here near the end is what sort of response did we get from the Android application authors. Well, out of the 23 thousand emails that were sent out, we received about 1500 responses. Now, a number of those are just automatic responses saying, "Hey, we got your email." Now, when you go to the actual count of people that responded saying, "Hey, we fixed this. Thanks for letting us know," it's actually 0.1

percent, so about 25 authors out of approximately 24 thousand notified. So it seems like the authors that are creating a lot of the Android tools-- and that's not to say that only 25 authors fixed their software. It's 25 authors told us that they fixed it. So there could be, after some amount of time, a good number of authors fixed their app, but we only know about the ones that actually told us.

Web Traffic Analysis with CERT Tapioca


Web Traffic Analysis with CERT Tapioca

SSL Inspecting Proxies

CERT | Software Engineering Institute | Carnegie Mellon University

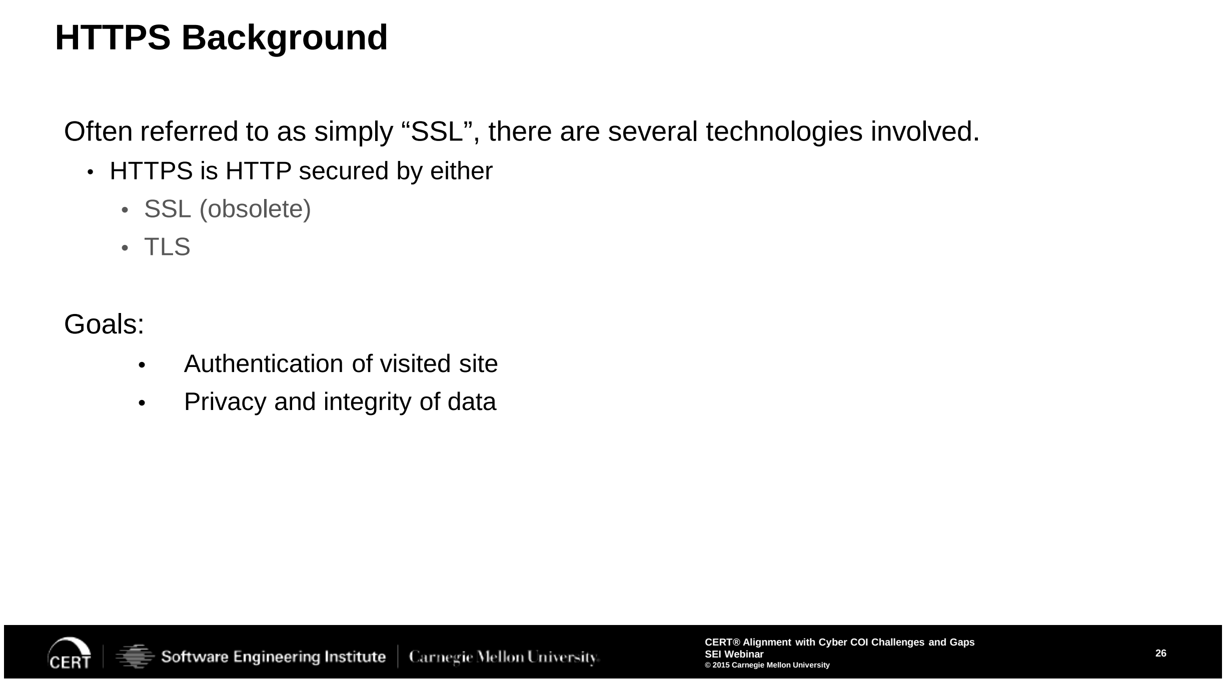
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

25

**025 So going into the other aspect of-- or, excuse me-- another application of using Tapioca, and that is I started looking at the Android applications, and since we already had this appliance that was premade and you can just kind of plug it in, I started looking into a different application for it. And this is not specific apps that are running on a phone, but this is actually an entire

technology called SSL Inspecting
Proxies.

HTTPS Background



HTTPS Background

Often referred to as simply “SSL”, there are several technologies involved.

- HTTPS is HTTP secured by either
  - SSL (obsolete)
  - TLS

Goals:

- Authentication of visited site
- Privacy and integrity of data

Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

26

**026 So just a little bit of a background about HTTPS. Most people say SSL because the two terms are used a little bit interchangeably. Under the hood, HTTPS is a security protocol. It's basically HTTP, which is the web protocol, but the S gives it the security aspect, and under the hood it uses either SSL or TLS. But I won't get into any of the real details, I just want to talk about the main capabilities that HTTPS provides, and that basically is authentication of the site that I visited. So when I visit my bank account, or if I visit my banking website, I want to know that that's actually the banking site and not some guy that's sitting at my local coffee shop.

Shane: Do we want to get into some of the responses from the polling question earlier? Is now a good time to do that?

Will Dormann: We can do that. That would be pretty good.

Shane: Okay. So we got a number of responses. The first one was HTTPS. Another one replied, "Sad to say, I don't know." Another one, "I do not trust any response I receive from the internet." Another one, "Verify the URL was correct." Another one is, "WRT, with regards to, authentic website, I cheat. We use OpenDNS to screen websites before we allow clients to reach them." Another one, "We usually just make sure that there are no errors when navigating to HTTPS sites, but you can look at the CA information for the cert and other details of the cert."

Will Dormann: Oh, that's a good one.

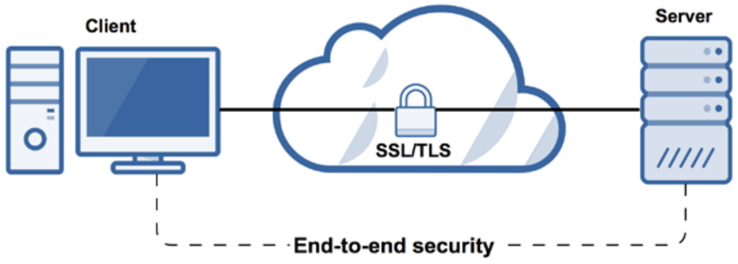
Shane: "I look at the IP address in the return. If it was intercepted at outset, I won't know. Good point." I guess to you. Somebody says, "I don't know." Another one, "HTTPS no underscore URL." Another one, "I check the URL and make sure that it's formatted appropriately for the site I'm trying to visit. I'll also frequently find the correct URL myself instead of a hot link for important sites."

Will Dormann: That's a good idea, and not using a search engine result to click on a link but actually typing it in, and actually if you type in HTTPS://, that is a really-- it's a good start to be able to realize that you are on the right site that you think that you are. Because if I were actually just to type in the domain name of my bank or any particular site that I want to go to, I'm relying solely upon DNS, which could be under attack or control, and I might end up at some other particular site. It's actually-- it's a really tricky sort of situation, and the unfortunately aspect is out of all the particular suggestions that I heard is none of them are actually going to guarantee that you're on the site that you think you are-- and I'll get into some of the details as to why that is the case.

So authentication of the site that's visited is one of the primary goals of HTTPS. The other thing is the privacy and integrity of the data. So if I'm on my banking site-- and yes, I'm sure that it is the site that I think that I'm on-- I want to make sure that anything that gets sent between my computer and the site that I'm connecting to-- I want to make sure that nobody can observe that information. So if it's a social security number, if it's a login or password, I want to make sure that nobody can actually access that.

HTTPS Expectation

HTTPS Expectation



CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

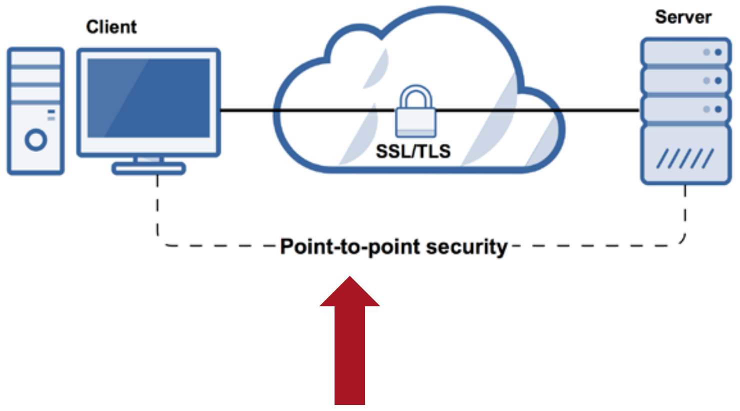
27

**027 So the general expectation that people have of HTTPS is that I have end-to-end security. I have my client system. It might be a Windows system, Linux. It might be a phone. But basically I'm connecting to a server and it's using HTTPS, which under the hood is either SSL or TLS, and I assume that I have end-to-end security between my computer or my system that I'm using and the server; that information is protected.

In actuality, there's a little bit different security that it gives you, and those two frames look pretty--those two slides look pretty similar, but the real difference here is this phrase "point-to-point security".

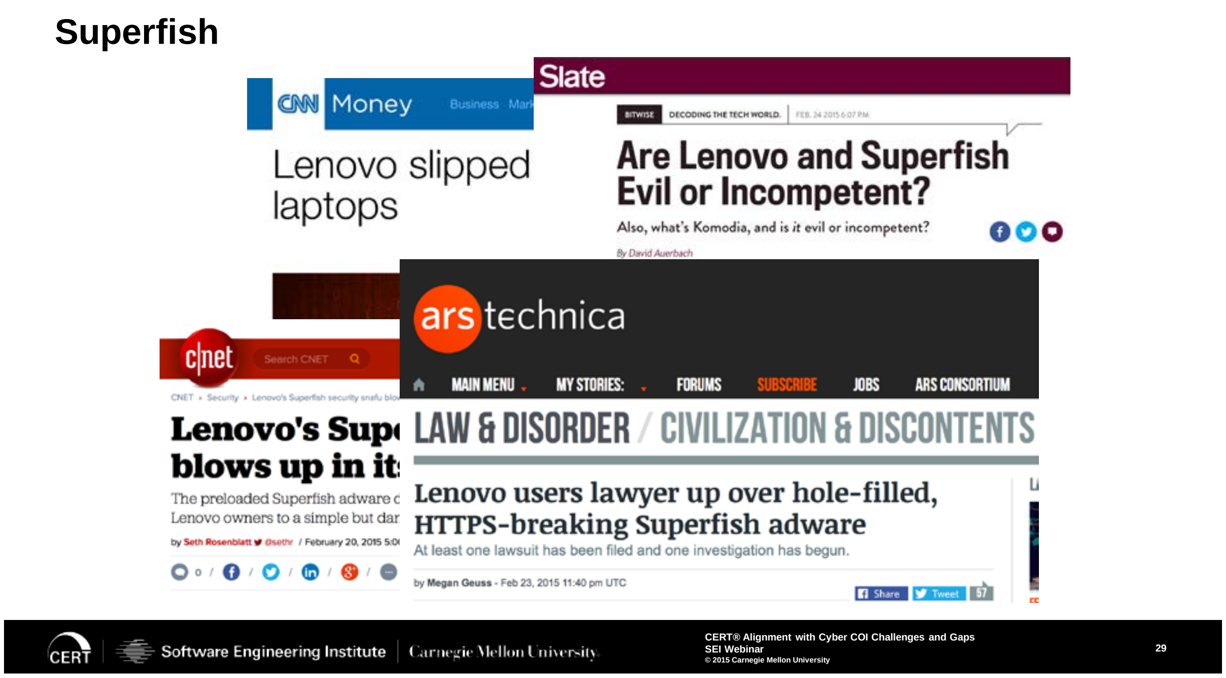
HTTPS Reality

HTTPS Reality



**028 When I am connecting to something using HTTPS, I actually do not get security between my client and the server all the way guaranteed. It's just to the next point, and I'll get into some of the details as to why that is important.

Superfish



Superfish

CNN Money Business Mark

Lenovo slipped
laptops

Slate

BITWISE DECODING THE TECH WORLD. FEB. 24 2015 6:07 PM

Are Lenovo and Superfish Evil or Incompetent?

Also, what's Komodia, and is it evil or incompetent?

By David Auerbach

cnet Search CNET

CNET › Security › Lenovo's Superfish security snafu blo

Lenovo's Sup
blows up in it

The preloaded Superfish adware c
Lenovo owners to a simple but dar

by Seth Rosenblatt @sethr / February 20, 2015 5:0

ars technica

MAIN MENU MY STORIES: FORUMS SUBSCRIBE JOBS ARS CONSORTIUM

LAW & DISORDER / CIVILIZATION & DISCONTENTS

Lenovo users lawyer up over hole-filled, HTTPS-breaking Superfish adware

At least one lawsuit has been filed and one investigation has begun.

by Megan Geuss - Feb 23, 2015 11:40 pm UTC

Share Tweet 57

CERT | Software Engineering Institute | Carnegie Mellon University

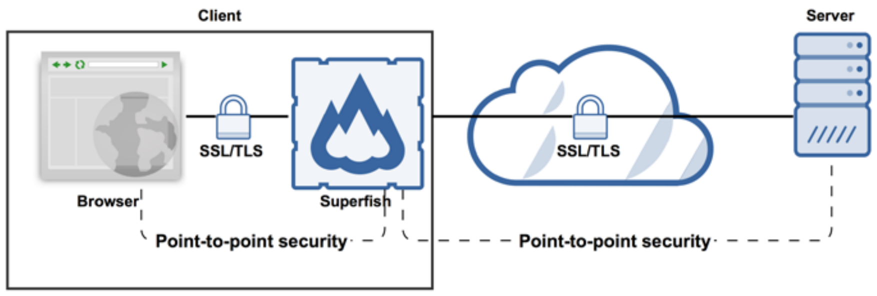
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

29

**029 So earlier this year there was a particular event in the media called SuperFish, and it happened to be a particular computer manufacturer bundled some software in, and some people labeled it as malware, adware-- the terms are generally interchangeable at this point-- and it was in the media in a lot of different cases, and SuperFish was something that actually injected itself into your web traffic, and even if that traffic was encrypted. So, for example, if I use a popular search engine, quite often that traffic is using HTTPS, but somehow this particular software was able to inject ads or other media into that particular web session, which makes you wonder how can that occur.

How Can Superfish Work?

How Can Superfish Work?



Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

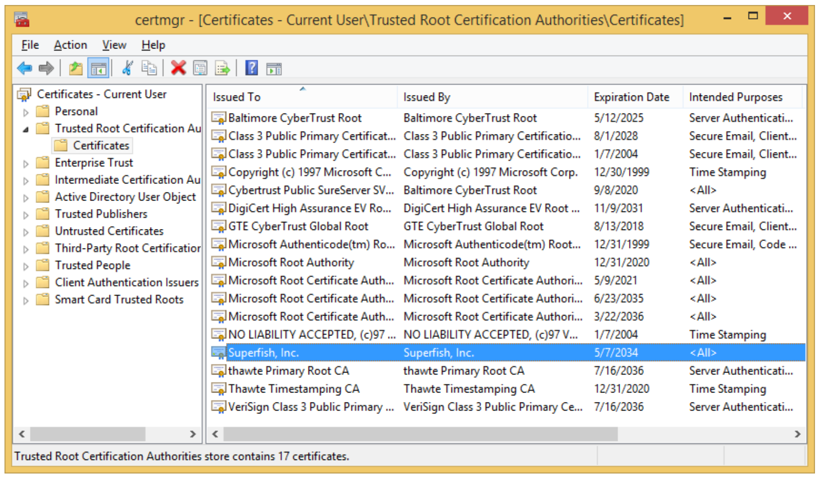
30

**030 How is it that this software is able to modify things that should be protected with HTTPS?

And if you look here, the client system on the left side-- it might be your computer system, and you have your web browser-- and SuperFish is actually a proxy that is doing HTTPS interception. So if I look, I can see I have point-to-point security between my web browser and SuperFish, and I have point-to-point security between SuperFish and the server.

How Can Superfish Work?

How Can Superfish Work?



certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificates]

File Action View Help

- Certificates - Current User
  - Personal
  - Trusted Root Certification Au
    - Certificates
  - Enterprise Trust
  - Intermediate Certification Au
  - Active Directory User Object
  - Trusted Publishers
  - Untrusted Certificates
  - Third-Party Root Certificatior
  - Trusted People
  - Client Authentication Issuers
  - Smart Card Trusted Roots

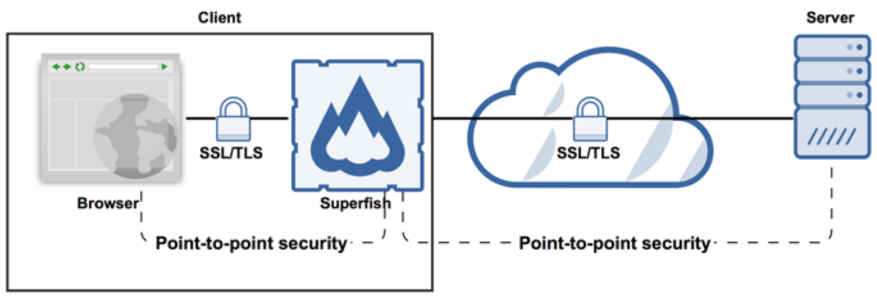
Issued To	Issued By	Expiration Date	Intended Purposes
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Server Authenticati...
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/1/2028	Secure Email, Client...
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	1/7/2004	Secure Email, Client...
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping
Cybertrust Public SureServer SV...	Baltimore CyberTrust Root	9/8/2020	<All>
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root ...	11/9/2031	Server Authenticati...
GTE CyberTrust Global Root	GTE CyberTrust Global Root	8/13/2018	Secure Email, Client...
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	12/31/1999	Secure Email, Code ...
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	5/9/2021	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	6/23/2035	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	3/22/2036	<All>
NO LIABILITY ACCEPTED, (c)97 ...	NO LIABILITY ACCEPTED, (c)97 V...	1/7/2004	Time Stamping
Superfish, Inc.	Superfish, Inc.	5/7/2034	<All>
thawte Primary Root CA	thawte Primary Root CA	7/16/2036	Server Authenticati...
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	7/16/2036	Server Authenticati...

Trusted Root Certification Authorities store contains 17 certificates.

**031 Well, the problem is that the
end-to-end security there--

How Can Superfish Work?

How Can Superfish Work?



CERT | Software Engineering Institute | Carnegie Mellon University

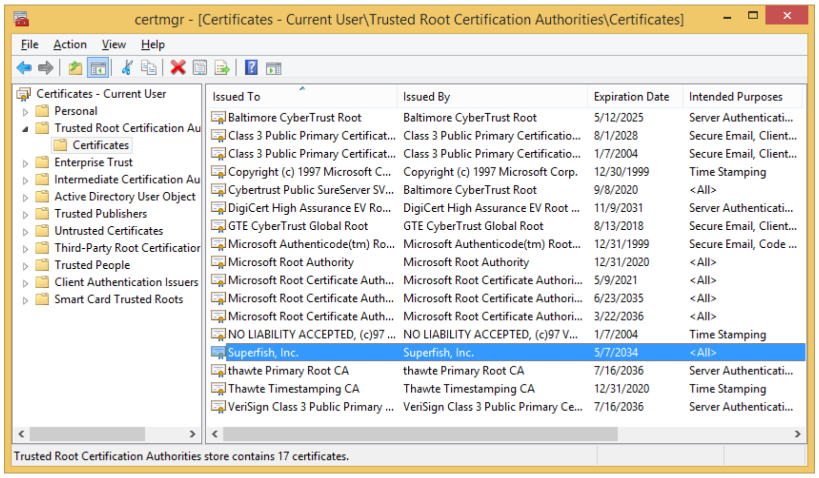
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

30

**030 --Is not guaranteed because there's something happening in between my system and the end server.

How Can Superfish Work?

How Can Superfish Work?



certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificates]

- Certificates - Current User
  - Personal
  - Trusted Root Certification Au
    - Certificates
  - Enterprise Trust
  - Intermediate Certification Au
  - Active Directory User Object
  - Trusted Publishers
  - Untrusted Certificates
  - Third-Party Root Certificatior
  - Trusted People
  - Client Authentication Issuers
  - Smart Card Trusted Roots

Issued To	Issued By	Expiration Date	Intended Purposes
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Server Authenticati...
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/1/2028	Secure Email, Client...
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	1/7/2004	Secure Email, Client...
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping
Cybertrust Public SureServer SV...	Baltimore CyberTrust Root	9/8/2020	<All>
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root ...	11/9/2031	Server Authenticati...
GTE CyberTrust Global Root	GTE CyberTrust Global Root	8/13/2018	Secure Email, Client...
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	12/31/1999	Secure Email, Code ...
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	5/9/2021	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	6/23/2035	<All>
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	3/22/2036	<All>
NO LIABILITY ACCEPTED, (c)97 ...	NO LIABILITY ACCEPTED, (c)97 V...	1/7/2004	Time Stamping
Superfish, Inc.	Superfish, Inc.	5/7/2034	<All>
thawte Primary Root CA	thawte Primary Root CA	7/16/2036	Server Authenticati...
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	7/16/2036	Server Authenticati...

Trusted Root Certification Authorities store contains 17 certificates.

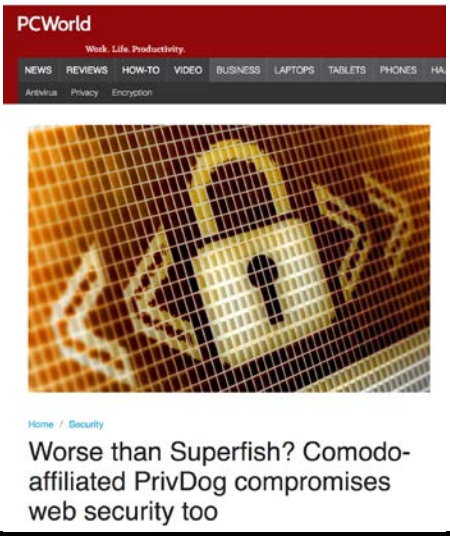
**031 And how is that able to work? So one of the comments that was brought up earlier is, "How do you know that you're on the actual site that you think you're on?" Well, web browsers will generally warn you. If there's a certificate problem, if it's provided with a certificate that's not what it's expecting, it'll tell you, "Hey, something is wrong here." Well, it turns out SuperFish installed on your system-- it installs something that's known as a trusted root certificate authority.

So if I look here-- this is on a particular-- this is a Windows system-- and I can see there is a certificate that identifies itself as SuperFish, Inc. So what that now means is because this is in the Trusted Root Certification Authority section of your system, that now means that this

system will trust anything that is
signed by the SuperFish certificate.

Not Just Superfish

Not Just Superfish


PCWorld
Work. Life. Productivity.
NEWS REVIEWS HOW-TO VIDEO BUSINESS LAPTOPS TABLETS PHONES
Antivirus Privacy Encryption
Home / Security

Worse than Superfish? Comodo-affiliated PrivDog compromises web security too

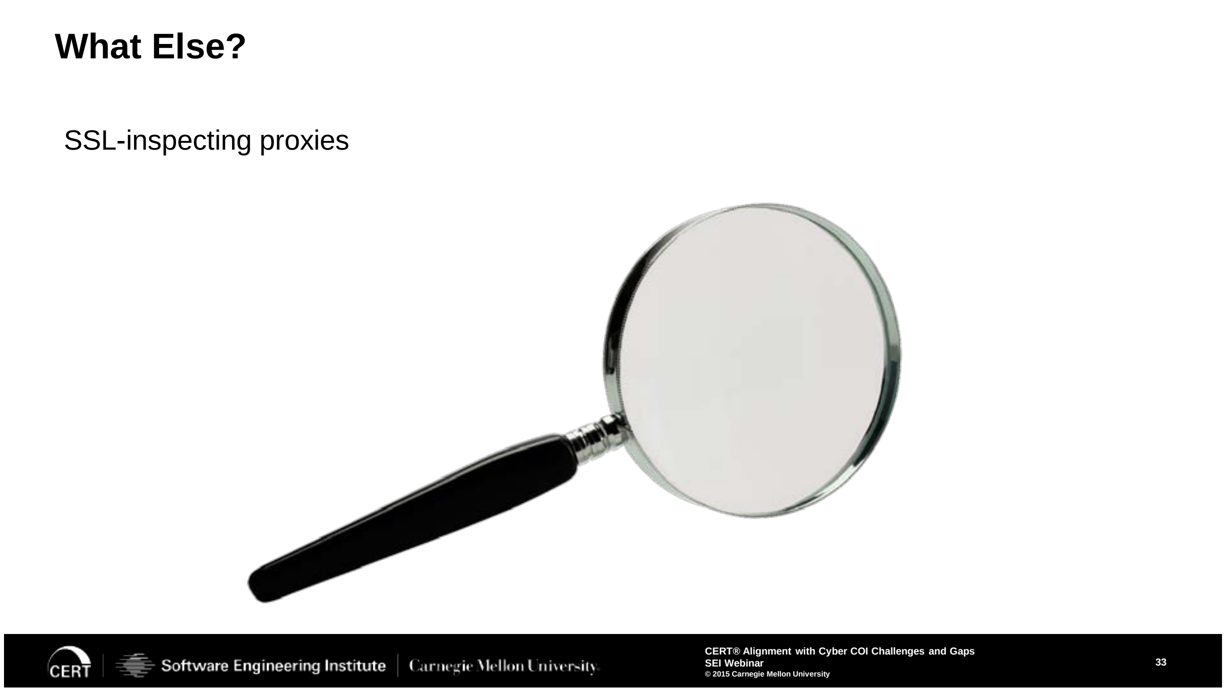
CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

32

**032 We'll get into that in a little
bit here, and it's not just SuperFish.
There were other applications. There
was one that I looked at called
PrivDog, which did just about the
same sort of thing. But that got me
thinking, "Well, okay, we've got
SuperFish, we've got PrivDog. What
else?"

What Else?



What Else?

SSL-inspecting proxies

CERT | Software Engineering Institute | Carnegie Mellon University

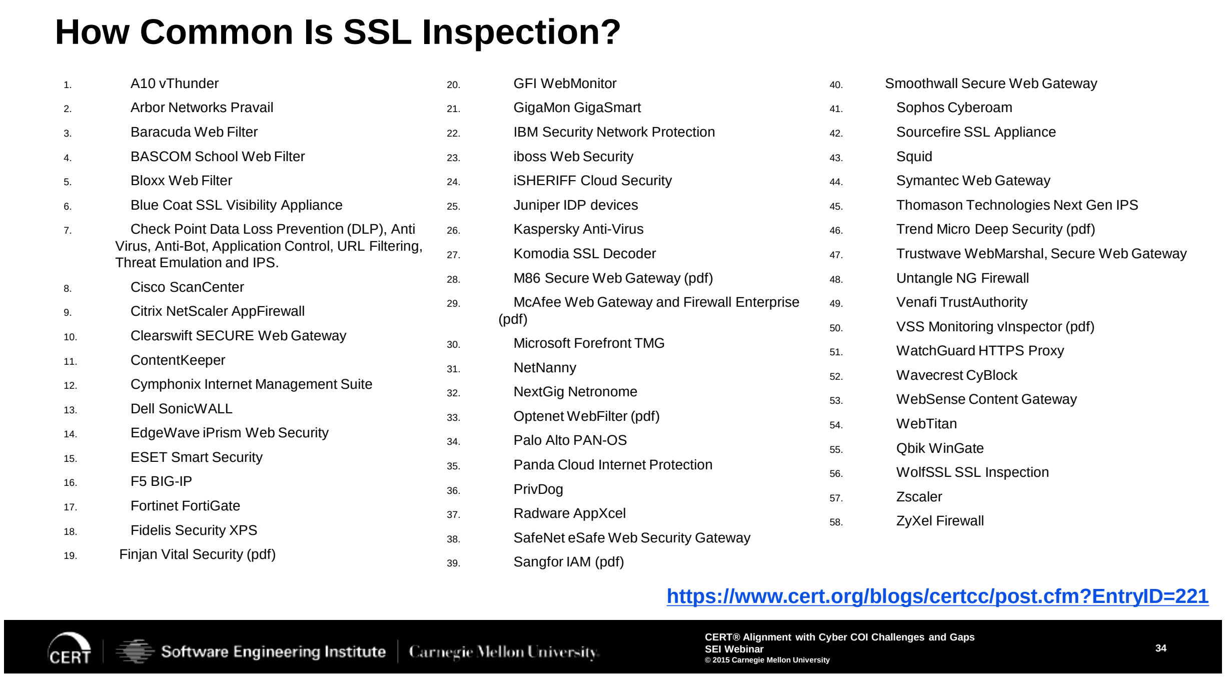
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

33

**033 Maybe there are other things that are doing this sort of interception of SSL.

And it turns out there's an entire enterprise-grade class of products called an SSL Inspecting Proxy. If I am a network administrator of an enterprise network, I might want to see what's actually happening over HTTPS, whether that be Facebook, it could be malware, it could be--there's a large number of reasons why somebody might want to look into SSL traffic.

How Common Is SSL Inspection?



How Common Is SSL Inspection?

1. A10 vThunder
2. Arbor Networks Pravail
3. Baracuda Web Filter
4. BASCOM School Web Filter
5. Bloxx Web Filter
6. Blue Coat SSL Visibility Appliance
7. Check Point Data Loss Prevention (DLP), Anti Virus, Anti-Bot, Application Control, URL Filtering, Threat Emulation and IPS.
8. Cisco ScanCenter
9. Citrix NetScaler AppFirewall
10. Clearswift SECURE Web Gateway
11. ContentKeeper
12. Cymphonix Internet Management Suite
13. Dell SonicWALL
14. EdgeWave iPrism Web Security
15. ESET Smart Security
16. F5 BIG-IP
17. Fortinet FortiGate
18. Fidelis Security XPS
19. Finjan Vital Security (pdf)
20. GFI WebMonitor
21. GigaMon GigaSmart
22. IBM Security Network Protection
23. iboss Web Security
24. iSHERIFF Cloud Security
25. Juniper IDP devices
26. Kaspersky Anti-Virus
27. Komodia SSL Decoder
28. M86 Secure Web Gateway (pdf)
29. McAfee Web Gateway and Firewall Enterprise (pdf)
30. Microsoft Forefront TMG
31. NetNanny
32. NextGig Netronome
33. Optenet WebFilter (pdf)
34. Palo Alto PAN-OS
35. Panda Cloud Internet Protection
36. PrivDog
37. Radware AppXcel
38. SafeNet eSafe Web Security Gateway
39. Sangfor IAM (pdf)
40. Smoothwall Secure Web Gateway
41. Sophos Cyberoam
42. Sourcefire SSL Appliance
43. Squid
44. Symantec Web Gateway
45. Thomason Technologies Next Gen IPS
46. Trend Micro Deep Security (pdf)
47. Trustwave WebMarshal, Secure Web Gateway
48. Untangle NG Firewall
49. Venafi TrustAuthority
50. VSS Monitoring vInspector (pdf)
51. WatchGuard HTTPS Proxy
52. Wavecrest CyBlock
53. WebSense Content Gateway
54. WebTitan
55. Qbik WinGate
56. WolfSSL SSL Inspection
57. Zscaler
58. ZyXel Firewall

https://www.cert.org/blogs/certcc/post.cfm?EntryID=221

CERT | Software Engineering Institute | Carnegie Mellon University

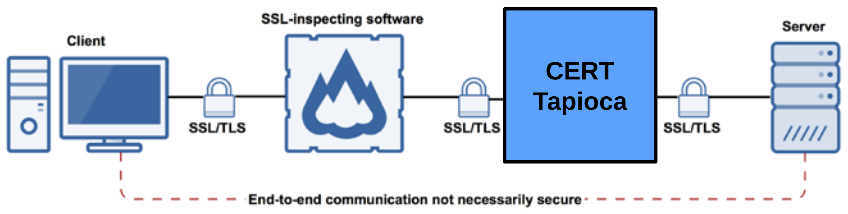
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

34

**034 So I just started to look-- and this is all part of a blog post called "The Risks of SSL Inspection." You can get a little bit more detail about this. But I looked for products that had evidence that they did SSL inspection, and I came up with about 60 different products. I'm sure there are a lot more, but these are-- a lot of big company names in here.

SSL Inspection Software

SSL Inspection Software



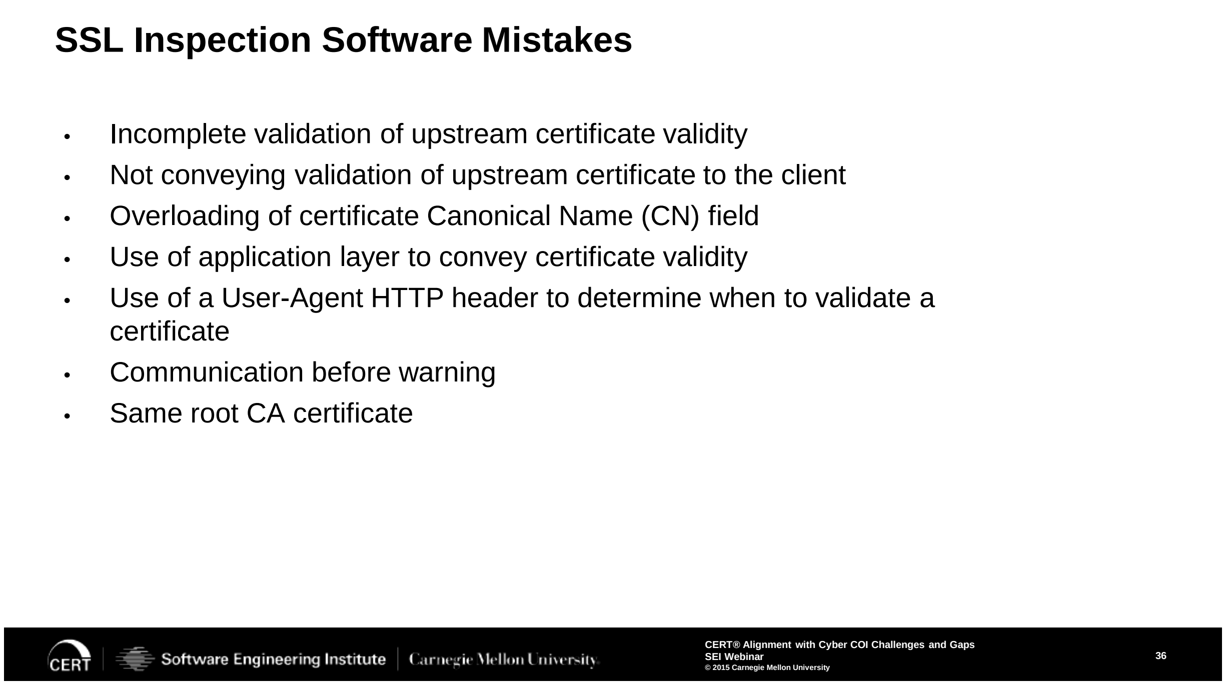
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

35

**035 And so when you're looking at SSL inspection software-- and this might be an appliance, it might be an application that I'm running on a server-- it's very similar to SuperFish in that it's basically injecting itself into the traffic. I can see on the client system on the left I make an SSL or a TLS connection to this particular appliance. On the other side of that appliance, I don't really know what's there. I don't get any visibility into what's happening on the other side. I assume it's doing the right thing, but it might not. I mean, if they're not doing everything perfectly, it's going to put the clients at significant risk. Even if they did everything properly, it still introduces some amount of risk to the client system because you're not actually talking to the server that you think you're talking to.

So I was able to look at some of these particular appliances by just taking my CERT Tapioca appliance and putting it on the far side of that particular application that I was looking at. So basically I was just injecting Tapioca-- so really, in this particular diagram, there are two SSL inspections happening-- but what that allows me to do is I can see, "Does the appliance that I'm looking at that I might have deployed in my network, is it doing the right thing?" If Tapioca is providing it an invalid certificate, is it relaying that back to the client, or is it preventing the client from making that communication?

SSL Inspection Software Mistakes



SSL Inspection Software Mistakes

- Incomplete validation of upstream certificate validity
- Not conveying validation of upstream certificate to the client
- Overloading of certificate Canonical Name (CN) field
- Use of application layer to convey certificate validity
- Use of a User-Agent HTTP header to determine when to validate a certificate
- Communication before warning
- Same root CA certificate

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

36

**036 So as I'm looking at these particular appliances, I noticed a series of mistakes that these appliances are making-- and these are appliances and applications, there's a whole spectrum-- and these are not mistakes that I theoretically came up with and decided, "Well, this is something that somebody could potentially do." These are all real-world mistakes that I have seen in software that is currently deployed on the internet.

One of the things is-- it's a very generic term-- but it's "Incomplete validation of upstream certificate validity." It could be that it doesn't check it at all. If it doesn't check it and it just allows the client to proceed, then that puts the client at risk. Sometimes there are some details where it does some amount of validation but it's not doing it as thoroughly as it should. The other aspect that I mentioned earlier is if it is doing validation, is it relaying the results of that validation back to the client? If it's doing validation but it doesn't relay the status of that validation back to the client, it might as well not be doing the validation at all.

This was "Overloading of the canonical name field." The CN field is a field in a certificate that's used for SSL traffic. We've seen some cases where an SSL inspection application is simply changing the CN field, such as verify_fail.google.com, for example. What they're doing is they're changing the domain name, and if the domain name of a certificate doesn't match the domain name of the site that I'm visiting, the browser says, "Hey, something is wrong here." It might not actually represent what the actual problem is, but it's at least telling the client, "Hey, something is not quite working right."

We've seen some applications that are using the application layer to convey certificate validity, and what that basically means is if I'm using a web server, this particular SSL inspection device might give me web content that says-- in HTML, it says, "Hey, something is wrong with this certificate." Maybe it'll give me the ability to proceed or not. The problem with that is that not everything that talks using web protocols is a human being sitting in front of a web browser. If I've got a client that's making a web request, if it gives me something in the application layer, who knows what the responses might be?

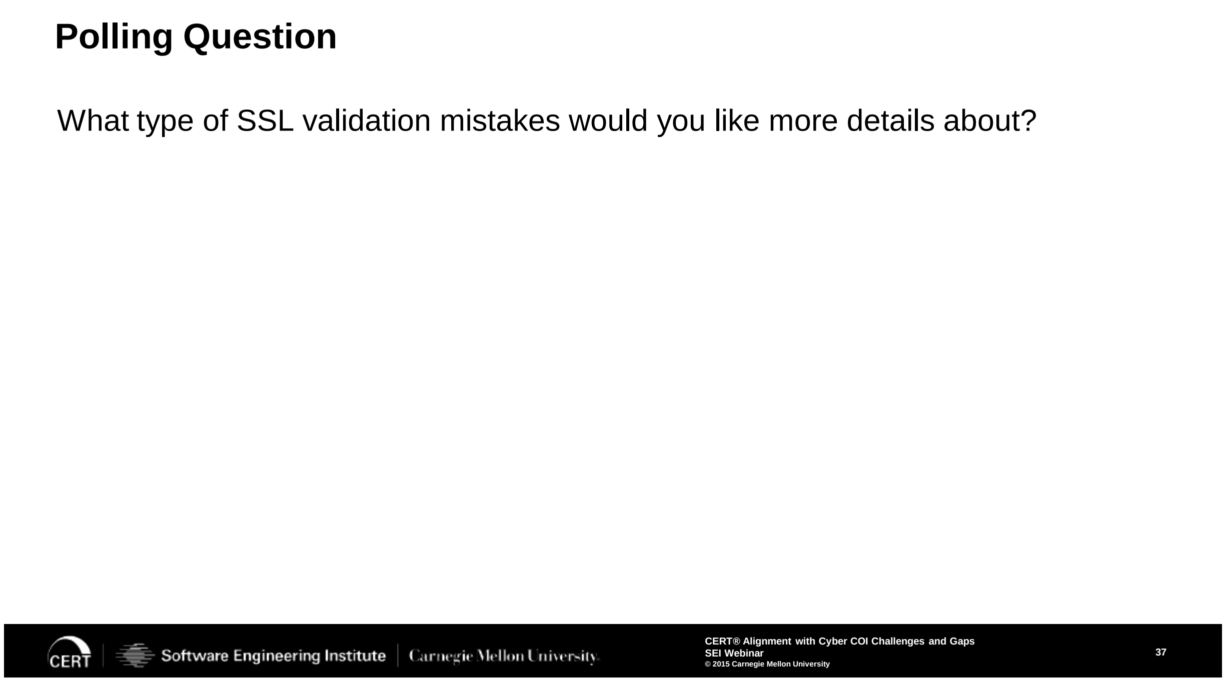
This is one that threw me off a little bit, but some applications are only doing validity checking based on the user agent that is provided by the client. And I was looking at a particular product and my first conclusion was, "This application is not checking validity at all." It was a fully patched system using IE 11, and it just didn't validate anything, and I figured, "Well, okay, this is the case number one." Only upon further investigation did I realize that if I used an older version of Internet Explorer, it did check the validity. So really it had a fixed list of, "Here are the clients that I want to inspect the validity of the certificate for," where obviously that's a problem.

And then the other case that we've got here is communication before warning. I've seen some cases where it'll warn the client, but it's only after it sent a get request to the server, or a post, if you're really lucky. But the thing is, that get request is going to have your session cookie, and if it sends the get request before it enforces the validity, I've now enabled an attacker to potentially steal my session.

And then the last case that I've got is some of these applications, they use the same root certificate authority certificate across all installations.

So if I have an appliance that does SSL inspection and somebody else has that same appliance, if they use the same certificate, that means that an attacker can say, "Okay, well I can just now use this certificate and I can now intercept traffic for any particular client that has this software."

Polling Question



Polling Question

What type of SSL validation mistakes would you like more details about?

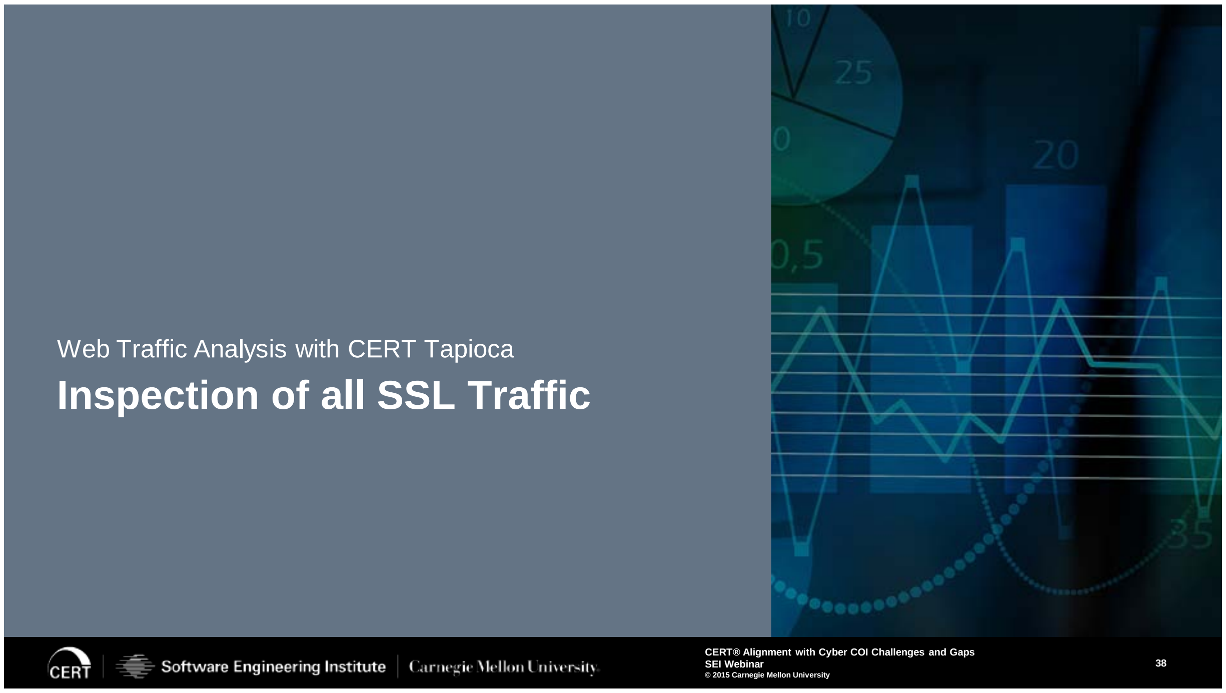
CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

37

**037 Shane: So another polling question to help us drive the flow of the next part of the presentation, Will wants to know, what type of SSL validation mistakes would you like more details about?

Will Dormann: Yeah, so I realized that that list was kind of a quick sort of run-through to determine-- or to just kind of outline the different mistakes that are made, but if anybody has the interest in getting more details about any particular one of those, I can easily follow up and kind of outline some of the details about that.


Web Traffic Analysis with CERT Tapioca

Inspection of all SSL Traffic

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

38

CERT
Software Engineering Institute
Carnegie Mellon University

**038 Shane: Okay, and I'll give you quick results here. So far we got two or 25 percent, Will: Same root, CA certificate, and use of application layer to convey certificate validity.

Will Dormann: Sure. And I think I can actually go into some-- since we've got some answers already, I could probably go into some of the details.

Shane: Please do.

Will Dormann: And the two are relatively straightforward, but it's interesting to see some of the details. So one of the cases, one of the mistakes that are made, is that any application that is doing SSL inspection, it is important that that application generates the root CA certificate that it is going to use at the time of installation. The reason for that is if I have a particular installation of an SSL inspecting proxy, I want to make sure that-- I already realize that I have this-- if I am an enterprise network administrator and I have deployed SSL-inspecting appliances, I've already made the conclusion that I want to-- it puts the client at a slight disadvantage in that they can't actually validate the certificate, but I might choose that because it's a tradeoff.

Now as a network administrator I get insight into the traffic. When I deploy that particular appliance, I would like to make sure that my organization does not use the same certificate as another organization, and the real importance of that-- and it's not necessarily that two different organizations are going to have traffic that are merging with each other and it's going to confuse things that way, but it is really-- from an attacker's perspective, if I know the root certificate that is being used for an SSL-inspecting appliance, as an attacker I now can construct an attack that uses that certificate. So if there's a particular vendor's product that I know all of them use the same certificate, I now know as an attacker I can target every single installation of that particular software or that appliance and the clients are not going to complain, because I'm providing the same valid certificate that is installed on the client.

So just with SuperFish, when you install the SuperFish application on a

Windows system, it installs a root certificate authority certificate in the system. When an enterprise deploys SSL inspection, they do the same sort of thing. They install a certificate on the system. So if that certificate is not unique across installations, that allows an attacker to really target that particular software. You'll have to remind me of the other one that we had.

Shane: I was just going to go back in there to make sure.

Will Dormann: Or if we have any other ones that come up.

Shane: So the other one was a use of a user--

Will Dormann: The application layer.

Shane: Yes. Yeah.

Will Dormann: So the application layer is basically-- if you have an SSL-inspecting appliance or an application that is using the application layer-- and when I say application layers, I don't want to get too much into the details of the OSI stack or anything like that, but basically if I've got a web browser, the application layer is the web content that comes to my browser. So when I go to a webpage, the application layer is the HTML content that provides me text and JavaScript and other things that I can see that my browser interprets. The problem with that is that without any sort of SSL inspection, when I visit a site, the SSL happens on a much lower layer in that it's not even interested-- you've not even reached the point of sending or receiving HTML content.

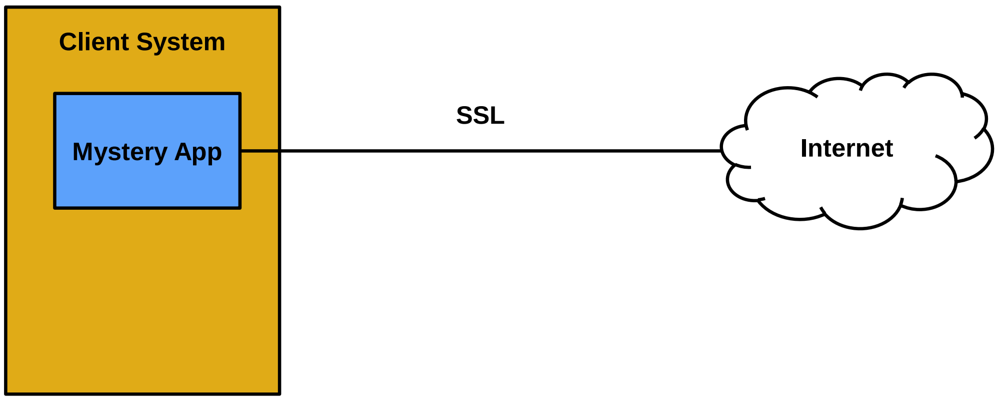
The SSL validation happens before any content is sent or received. The problem with using the application layer is it makes an assumption about the client. It assumes that the client is a human being, or some really intelligent software, that is sitting in front of a web browser. So if I open my web browser and this particular appliance is now injecting HMTL content saying, "Hey, this certificate is not valid," I as a human being can read that and say, "Okay, something is fishy here. I'm not going to proceed."

The problem is things that are making web requests that are not web browsers and they are not human beings sitting in front of web browsers, I might have an application that has an auto-update capability, and that auto-update capability-- I'm not sitting in front of my computer running it; it just manually checks, maybe using a web protocol, "Is there an update available for this particular application?" Now, if this SSL-inspecting appliance is now injecting content saying that this certificate is not valid, but the actual communication is using valid certificate, you can get a wide range of unexpected results. So it's really-- the impact of that depends on the application, but the mistake there is the assumption that, "I'm dealing with human beings that are going to read the message, or they're going to see that red X on the screen and take some action based on that."

So I just wanted to get into the very last-- the two prior sections were focusing mostly on applications that are not doing SSL inspection, or not validating SSL properly. I just briefly wanted to touch on the other operating mode of Tapioca, and that is inspection of all SSL traffic.

Observing SSL Traffic

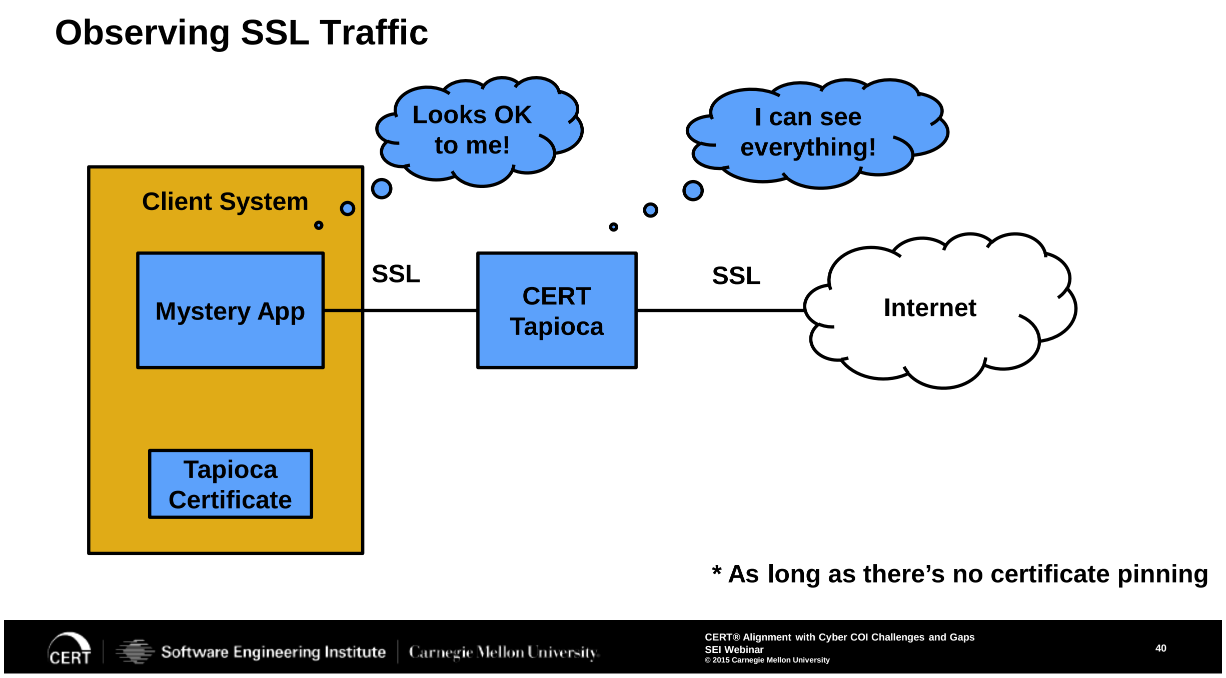
Observing SSL Traffic



**039 So let's say I have one particular application that I'm looking at. I'm not doing things at scale. I'm not looking at a million applications. If I'm looking at a million applications, like the Android applications, I just want to check is it validating SSL or not. However, if there's a particular application that

I'm interested in, I might want to see, is this application sending something over the network that I don't want it to, or that it should not be sending? So if I have this application--

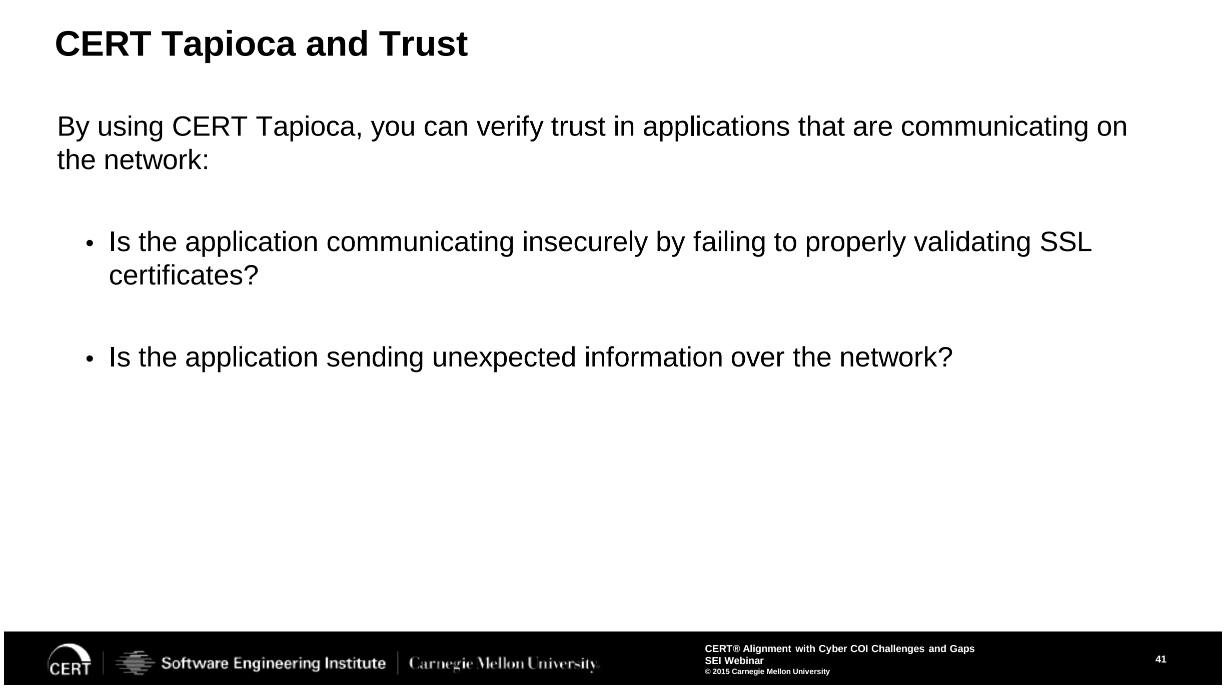
Observing SSL Traffic


Observing SSL Traffic
Looks OK to me!
I can see everything!
Client System
Mystery App
Tapioca Certificate
SSL
CERT Tapioca
SSL
Internet
* As long as there's no certificate pinning
CERT
Software Engineering Institute
Carnegie Mellon University
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University
40

**040 What I can do is I can put Tapioca in line with its network traffic. Same sort of thing: Internet on one side, the application I want to test on the other. The main difference here is that in order to observe traffic, we're going to assume that the application is checking SSL validity. What I need to do is I install the Tapioca certificate-- and I say Tapioca certificate. Really, under the hood it's MITM Proxy-- but there's a certificate on the Tapioca virtual machine.

If I now install that certificate on the client system, assuming I have the ability to do that, my app says, "Well, everything looks fine. I think everything is good because I now have valid SSL traffic." So at this point, where the client assumes that it has valid SSL traffic, I now can see everything with the CERT Tapioca appliance, so I get the ability to see what might be getting sent using web protocols. As long as there's not something called certificate pinning, which means the application is looking for a very specific certificate and not just a valid certificate in general.

CERT Tapioca and Trust



CERT Tapioca and Trust

By using CERT Tapioca, you can verify trust in applications that are communicating on the network:

- Is the application communicating insecurely by failing to properly validating SSL certificates?
- Is the application sending unexpected information over the network?

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

41

**041 So just to tie it back in-- I wanted to connect Tapioca with just the concept of trust-- by using Tapioca you can actually validate-- you can verify trust in applications that are communicating on the network, and that is: Is the application using SSL properly? Is it validating certificates or is it just sending HTTPS traffic without doing that validation? Or, if there's an application that I want to specifically look at and determine, "Tell me everything that it's sending over the network," I can do that with the Tapioca appliance.

Q&A

http://www.cert.org/flocon/

FloCon 2016

Daytona Beach, FL

January 11–14, 2016

CERT | Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University

42

**042 I think that brings us to the end.

Shane: Great. So if you have any questions for Will, feel free to log them in now. We got about three minutes left, so we'll dive right into a question from Robert. And if you work for a credit card, you may want to hold your ears here, but Robert wants to know should he be doing bill-paying by snail mail and get rid of passwords for credit cards?

Will Dormann: Ah, that's a very good question, and it's an interesting aspect. I work on the Vulnerability Analysis Team at the CERT division, and a lot of us are kind of inherently paranoid people-- either inherently when you started work here or after you're exposed to the things that we're exposed to, you become a little bit more cautious about the things that you do. When it comes down to using smartphone applications, those applications are less-- in my experience, they're not as fully tested as web browsers on a standard client.

So if I have a Windows system and I have an up-to-date, fully patched web browser, lots of people have looked at those web browsers and they've put a lot of effort into making sure that they are doing things securely. The problem about an application on a phone is if you don't have CERT Tapioca in line and you're monitoring the traffic, how do you know what it's doing? And that's the case with any application, phones in particular, is you really don't have the same amount of information provided to you that you can make the security decision. So as with anything security, it's a tradeoff. I'm not saying that nobody should do anything on the internet, but--

Shane: I was going to tell you, our audience went down to zero, because you scared everybody off. I'm just joking.

Will Dormann: But yeah, it's a tradeoff. Using things-- using smartphones or other less capable devices for security or financial transactions, it's a little bit of a riskier operation.

Shane: We got about a minute left, so we're going to go one more, from Kim, asking, "Why did you choose to look at Android applications?"

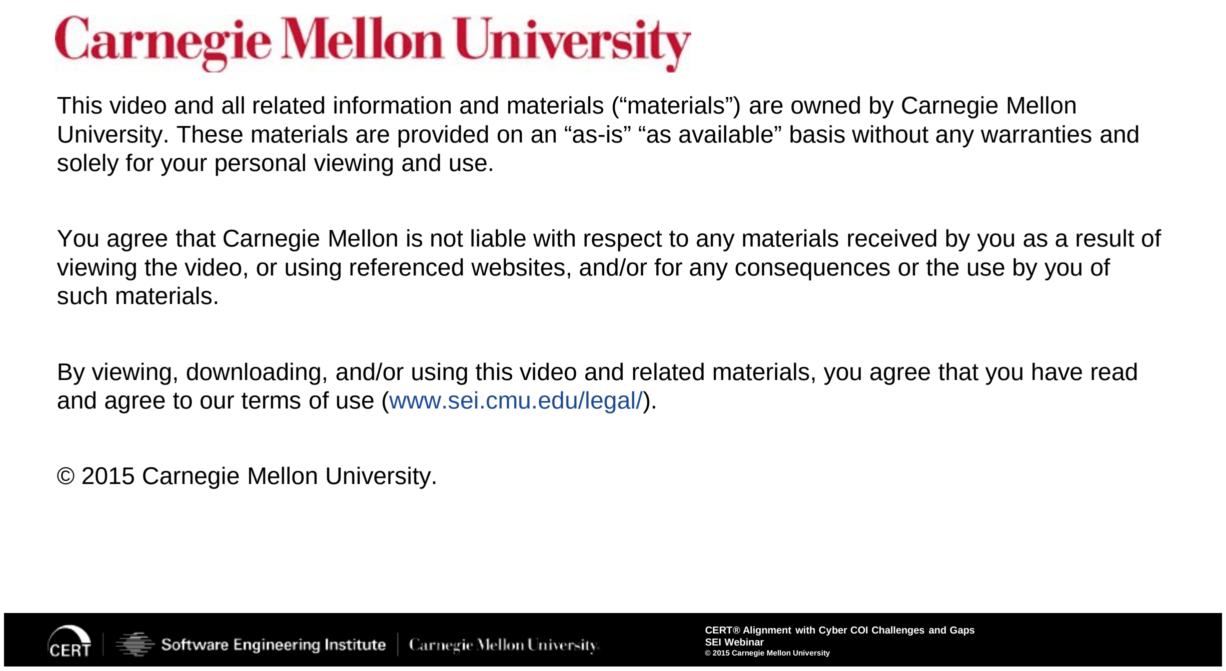
Will Dormann: Oh, that's a good question. Some folks might have looked at the research that we did and concluded, "Well, if Android has this many vulnerabilities and iPhone doesn't, then Android is more secure, and that's not necessarily the case. Really, this was the first application of Tapioca, and the Android platform, being an open source platform, they have emulators available, they've got a lot of capabilities about it that allow you to get applications from the store and actually load them into a virtual environment. It was really the ease of testing that platform. We still have on our radar-- I'll probably get to it, even just starting this afternoon-- but we've got other platforms, such as iOS, that we would like to replicate the same sort of test with.

Shane: Will, great presentation, great tips. Thank you very much.

Will Dormann: Great. Thank you.

Shane: Everyone, that's going to lead us to our first break, so we'll be back at about 10:55 exactly, actually, for Jose Morales speaking on enhancing mobile device security, so you won't want to miss that. We'll be back in a few. Thank you.

Carnegie Mellon University



Copyright 2015 Carnegie Mellon University