

Enhancing Mobile Device Security - Jose Morales

Table of Contents

Enhancing Mobile Device Security.....	3
Introduction	4
Current State of the Art	5
A potential solution	7
Polling Question	9
Prototype	10
Polling Question	12
A potential solution	13
Prototype	15
Prototype testing	16
Snapshots.....	19
www.apk4fun.com/apk/1748	20
www33.zippyshare.com/v/2	21
7.11	22
Saving screenshot	23
Security	24
Fake option	25
Fake option	26
Fake option	27
Fake option	28
Access Detail	29

Fake option	31
Defensive options	32
Polling Question	35
Defensive options	36
Polling Question	36
Offensive Options	39
Modern day ghost device army – WWII operation fortitude	42
Current Status	43
Q&A	44
Carnegie Mellon University	50
Copyright 2015 Carnegie Mellon University	50

Enhancing Mobile Device Security



Enhancing Mobile Device Security

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Jose Andre Morales, Ph.D.

CERT | Software Engineering Institute | Carnegie Mellon University

© 2015 Carnegie Mellon University

**045 Shane: And welcome back to the SEI virtual event, CERT Alignment with Cyber COI Challenges and Gaps. We're pleased to welcome in Dr. Jose Morales who will give our next presentation which is enhancing mobile device security. And Jose is a researcher for the CERT division within the SEI. He has conducted research in cyber security since 1998 with a current research focus on forensic analysis, behavior based malware analysis and detection, suspicion assessment theory and implementation, mobile malware, and malware distribution networks. He graduated with PhD in computer science from Florida International University in 2008. And before coming to Carnegie Mellon, he was a post-doctoral research fellow in the Institute for Cyber Security in the University of Texas in San Antonio.

Now, I'll turn it over to Dr. Jose Morales. Jose, all yours.

Presenter: Thank you very much for that introduction. Good morning, everyone. I'm going to talk to you today about my topic of current research, enhancing mobile device security with defensive and offensive options.

Introduction

Introduction

Talk about

- How mobile devices are currently protected
- What is not protected → private data! And why
- Options to protect private data with fake data
- Prototype construction and testing
- Defensive and offensive options: leveraging GPS and fake data



**046 The things I'm going to talk about is how mobile devices, how are they currently being protected and what is not being protected, which I'll tell you right now is private information stored on the device. And we're going to look at an option to protect private information using fake information. We're going to talk about the prototype that we have, how we've constructed it, how it's been tested. And we're going to

introduce some defensive and offensive options and how you can take fake information along with GPS location and leverage it.

Current State of the Art

Current State of the Art

Mobile devices have:

- Encryption – avoid physical access
- Anti-malware – signature based
- Static analysis tools – post infection
- Dynamic analysis tools – heuristic
- Passwords – typically 4 characters!
- Privacy settings – apps and notifications
- OS level app/process isolation (linux)

Overall not bad **BUT** malware just needs your permission!

- App install all or nothing model eases malware introduction!
 - PII at risk!



**047 So, first the current state of the art. What do we currently have in mobile devices? You can have mobile devices that are encrypted. That only avoid physical access to the phone. You have antimalware, which is mostly signature based. You have stack analysis tools. Those are only useful in a post-infection environment, dynamic analysis tools, which are basic heuristics, not very effective. You have passwords. Typical password on a phone, four characters. Privacy settings, this allows the user to control how apps interact with the phone and what information apps will give to the phone and to the user. We also have,

in Android specifically, OS level app isolation, which means each app runs its own process, memory, its own VM. And they don't interact with each other.

Overall, currently mobile devices are pretty well protected. But the truth is that malware only needs to get your permission to install on your phone. The current way in which apps are installed, they ask the user. Grant these permissions and install the app. Don't grant the permissions, you don't get the app. Well, the user wants the app. So, he has to grant all of the permissions, even if those permissions seem a little dubious and over the top for the purpose of the app. This allows malware to just come in a legitimate way.

In the classic way of malware infections, they exploit vulnerabilities. They use social engineering. Here, none of that exists. The app just has to be chosen by the user because they like what it does. They install it. And that's it. Game over.

What's at risk here is the private information stored on the device by the user. Since the user legitimately allows the app to install and grants all the permissions, it can freely access all of that information. That information currently is not protected by anything that I have listed on this slide.

A potential solution

A potential solution

How can PII be protected from malicious use and exfiltration by an app?

- Purposely provide fake PII! And other data
- Benefits:
 - App cannot distinguish fake from real
 - App will function as expected
 - User can be notified when fake data is accessed
 - User can run any app they choose while protecting their data
 - Too much fake data read? User can uninstall app
 - Data is kept safe!



**048 So, what do you do? Well, a potential solution to protect private information is to use fake information provided by the phone to specific apps. And this can be done when the app is first installed. The user can choose to fake the information that app request from the device.

Some of the benefits of doing this is the app cannot distinguish fake information from real information on the phone. You provide the same data structure with the same information. The app just requests the device ID. It gets a device ID. There's no way for it to tell this is not the device's actual ID. It's being given to the app by the operating system. So, there's no distinguishing features. The app will function as expected.

There are approaches to avoiding apps abusing private information by denying access to information or returning a value equivalent to a null or void. But what that does is you don't know how the app is written. And you don't know if the app requests GPS locations. And you say I'm not going to give you that. I'm going to deny you access to the GPS. How does that affect the app's functionality? It could crash. It could stall. It could completely stop functioning in the expected manner. And that could further put the device at risk. Those are not good options. You want the app to function as expected. And by giving fake information, the app requests, gets what it wants. And it functions as normal.

The users can be notified when fake information is accessed. You're purposely giving it fake information. Let the user know that that's happening. You'll see later we have a flashlight example where it's reading stuff off the device that you wouldn't think it would need for the flashlight.

And the benefit of fake information is the user can install any app they want with the peace of mind that their information is protected. If they're feeding fake information, they're actual information is not being touched by this app to which they're having some doubts about.

Another aspect to fake data is we-- you can keep track of how much fake information is being read. And the

user at one point can decide this app is reading too much fake information. I don't want it anymore. And it can uninstall. So, you actually help guide an undecided user as to should I keep this app, should I not keep this app, by reporting how much fake information is being read.

At the end of the day, by providing fake information, the real information, the private information, the actual, is being protected. And the user can go on using his device without any actual concerns.

Polling Question

Polling Question

Do you want to hear some more benefits of fake data?



**049 Shane: Okay, another polling question. This one from Jose would like to know do you want to hear some more benefits of fake data. So, we'll give you about fifteen or twenty seconds to vote. And Jose, we can move on until we get some results here.

Prototype

Prototype

- Modified Android OS to populate random & relevant fake profiles when a phone is activated
- When an app is being installed we provide a 3rd option, called “fake”
- The app will run but will be given fake data when requested
 - Phone details
 - Photos
 - Contacts
 - Outgoing SMS
 - Geo-location (GPS)
- Providing fake data is preferred to blocking or “null” data since those can cause app to crash and destabilize device
- User notified each time fake data is read and has option to uninstall
- Serves to protect private data while user evaluates apps



**050 Presenter: All right, so continuing the line of fake information, we developed a prototype that creates fake profiles. What we did was we modified the Android operating system to create three or four fake profiles when the device is first activated. We provide an installation option called fake. When you install an app on Android, it usually says grant the permissions, yes or no. We have now yes, no, or fake. You choose fake, the fake option, as we call it, is turned on. And that app will be fed fake information.

The type of information that we are providing fake data for are the details of the phone like the IMEI, device ID, photos, contacts, outgoing SMS, and the GPS, your longitude, latitude. That's what we have so far. But we

found that we can fake a lot more information. We just felt that those were the most critical as far as privacy is concerned.

By providing fake data the way we do, we do not deny an app access to information. And we never give it a null return value. That's very important as we discussed before because this allows a scenario where the app could crash. The device could destabilize.

Every time we built into the system settings a notifier, every time fake information is read, the app reports it. The user can look at it anytime. We're going to see some snapshots. The user can look at the report for any specific app. Not only will it tell you how many pieces of fake information it's read, how many times it's read, but it will tell you exactly which fake information has been read how many times. From an analysis point of view, that's also very useful. That wasn't our primary focus of the work. But it's a secondary benefit.

And it serves to protect private data while the user evaluates the app. As we said, our prototype allows users to install desired apps with questionable permissions. And it gives them an evaluation period where they can try the app, play with the app, do whatever they want with the app with the peace of mind that they're private information is actually not being accessed by this app. It's almost like a protected trial period where I decide at the end, I don't

want it. I'm going to uninstall it. And nothing specific to me or my phone has been put at risk.

Shane: Jose, I'll jump in here real quick with the survey results.

Polling Question

Polling Question

Do you want to hear some more benefits of fake data?



**049 Seventy-two percent would like you to work in some benefits of fake data.

A potential solution

A potential solution

How can PII be protected from malicious use and exfiltration by an app?

- Purposely provide fake PII! And other data
- Benefits:
 - App cannot distinguish fake from real
 - App will function as expected
 - User can be notified when fake data is accessed
 - User can run any app they choose while protecting their data
 - Too much fake data read? User can uninstall app
 - Data is kept safe!



**048 Presenter: So, I'll go back to that slide there. The key thing to providing fake information is you don't want an app to unexpectedly crash on your device. And you want to be able to give users the ability to install any app they want from any source they want. We all know that there's Google Play, and there's Amazon Marketplace. But there's also third party markets. And the third party markets, you get free versions of the apps you would typically pay for. But those free versions, are actually repacked legitimate versions with malware in it.

When a user decides to install an app from any of these third party markets, even from Google and Amazon, which research has shown does not block all malicious apps from being available for download,

the either have to say yes or no. so, if they grant all of those permissions, you're going to have access to all your information in the legitimate, authorized fashion.

But if the user is a little concerned, the fake information will provide them the cover to not access their real information. It provides the app the information it requests. And it doesn't put the device at risk.

Also, on top of that, as we'll see later in the defensive and offensive options, you have fake information that the way we have it, it's automatically populated on the device when first activated. But you could also modify that information to show what you want. And there's benefits there. So, I hope that addresses there.

Shane: Excellent, thank you.

Presenter: What they were looking for.

Prototype

Prototype

- Modified Android OS to populate random & relevant fake profiles when a phone is activated
- When an app is being installed we provide a 3rd option, called “fake”
- The app will run but will be given fake data when requested
 - Phone details
 - Photos
 - Contacts
 - Outgoing SMS
 - Geo-location (GPS)
- Providing fake data is preferred to blocking or “null” data since those can cause app to crash and destabilize device
- User notified each time fake data is read and has option to uninstall
- Serves to protect private data while user evaluates apps



**050 So, we'll continue here with the prototype.

Prototype testing

Prototype testing

Tested with several apps from Google Play, Amazon, and other 3rd party markets

- Implemented in the Android OS source code, very difficult to subvert and detect
- No messages from apps indicating they suspect fake data
- Apps worked as expected, no crashes, slowdowns, or halts
- Other apps not reading fake data worked as expected
- Vast majority of apps request too many permissions and exfiltrated alot of faked data
 - Primarily device specific data such as ID, owner email, and GPS coordinates
- Serves its purpose to empower the user to install and use desired apps with questionable permissions while protecting private data



**051 So the testing of our prototype, we took the Android OS source code and we modified it to implement fake profiles.

We downloaded several aps from Google Play, Amazon, and third party markets. We wanted to make sure these apps would run as expected, would not crash, would not cause unhandled errors, and the device would function as expected, and the user experience was not ruined, while at the same time assuring that when app requested a piece of information that was being faked, that fake information would be given to it without delay and without any OS layer errors.

So, the reason why we implemented it in the OS is because it's very difficult to subvert. If we had done

fake profiles as an app, it could have been subverted by other apps. An app can figure out a way to work directly with the OS and ignore the app. But by building in the security of fake profiles directly into the operating system, you're being proactive. And by being in the OS, it's very difficult for any app to subvert, especially when the way in which the private information is requested by an app is standard API calls through Android OS.

During our testing, we didn't have any messages from apps suggesting that they suspected that fake information was being fed to them. Therefore, there was no distinguishing features. The app could not tell that we gave them a fake email and not the actual email on the phone, that we gave them a fake photo, or we sent out a fake outgoing SMS as opposed to the actual one that they wanted to go out.

The apps worked as expected. There were no crashes. There were no slowdowns. There were no halts. This is very important. The app functioned as expected. The user and the OS and the app were unaware that there was fake information in the background being fed to it.

Other apps on the device that are not being fed fake information also worked as expected. So, when you have fake profiles, you have two separate data structures, one with the fake information, one with the

real. What we found was the apps that were being fed the actual information off the phone worked as expected in spite of having another set of information or here identical to it which was faked. Both the faked information and the real information was fed to their relevant apps with no problems.

One interesting observation, I don't think this is a surprise to anyone, the vast majority of the apps that we tested over request permissions and exfiltrated a lot of the faked information. Things that the app should not be accessing, it did access as fake information. And it would exfiltrate it out. We never followed exactly where it was sent to. We just know that it was sent out of the device, typically via TCP over Wi-Fi.

Some of the main things that we noticed that were constantly read was the device ID, the owner email, and the GPS coordinates. And we reasoned that this had to do with ads for free apps. The ads are relevant to your regional area, gave you a more regionally located advertisement in the hopes to click, make a few pennies.

The prototype showed that fake information did what we expected it to do. It empowered the user to install any app they desire from any location they want. Even if it has questionable permissions, they can install it, run it. And their private information is protected. And the prototype showed that that can be

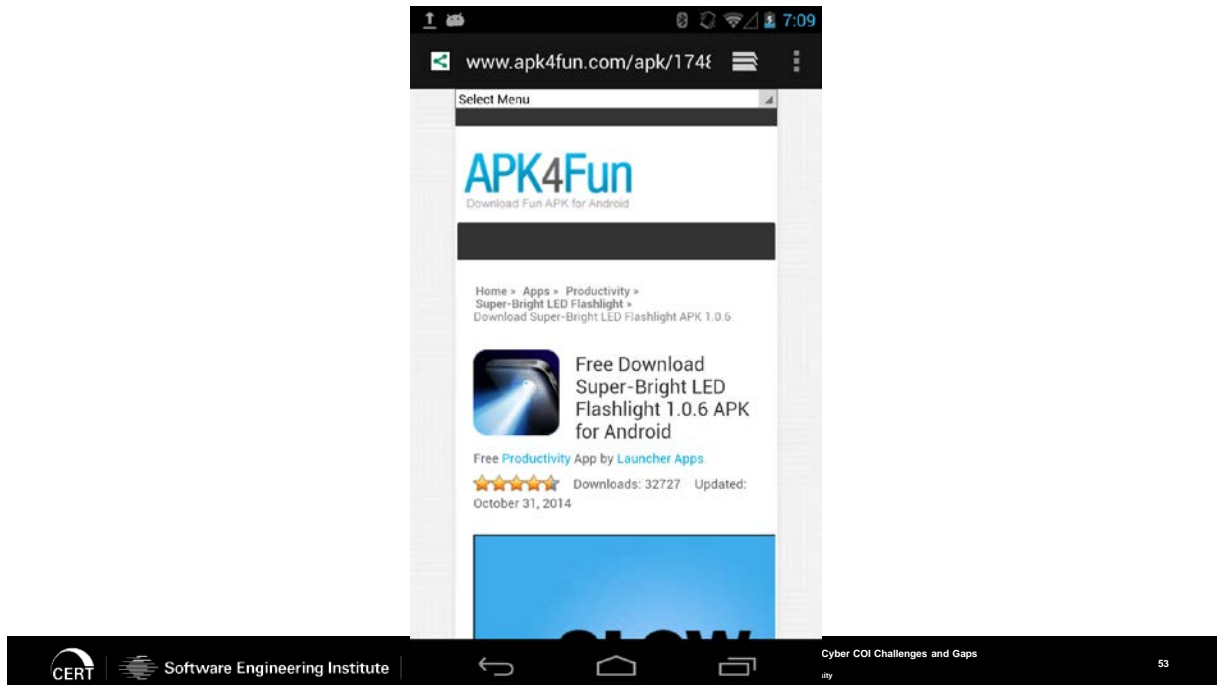
implemented, used, and actually carried out in a practical manner with no concerns, slow downs, crashes, or anything else being negatively affected on the device.

Snapshots

Snapshots

**052 So, now we'll look at snapshots. I love snapshots.

www.apk4fun.com/apk/1748



**053 So, first we chose this app here called Super Bright LED flashlight. We chose this on purpose. It's a very basic app. It should only access the lightbulb on the back on the device to actually function as a flashlight. We got it from a third party website called APK4Fun. This is its download page.

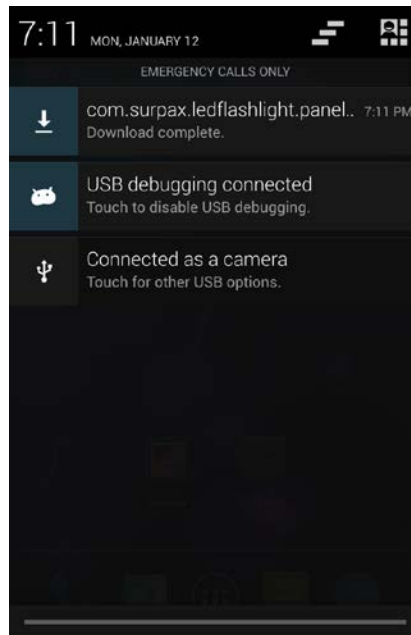
So, you visit this page from your device. You click on the link. Then the download process starts. At this point, it will initiate the device's and your OS's native installation which provides our extra option for fake information.

www33.zippyshare.com/v/2



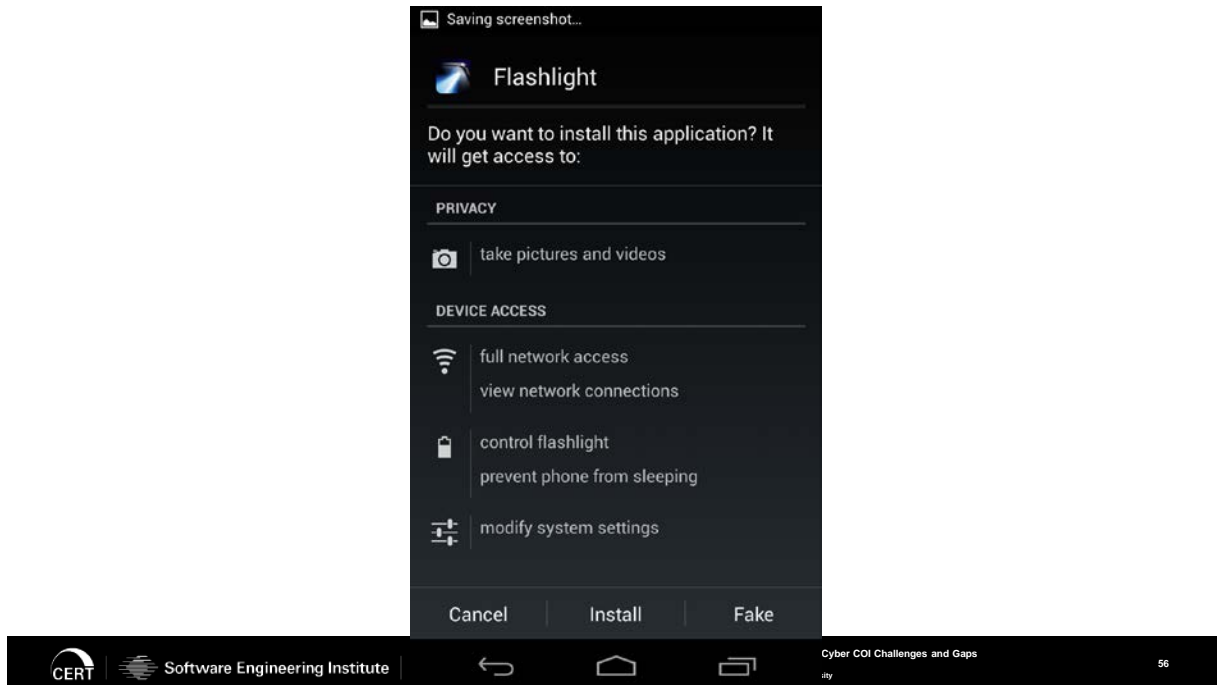
**054 So, here we have the details of it. You hit the download button.

7.11



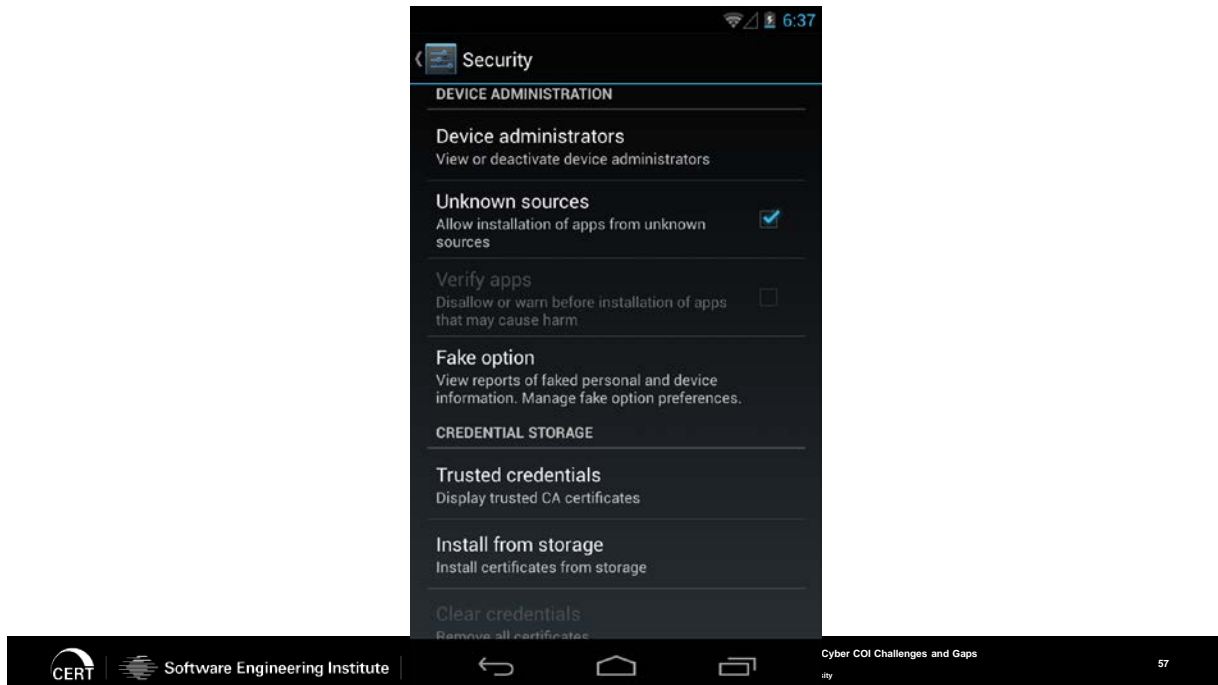
**055 Now, here we see that, at the top, LED flashlight download complete. This shows I've finished. Now, we're going to get the question.

Saving screenshot



**056 It shows information about the flashlight. Here is the typical installation screen. It shows you all of the permissions that are being requested by the app. On a standard Android OS, on this screen, at the bottom, you would only have two options, cancel, and install. But we added the third option called fake. You choose that option, the app will be installed as expected. It's not modified in any way. But on the OS side, we make a little modification to say this app will be fed fake information and not real.

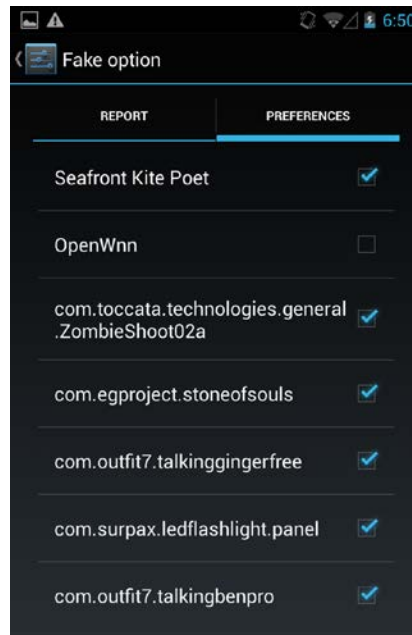
Security



**057 So, on this next slide, we have here the system setting for Android. And if you look, one, two, three, the fourth option in the system settings is called fake option. So, we put this new settings menu here.

If you click on it, it'll show you reports and preferences.

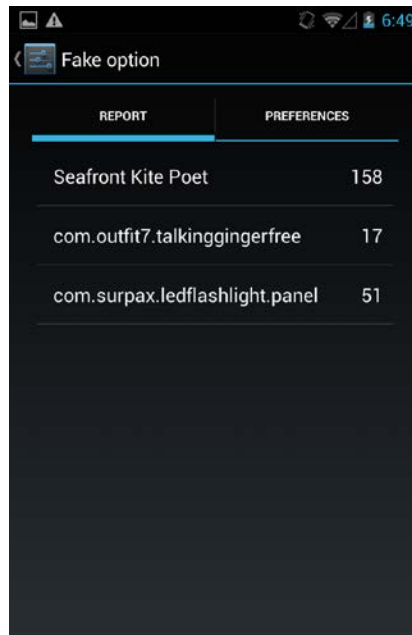
Fake option



**058 The preferences, as we see here, are all the apps currently installed on your device. The ones with the checkmarks are the ones that are being faked. In other words, these are the ones that are receiving fake information.

Now, I presented a few slides ago how to activate the fake option at the moment of installation. But if you're not able to, or you decide not to put on the fake option when you first install it, you can always come to this screen later on and just click and put a checkmark next to any that are not checked. And by putting that checkmark, you will activate fake profiles on that app. And we see here at the second from the bottom is LED flashlight with the checkmark on.

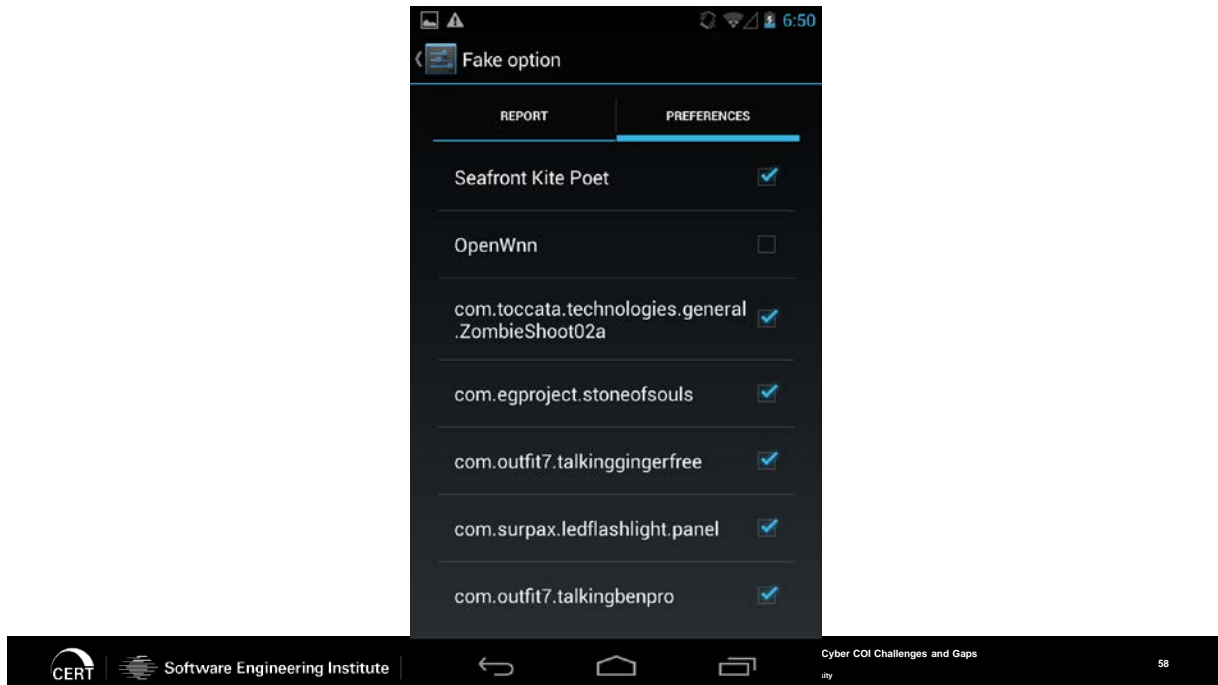
Fake option



REPORT	PREFERENCES
Seafront Kite Poet	158
com.outfit7.talkinggingerfree	17
com.surpax.ledflashlight.panel	51

**059 So, now this is the report side of the system settings. I'll go back one.

Fake option



**058 This is the preferences.

Fake option

REPORT	PREFERENCES
Seafront Kite Poet	158
com.outfit7.talkinggingerfree	17
com.surpax.ledflashlight.panel	51

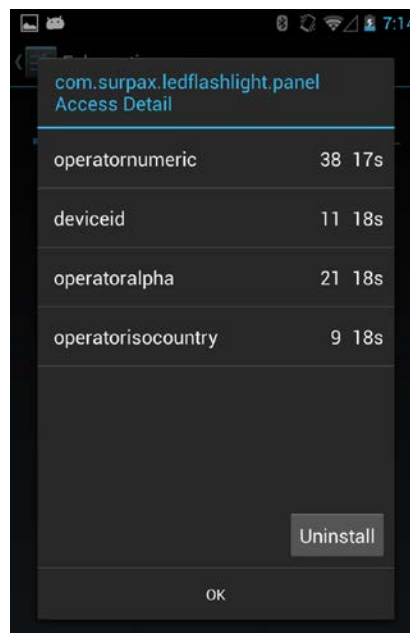
**059 This is the report side. What you see here are a list of the apps that have read fake information since the time they were activated to read fake information. For example, we have here Seafront Kite Poet 158. That means it has read a piece of fake information a hundred and fifty-eight times. Second one, we have talking ginger free, that's read seventeen times fake information. And the last one is our LED flashlight app. That's read fake information fifty-one times.

Now, it's important to notice that, from the time we took this screenshot, from the time we installed the app to the time we took this screenshot there is about thirty-five seconds had gone by, maybe no more than fifty seconds had gone by with us going between the screens.

In that time period, this app had already read fake information fifty-one times. So, it's important to notice, from an analysis standpoint, a high volume of reading in a short period of time should make you a little suspicious about the app, especially if it's a flashlight. It should read anything.

So, now if you click on any one of these app names in the report--

Access Detail



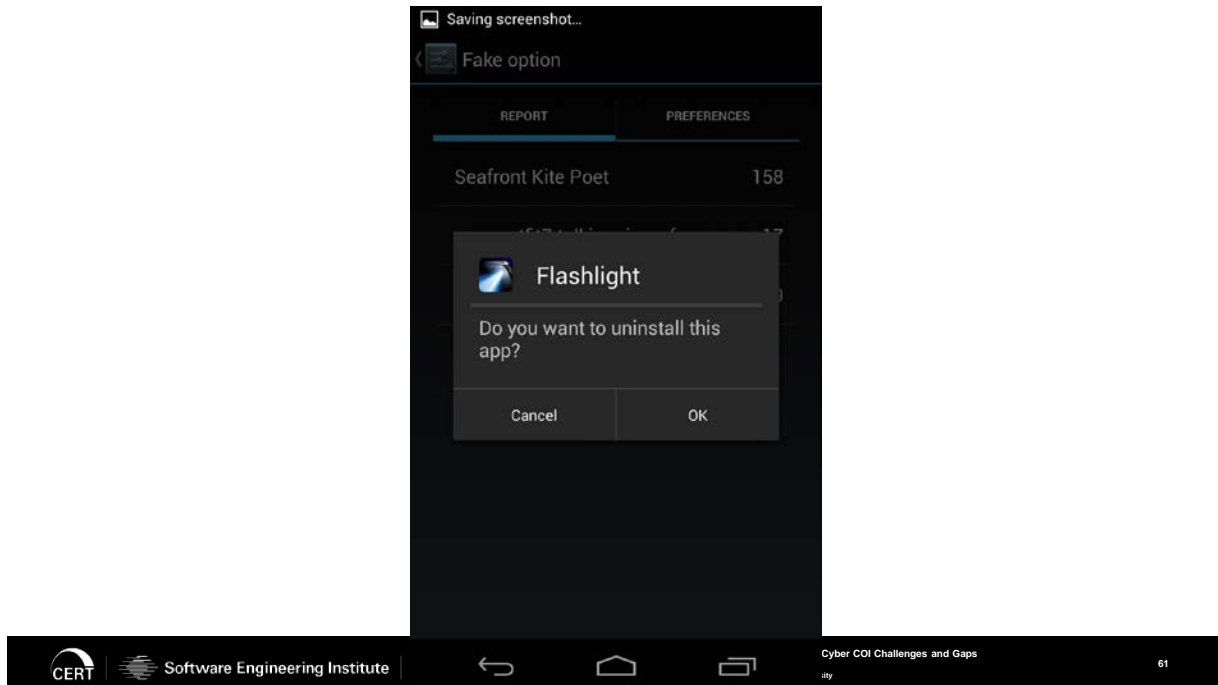
**060 You'll get details about it. Here is a screenshot of what happens when you click on the LED flashlight panel. It shows you the specific pieces of fake information that have been read. And these slides have a prototype where the names were the actual names that Android OS uses. It's not very user-friendly. But since then, we've modified them. So, the

value has a name that's more understandable by anybody.

We can see here for each value you have next to it the number of times that it's been read and the last time in which it was read. So, for example, second one down, device ID has been read eleven times. And the last time it was read was eighteen seconds ago. We have here, the last one is the country of the device. That's been read nine times. And the last time it was read was eighteen seconds. So, we want to provide not just a number of times each piece of information has been read, but we felt that showing a timestamp of when it was read last was important to give the user the idea if it's being active or not, if it just reads it once and that's it, or if it continuously reads.

The last thing you want to notice on this screenshot is at the bottom, you have the uninstall button. That's also something that's not available. We put that in there. The reason why that uninstall button is there is when the user comes here and sees that their app is reading too much fake information, they've decided they don't want it anymore, we give them the option to uninstall it right away as opposed to having to go and follow some other process that Android provides. It's easier. And it's direct, and lets them get rid of the app right on the spot.

Fake option



**061 So, here, this is what you get when you click on the uninstall button. You get the standard Android uninstall process, okay or cancel. You choose okay, it will be removed.

Defensive options

Defensive options

Leverage GPS and fake data

- Based on device's current location (GPS) enforce fake profile policy
 - Specify fake data as input to potentially all apps
 - Provide crafted fake data
 - Gives the illusion the device is somewhere else
 - The data is "boring" and unusable
 - System specs are not compatible: fake Android OS version
 - Result is app becomes disinterested, device and its data not a target anymore
- Useful when device is out of home base
 - Another state, region, country
 - Civilian workforce abroad!



**062 All right, now we're going to talk about some defensive options. We chose to take fake information. And we chose to take the device's GPS location. These are two very powerful data points on the device. What we did was, based on where the device is currently located, we can enforce fake information profile policies. We can specify fake information as input to potentially all apps.

What this means is we create a security policy that checks where the device is current location is. Based on that location, we can say I want fake information given to all of the apps, not just the ones that are currently made to follow fake information, to be fed fake information for everything. And the scenario that we had in mind was the device is in

location way out of its usual, what we call the comfort zone, in another part of the country, in another country, across the world. And there are security concerns. This device is normally not in this part of the world. So, let's fake everything because we don't know exactly where it is. There might be some questions about how or what is being done to it.

We can provide crafted fake information. If the device falls in a GPS coordinate that's outside of our comfort zone, or of what we decide to be a safe or a green zone, we can purposely not just provide fake information, but specific fake information for specific reasons, such as to give the illusion that the device is not where it actually is.

Let's assume your device is on the opposite side of the planet of where you currently are. You can create a profile that says if it requests the GPS coordinates, and we're on the other side of what we consider to be a green zone, don't give it the current information. Don't give it fake information. Give it these specific coordinates. And those specific coordinates may actually be of back home or anything else you want, something that will not entice or call the attention of the people reading or requesting that information.

What you want to do is you want to provide crafted information that's boring and unusable. Boring and unusable is whoever's collecting it will say this is not for us. The hope is

they'll just let it go. And they'll forget about that device, not touch it again.

If it requests information about the Android OS or the device itself, you can give it fake information about that, too. What we did was we give it information that gives it an older version of Android, a different device model, a different device type, anything that makes the device look like something that's old and probably not attractive to the people collecting the information. The point of all that is to have people stay away from the device so you can defend the information.

Hopefully, the result of this is people become disinterested. They become disinterested with the app, the device, and the data on it. And it's not targeted any further. That's important.

This type of approach is very useful when a device is out of home base, which we call the green zone. You can be in another state, another region, another country. An example is a civilian workforce abroad when they take their private phones. They're doing contracts in other parts of the world.

Polling Question

Polling Question

Would you like an example of “out of home base”?



**063 Shane: Okay, we're going to launch another polling question just to help drive the flow a little bit of the presentation. And again, Jose would like to know would you like an example of out of home base. And while you're voting on that, just a quick reminder to everybody to visit the files tab that's found in your console. And you can walk away with a PDF copy of all the presentation slides today along with CERT and SEI work in the cyber security space. And let's see what our results here, Jose. Seventy-one percent yes. So, can we get another example in?

Defensive options

Defensive options

Leverage GPS and fake data

- Based on device's current location (GPS) enforce fake profile policy
 - Specify fake data as input to potentially all apps
 - Provide crafted fake data
 - Gives the illusion the device is somewhere else
 - The data is “boring” and unusable
 - System specs are not compatible: fake Android OS version
 - Result is app becomes disinterested, device and its data not a target anymore
- Useful when device is out of home base
 - Another state, region, country
 - Civilian workforce abroad!

**062 Presenter: Sure, so--

Polling Question

Polling Question

Would you like an example of “out of home base”?

**063 Currently, U.S. military has

bases and embassies, the government has embassies, the military has bases all over the world. Part of the workforce in those bases are civilian contractors. Now, civilian contractors travel to a country to work under contract on a military base. They have their work phones, their work devices. And they have their personal devices. Personal devices are used on personal time.

So, now you take a civilian contract employee working in a foreign country where we have an embassy or a military presence. The country supports our presence there. But they don't necessarily agree with us, or don't necessarily like us to be there. They just support the fact that we're there. So, this civilian contractor on his personal time goes to a cafe. Every cafe I most other countries have free Wi-Fi. He gets on the Wi-Fi and uses it for Skype, or FaceTime, or What's Up, or anything else to talk with family, to talk with friends. While he's on that Wi-Fi in that cafe in a country that doesn't necessarily agree with our presence there, that Wi-Fi signal can be maliciously read and recording everything coming out of the phone.

In those scenarios, if there's no fake information on there, the apps to which the user, that civilian employee, is running can be sending out private information into the hands of people it doesn't recognize. And we don't know what the result of that information could be, intelligence, coordinates, anything. It

could give away, to people that we are not in favor of, where we are, our locations, by just knowing that once civilian employee contracted by a specific company, his device, or her device in a specific part of the world is a very valuable gem of information. In those scenarios, providing fake information, purposely telling anyone when you're on a Wi-Fi in an unrecognized region that you're not really there, that you're somewhere else, along with fake information about you and your device, if that Wi-Fi is being maliciously monitored, they will be disinterested and possibly walk away.

The key thing to remember is the people who are looking at the Wi-Fi signal may not physically be there. You don't know where they are. All they see is traffic on a Wi-Fi. You provide crafted fake information, they could become disinterested. And that person is safer along with the rest of the civilian workforce and our interest in that region of the world.

So, I'll continue now.

Shane: Excellent. Thank you.

Offensive Options

Offensive Options

Purposely mislead the enemy with specific fake data

- Faking data can misinform the enemy to think what we want them to think!
- Used as a tactical device in an operation with preinstalled app:
- App designed to transmit via wifi crafted fake data based on GPS location
 - Fake GPS to predetermined location(s)
 - Mislead and position enemy to our advantage
 - Move enemy focus to bogus theatre
 - Fake contact details that we control
 - Discover their IPs
 - Cyber offensive capabilities
 - Compromise their systems
 - Fake photos and outgoing SMS messages
 - Misleading intelligence creates beneficial opportunities
 - Other limitless ideas!



Software Engineering Institute | Carnegie Mellon University

CERT® Alignment with Cyber CQI Challenges and Gaps
SEI Webinar
© 2012 Carnegie Mellon University

64

**064 Presenter: So, now let's look at some offensive options. The key thing with fake information is you can purposely mislead the enemy with specific fake information. This is very important. Faking information can misinform the enemy to think what we want them to think. And that's key. You can use this as a tactical device in any sort of operation with a preinstalled app.

The scenario is this. You have a device. You put an app on the device designed by us that transmits specific fake information depending on where the device is located.

You take that device. You issue it out into the field in any capacity you see possible. It could go out in any type of operation, any part of the world. Depending on where the device

lands, the app that we've designed starts to purposely send out crafted information that we predesignated and preinstalled on the device. You can abandon the device. Or it could be placed in a tactical scenario.

When this happens, if the Wi-Fi or the signal that app is using to send out the fake information is being read by others, you're going to be feeding them fake intelligence is what it is. You can give things like fake GPS locations to make the enemy think that we're somewhere where we're not. This can mislead the enemy and give us an advantage to better prepare for any type of operation.

You can have the enemy focus on a theater that's not real, or a bogus theater. If you're in point A, and the app is saying you're in point B, and they pick that up, they're going to focus on B. But you're really in A is where you are.

The other thing we can do is we can fake contact information that we control. This is more of an intelligence aspect. By giving fake contact information, if they start to exploit the contact information that they've gathered from the device, you can pick up things like the IPs that they're working from. You can cyber offensive capabilities.

Let's say you give them a fake email address, or fake website, or a fake phone number. And they reach out to it. We can record where it came from. And now we can explore that

number and gather our own intelligence from it. If it's an IP address, we can hack back onto it to try to see what we can do there.

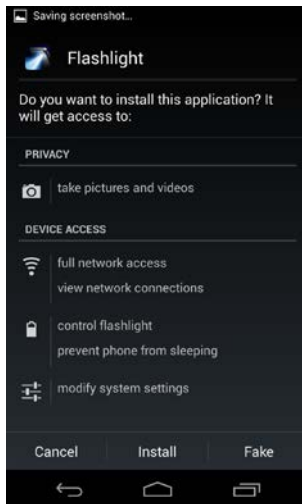
At the same time, they think they're getting real information. But they're not. They're getting what we want to give them. That's the key. And you can further compromise their systems.

The next thing you can do is you can provide fake photos and outgoing SMS messages. This can mislead intelligence which creates opportunities for us. If we tell them, if our app tells them the device belongs to a specific person of a specific role, and the apps relate to that person's role, they're going to think they're getting their hands on intelligence of pictures and messages relative to our operation in that region. By us purposely faking that, we can make them think differently. And that gives us a strategic advantage.

As far as other offensive options are concerned, there's limitless ideas as to how you can leverage the type of fake information you want to send out from the device depending on where you are in the world. It's limitless to your imagination.

Modern day ghost device army – WWII operation fortitude

Modern day ghost device army – WWII operation fortitude



**065 Now, when you think about this, what we put together was by modifying devices in this way, providing these apps that purposely send out crafted fake information that we predesignate and put out in the field, what you end up with is a ghost device. You can make a whole army of ghost devices where each device is controlled by us, set up by us, sent out, deployed by us for the purpose of misleading our enemies to our advantage.

And this reminded me of the ghost squads of World War I and World War II where they would make fake tanks. As we see here, we have a picture of an inflatable Sherman tank being used in World War II for operation Fortitude. What did that tank do in that operation? We landed at Normandy in France. Those tanks

were set up in a different location to make the Nazi's think we were somewhere else. And it gave us the advantage to better plan and face less of defense when we came in.

With that--

Current Status

Current Status

Fake profiles

- Fake data prototype up and running
- Not publicly released yet, seeking partners and evaluators

Future work

- Patch updates for fake data to commercial phone versions
- Ghost devices only in a design phase
- iOS
- Seeking stakeholders

**066 I'm going to give finally the current status of what we're doing. This is my last slide. The fake profiles, the prototype is up and running. It hasn't been publicly released yet. We're seeking partners, evaluators, stakeholders.

Our future work, the fake profiles that we have, we want to create patches. We've only tested this a little bit. We want to create patches. You can apply fake profiles to any commercial phone running Android OS. And we want to make those

available to partners that we work with.

Those devices have been designed but not implemented. We're looking for stakeholders who are interested in further pursuing that. And we want to take our work from Android OS and apply it to iOS, which the operating systems at a high level are different, but they both sit on the Linux kernel. They both can be used in exactly the same way that I've provided here. They both have the same model of grant all permissions, install, grant no permissions, don't install.

And we're seeking stakeholders. So, if you're interested in any of this, reach out. My email will be on the next slide.

Q&A



Jose Andre Morales – jamorales@cert.org

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

A banner for FloCon 2016. The left side has a blue background with white text: 'FloCon 2016', 'Daytona Beach, FL', and 'January 11-14, 2016'. The right side shows a dark blue background with glowing yellow and orange lines and a bright light source. At the bottom, there are logos for CERT, Software Engineering Institute, and Carnegie Mellon University, along with text: 'CERT® Alignment with Cyber COI Challenges and Gaps', 'SEI Webinar', '© 2015 Carnegie Mellon University', and the number '67'.

**067 Shane: Okay, before we start our Q and A with Jose, I just

want to remind everybody or anyone that's just joining us for the morning to make sure you fill out that survey upon exiting as your feedback is always greatly appreciated. And we'll get into the questions for Jose. Some we should just answer, Jose, but we may have to repeat some.

From Brent asking if possible, could you please differentiate between fake and camouflage.

Presenter: Fake and camouflage, I've never heard the term camouflage. Just thinking about it a little bit, I think they're probably both-- camouflage might be real information that's encrypted and ciphered, sort of made to look to hide the information. But fake information is an actual different value completely separate from its real counterpart. So, let's say you have the device ID. You have the device's actual ID. Then we provide fake ID, two separate pieces of data that represent the same thing.

I'm not sure. I apologize for not knowing exactly what camouflage is. But it seems to me a camouflage would be the real device ID sort of hidden or modified slightly to look a little different. And if it's like that, we have two separate values. Camouflage might just be one value.

Shane: And part B may help here, too. It says historically, it has been possible to identify some categories of fake identifiers in oh one. How does one ensure fake identifiers are not immediately obvious?

Presenter: Well, what you want to do is-- and the GPS location is one good example of this. You don't want to have information that's way too far off from where you really are. So, if you have someone on the ground who knows that a physical device is in a certain place, and they pick up a Wi-Fi signal, let's just say California for example, if you're in California, the people know it's somewhere in California, but your device is sending back that you're in Florida, that's an obviously flag. What you want to do is you want to have things that are relatively in the same region but not specifically in the same region.

So, if you're in Los Angeles, California, you can send out things that say that you're in San Diego or San Francisco or even Portland, Oregon, or Seattle, something relevant within the region. And for every other piece of information, things like device ID, email. Or email, you want to have a fake email address but that is active and usable. You don't want to have something that doesn't exist and bounces back.

For other information, photos, contacts, you can make relevant phone numbers and emails that are active and can be reached out to as opposed to things that, if they're used, are blatantly not valid. That's what you want to avoid.

Shane: Okay, we've got two questions asking about iOS, when it will be available, when Differentiator- will Apple Lock make this not possible to do on iOS?

Presenter: So, all the other types of security that have been put out on these devices is to avoid access to the device, avoid access to the information. But we built fake profiles under the assumption that the app-- the device is being used legitimately. And someone downloads an app that has been granted all of its permissions and therefore is authorized to get all of the information. So, anything you put in the way of accessing information, the user using this app with these questionable permissions will give access to it because they chose to install the app as opposed to malware in the past that sort of hides in the background and exploits things to get access in a stealthy hidden manner. This is wide open. So, there's no blockades between the app and the information. Therefore, anything like Apple Lock, as long as the user allows that app to access the information, it will get it.

Shane: Okay, can we put up Jose's email address again when we get a chance as well. We have a request for that. Next question from Frank asking how can we obtain the Android OS that permits fake identification?

Presenter: Just reach out to me, to my email address. And we'll talk offline.

Shane: Okay, the next we've got just another question about is there an estimated timeframe for the fake application to be ready for iOS or non-Android phones?

Presenter: IOS development hasn't started yet. The reason is we've been focusing on the Android version. If you are interested in the iOS version, which is what we want to pursue, we're seeking stakeholders for that work. Reach out to us. Express your interest to us. And then we can put a plan together to implement it. The approach is-- abstractly, the approach is the same. The implementation is just slightly different.

Shane: Oaky, question from Dawn asking are there any legal ramifications of using the fake data that you've come across or any issues there that you're aware of?

Presenter: We haven't considered legal aspect of this. But if I have my email address, and then I create a second email address for spam, and when I log on and join mailing lists and websites I don't use very often, I put the spam email address because I know I'm going to get inundated, is there a legality issue there? I don't think so. It's the same thing on the device. I have my legit email address. Then I have another email address that's a legit address, but it's fake. It's not mine. The fact that it's fake means it's not my email address. But it's still a valid email address you could use. It's just it's not my private one. I don't think there's a legal issue in that scenario.

Shane: Okay, question from Raj asking will this work on any version of Android OS?

Presenter: One of the things that we want to do is create patches for any Android OS version. When you use the Android SDK, you can build it for specific devices, specific OS versions. If you want us to test it on yours, reach out to me. And we can-- we'll make a patch. And we'll provide it. And we'll test it before handing it out to you.

Shane: Okay, folks that's all the time we have for this presentation. Jose, thank you very much for your time.

Presenter: Thank you for inviting me.

Shane: Great presentation. Folks, we're going to take a break here from eleven thirty-five to twelve ten. So, we can catch a lunch break here for the East Coast. We'll be back promptly at twelve ten for SEI COO Robert Behler giving a presentation that you don't want to miss. So, we hope you join us back at that time.

Carnegie Mellon University

This video and all related information and materials (“materials”) are owned by Carnegie Mellon University. These materials are provided on an “as-is” “as available” basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

© 2015 Carnegie Mellon University.



Copyright 2015 Carnegie Mellon University

Copyright 2015 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0002555

