Corellium

Video Webinar

Episode 1: *Corellium 101: Mobile AppSec Testing with Arm Virtualization*

Full Transcript

00:07)
Brian Robison(:
Good morning, good afternoon, good evening, depending on where you're all coming from. There's a lot of great folks joining in today and I really appreciate you taking the time out of your busy day, your busy schedules to come in and kind of learn a little bit about Corellium and what we do and how we're different. I promise that today's session will hopefully be a good interactive session. I like lots of Q&A. If you wish to ask questions throughout the webinar, please do so. I can keep an eye on them a little bit on my screen and would love to have some interaction. And I actually have a couple of polls that we can do today, but I just wanted to welcome everybody, and thank you again very much for coming in this morning and this afternoon or evening and hanging out with me.

(01:00)
And so my name is Brian Robison, I'm the Chief Evangelist here at Corellium and it's basically my job to be out talking with folks like you and kind of showing you what we do. And so today's session is going to be a few slides, but mostly I'm going to be showing you the actual product and doing some demonstrations for you. So you're actually going to spend most of the time today actually in the technology. So again, I look forward to your interactions. I do have some time towards the end set up to address questions, but please feel free to ask them as we go along if you have questions and we can address those. And then if we don't get to your question during the webinar, we'll get back to you as quickly as we can. I would like to begin, because this is our first inaugural webinar from Corellium. I would first of all like to get a little bit of information from what you guys are looking for. I've got a couple of polls here that I'm going to bring up to start today's session, and I'd love to get what level of experience do those of you on the phone or on the webinar today have with the Corellium platform?

(02:37)
I'm getting a question coming in where people can't hear me. It says I'm transmitting. If you can't hear me, please just drop a question in or if you can, let's let me know please. Okay, so I'm going to go ahead and end the poll and let's take a look at the results. So the great thing is, and

that's kind of the design about a webinar, we have a wide variety of experience on the webinar today and that's great because even though this is kind of titled as a Corellium 101, those of you who are experienced using the product might see some use cases that maybe are outside of your expertise area. So I really appreciate you taking the time and filling out that poll for me. So I am getting some questions about these old machines in the background. It's something I've done for quite a few years, but I love the SE/30 on my left and I've got a IIci on the right.

(03:59)
They're actually running 30-year-old operating systems essentially. So it's fun, it provides good stuff in the background and I actually have some neat Corellium logos flying around. So I like to have a little bit of fun when we do our topic. So let's go ahead and jump in. And I think we want to explore why do we need mobile security? Well, a lot of us have been involved in security on x86 platforms for a very, very long time. But really, the new endpoints that are out there— and this has been growing over the past decade or so—our endpoints are changing. Our endpoints are changing from these dinosaurs I have behind me here to we're doing much more work on our mobile phones, on our mobile tablets, on our devices that are not running the traditional type of operating systems that we've basically grown up defending against.

(04:54)
So we need to look at a different way of doing mobile security and lots of vendors for the past years, some that I've been involved with like Good ATechnology and Citrix on the Zen mobile side have looked at different ways to try to secure the apps and the operating system and things like that. But it's more of a kind of compliance kind of thing. So when we get into why do we actually need this stuff? Well, here's a great article on BleepingComputer that was published just the beginning of September here where the researchers at Symantec actually found over a thousand iOS apps exposing hardcoded AWS credentials. And a quick little search will show there's an article on BeVigil from last year that shows, again, mobile apps exposing AWS keys really share a lot of information and it's bigger than a lot of us think.

(05:52)
And then also in that article, I love this as well because they actually list out a lot of these major attacks that have been actually contributing to the sharing of AWS keys. Now this is just one little tiny example, but the reason that we need mobile security research and testing is because we need to guarantee that our organizations aren't putting vulnerable data or storing data in our applications or exposing it in such a way that we cause a breach for our company or potentially somebody else. So this is really kind of why we need this whole area. We also have the area of vulnerability research into the core operating systems themselves and mobile operating systems because they operate differently and they're more locked down by specific vendors. It's more difficult for us to actually do the level of research that we need to do to find these vulnerabilities hopefully and presumably before the bad guys do.

(07:00)

Now, this is really, really hard because of these kind of limitations that I mentioned just a moment ago imposed by specific vendors or by different types of technology that's out there. But I'm going to bring up three different reasons why mobile security research and testing and things like that are actually hard to do. It's not as easy as it is in the x86 world. And so why is this so difficult to do? The first thing is devices' access to obtaining all these kinds of fun things. The second reason is the operating systems. And we're going to drill down through this agenda here in just a few moments before we go into the playground. And the last one is the time. So let's begin by examining devices. How many of you have been out there doing this type of situation trying to scour and find devices?

(08:07)
So a lot of researchers end up out there dumpster diving. It's difficult to obtain the level of devices that you're going to want to do the research or the testing that you need. Obviously new devices aren't problems, but most often with security research, we need older devices with older operating systems on them. And we'll talk about that in a little bit. And obviously the used phone market that's out there can actually be a relatively interesting environment. You've got potentially security researchers meeting up with people off of Facebook marketplace or Craigslist or some other type of thing trying to find phones and then having to go physically to people and it feels kind of like a shady deal that's actually happening out there. But to do the level of research and testing that we need, we have to be able to find those devices' limited resources. So even if we have physical devices, these physical labs that we're building are limited in two ways.

(09:19)
They're limited in scale.—so the number of devices that we can have online at any one time and the scope of those devices. So maybe you have one or two devices or maybe you've got one old one and one new one and maybe an iPad or something like that. And so you have this kind of limitation of how many of these can we have? Can we have them all? Do we want them all or do we even need some of these? And a lot of times with dynamic testing especially, it's going to kind of require that we have that physical connection to the device to actually feel like we're connecting to it. We actually can USB connect to the device and do things that way beyond even network redirection and things like that, or man-in-the-middle type of scenarios where we're looking for unencrypted information flowing through the wire. And then getting access to these.

(10:18)
So if you don't decide to do it yourself and stand it up and incur all of the issues and things like that around having these things on-prem, then you can—a lot of times there's services out there that will rent you access to their physical lab for a specific point in time—but you've got parallelization issues with that, right? You can't access the same device that somebody else is accessing at that time. So it's a time-slicing kind of thing akin to kind of old mainframe type scenarios. And these online labs, they're usually pretty good for doing some levels of QA testing and things like that, but make it somewhat difficult, especially if you're doing kernel vulnerability research as an example, for you to dig in and get into those devices at that point. And even

sometimes application pen testing because you may not have access to the device at the level that you need.

(11:22)
And let's talk about operating systems for a moment, because operating systems, obviously you're going to need both iOS and Android and there are services out there that offer emulated Android. You can run emulators yourself, you can scale them up on x86 hardware, whatever you're working on to do. All of those things become available. It's more on the iOS side that these become an issue, right? You might be able to rent access from somebody who offers a few hundred iOS devices or something for you to do tests on, but you really need both iOS and Android if you're testing for application specifically or if you're doing training on how to do mobile app pen testing or something like that. Or if you're going to do kernel level research on iOS, obviously you might have your own physical devices, but then you might be, as one of our folks said, carrying a dozen devices through an airport or something like that.

(12:25)
And that just makes it kind of unwieldy. And then you're limited as well on operating systems because of the versions, right? One physical device can only have one OS on it at a time. And if you're going to be moving these devices, let's say you've got an iPhone 7 and you're going to move it from iOS 12 to iOS 15 or something like that—those images are large and it takes a long time to restore those devices and put different operating systems on them, maybe to just conduct a quick test. So that's a major limitation. Probably the biggest limitation that is out there is, especially on the iOS side, access to jailbroken devices or rooted devices. And this is often why we're looking for older phones with, let's say iOS 14 or something on them, that have known public vulnerabilities that we can then use to do a jailbreak on and be able to gain access to the file system at the root level.

(13:30)
Because if you're not doing application testing, especially with access to the root level of the operating system, then you really can't guarantee that your application is not storing data at rest in a vulnerable or insecure manner. So in dynamic testing, when you're running the app and it's saving data, you're signing up for accounts and things like that inside the app, you really can't verify that application or that the engineers wrote those storage procedures in a secure way without the ability to access that operating system at that level. So that's probably the biggest limitation on the operating system side that impacts the level of testing that we can do. And if you're going through the OWASP MASTG on the mobile testing security guide and things like that, there's a significant portion of that testing that revolves around data at rest. And without a jailbroken OS essentially or Rooted OS, those tests are going to be very, very difficult for you to run.

(14:38)
Now time: this is the biggest thing in my opinion, even if I have the devices and I have the operating systems I need, I can do my job. We've been doing this for a long time in these

scenarios, but if you're old like me and you have no hair left, you potentially remember the x86 world when we had to do this physically. We had to have these large rooms with relatively cheap desktop computers with crash carts running around with keyboard and video and mouse. And we used to use a product like Symantec™ GHOST, which would allow us to take a snapshot of a hard drive and restore that snapshot back to a clean state if we're doing malware research, for example. So this endless loop of having to restore physical devices and things like that, this cost us a lot of time back in the day. If you're old like me and remember this, you might even remember these screenshots.

(15:41)
So when we're dealing with physical device labs or even desktop emulators or simulators, we have issues of managing and maintaining them. We have keeping them live and plugged in and chargers in good shape and these things wear out after a while, keeping them updated or potentially not right? Do we let these devices update automatically to the latest and greatest versions? Do we keep them reserved at specific versions? Do we move them back and forth? And if you have a globally distributed research team or testing team, you're now shipping these devices potentially to multiple locations around the world, which is always fun in maintaining hardware inventory. And when we are doing a lot of our security testing and research, we are in this kind of endless loop of trying something, it fails, we have to reset and restore, we try something again, it fails. Or if we're doing malware research, for example, we're executing malware, the device is now compromised, we now have to restore it.

(16:53)
We have to re-flash it back to its clean state. Again, we're kind of in this semantic ghost model where we're re-imaging back to a known good copy of the hard drive essentially. And the physical devices that we have, we have to wait for them to reboot. We have to wait for them to flash. We have an iOS device, we have to maybe jailbreak it, which might require holding specific keys down when the moon alignment is right and the color of your shirt is pink or whatever. These kind of things, they're not a hundred percent accurate in producing results every time, especially with jailbreaking nowadays. So these things that we deal with on a daily basis, the biggest problem with them—and like I said, we've been doing this for a while—but it's really limited what we can do as far as our scope and our scale.

(17:48)
And the biggest limitation is because it distracts you from doing what you need to be doing. The distractions of acquiring devices, managing the OSs, and managing and maintaining these physical device labs or even simulator labs and things like that require a lot of work from security researchers to maintain. And every minute of work that you're spending basically being an IT person is a minute that you're not doing some valid security research or some testing or even some training. So that to me is the biggest thing. And the way we actually resolve these three big problems is we move from physical devices and even emulation and we move into the world of virtualization. So just like what VMware did for the x86 world with malware research and other threat research—the ability to right click on a machine and take a snapshot, do your test,

and then right click restore the snapshot—that kind of power really helped the explosion of the x86 threat research and security research because we could quickly get back to a specific known good version or a known good OS or known good scenario very, very quickly.

(19:18)
And we no longer had rooms of physical devices. We had servers or now Cloud, but we had the ability to access hundreds if not thousands of virtual machines from a centralized location through a web interface, and it saved us a lot of time. So Corellium is not an emulator. We're not emulating iOS and Android running on x86 hardware. What we've actually built is a hypervisor that runs on ARM hardware and runs ARM-based operating systems inside virtual machines. We're not talking ARM-based operating systems like Windows or Linux or something like that. We're talking about literally building a model of a phone and putting it on virtualized hardware so we can run these phones and we can run them and gain access to the root file system without having public vulnerability because we can control the boot process of that. Every one of the iOS devices and iOS versions that we have access to has instant jailbreaks available even up to the latest and greatest iOS 16, which just came out. There are no public known vulnerabilities for it that would give you jailbreak.

(20:38)
So if you're going to be doing pen testing on iOS 16, then this is really the only way that you're going to be able to access that root file system. And there's lots of other things of built-in tooling and built-in capabilities and multi-user and RBAC and all these kinds of things. But basically bringing a single platform that lets you run iOS and Android devices as virtual machines and then letting you take advantage of the time savings of virtual machines. So creating them, deleting them, snapshotting them, restoring them from snapshots, the ability to share a snapshot with other users inside the environment so that you can actually say, Hey, I've got this machine, this phone, it's got this app on it and if you do this, it's going to cause it to crash. Here's a copy of that so that you can test it and debug it and look at it or whatever.

(21:32)
All of these things become available with virtualization, just as I said before. Just as what VMware did for x86, we have done for ARM-based platforms like iOS devices and Android devices, and we run them natively on ARM servers. Now these are deployment options and you can have these servers and these appliances on-prem with you, so you can actually air gap them if you wish. If you're doing some really heads down stuff or maybe you just want to be able to test your applications on your specific network, a very common use case, you can deploy on premises. We also have a Cloud offering that allows you to access these, the access the product through an AWS hosted Cloud product that runs on the AWS Graviton ARM-based platform. So lots of different ways to get to it. Now, before we go play in the software a bit, I want to open up my second poll that I have. And this one is really designed to, and I do apologize, I didn't put another on there, but I'm trying to kind of figure out who is our audience, what knowledge you have of Corellium, but also what area of expertise are you primarily focused in.

And I know you're going to be in several different areas of expertise, but what is the most primary area that you work with?

(23:14)
We'll just give that a few more seconds here. Little over half of you have responded. I appreciate the interaction and thank you for staying interactive with me this morning. Alright, so we're going to go ahead and I'll end that poll and show you the results for a few moments. So cool. We have again, a relatively good mix. Application pen testing is a very popular area and one that we are rapidly expanding in. And also the kernel vulnerability research is great. Mobile malware and threat research—this area probably experiences most of the hold backs of those limitations like I mentioned before because it's just difficult to do. It's not as easy as x86 malware research where I can run a sample of malware and destroy the device, compromise the system and then right click on it and say restore my snapshot. But this is a key area where I think you're going to see a lot of value in our platform.

(24:25)
So as I said before, a few moments in slides and then we're going to go ahead and switch over and it's time to play. So whether you choose Cloud or you choose on-prem, the user interface is essentially the same. You basically have your project and devices that are inside your project and individual users can be members of different projects. I can have other users in the systems that are members with me, but think of this as your project or your device container folder if you will. I call it my demos. And then you actually have the super administrator that can actually assign you resources. So you have CPU resources, you have storage resources, so kind of like vCenter on the VMware side, you can assign and allocate resources to your specific users. Now let's go ahead and I'll just run you through solving that device and OS problem first.

(25:34)
So on the iOS side, we actually have devices from the iPhone 6 all the way up through the 13 pro Max. And I'm actually going to show you a screenshot of the iPhone 14 pro that we already have running, not in my demo environment, but it is in our development environment. It'll be coming out very shortly. So again, you have this vast array of devices that you can choose from, like let's say an iPhone xs. I'm going to go out and I might be able to find one of these for a couple hundred dollars, used, on the used market somewhere. And then we have a device and we have a plethora of devices. We have Android devices, we have iOS devices, but this is probably the most powerful capability is now I get to choose what operating system I want on this device, which build I want.

(26:31)
And I can go all the way back to the very first released version of the operating system that that device had or any of the ones that it's currently compatible with, including 15.7and even 16.0. And I again have the option of do I want this jailbroken or do I want it to be non jailbroken? And this is not going to be seen for a very long time in the public world. We just got some public known vulnerabilities in iOS 15.0 and 15.1 that have led to a tethered jailbreak being available

that's over a year after those versions of the OSs have come down the line. And that's 15.0 and 15.1 clear back here. It's not anything newer in the scale and definitely not the latest 15.6 and15. 6.1 and 15.7 and things like that, not to mention 16. So again, this is how we help solve the device and the OS problem.

(27:41)
And then I can continue through some options here, but we're going to just take a look. So for example, here I have an iPhone 13 pro that is powered down. It's running iOS 16 and it is actually a jailbroken device. And again, because we don't rely on a public vulnerability in the operating system to apply the jailbreak, we can actually jailbreak devices that don't have those public jailbreaks available or known public jailbreaks available. So the next element that I want to talk about when I talk about our platform a bit—so this is a demonstration that we built to show some kernel vulnerability research. And as a kernel level vulnerability researcher, my job is to look for bugs in the operating system that let me gain control of some element of the operating system that I can then use to create an exploit. And I'm just going to show you how some of the tools work, but the devices are essentially real and they interact with me just like a real device would.

(28:55)
I can scroll around on the screen, I can click on things. And when I get to the point, for example, here is the console of the device right here in a web browser, I don't have to SSHN if I don't want to. I can, but I also don't have to. I already have essentially the console access here. And when I get to a point where I have through static analysis or whatever identified a vulnerability in the operating system and I want to actually create an exploit for it and I want to test my exploit to see if it actually does work, this is where it comes in. Where Corellium comes in is in that dynamic testing. So let's test an exploit and see if we get the actual what we're looking for. And in this case, we actually didn't get what we were looking for.

(29:48)
What we're trying to do with this exploit is actually gain control of the program counter and inserts the memory address of our next instruction. And this should say OxFEEDFACEDEADBEEF in here, but sometimes with kernel vulnerabilities especially, it's a try/fail, try/fail scenario. So this is actually not the response that I'm supposed to get, but it still causes the kernel to panic. And when I get into this situation with a physical device, I'm now rebooting the device, I'm jailbreaking it again so that I have access to the command line and to be able to test my exploit again. But in the virtualized world of Corellium, I can come down here to my snapshot and click restore snapshot and this takes seconds rather than minutes to hours of time to get me back to where I can now test my exploit again.

(30:53)
And while that restores, we're going to come back and look at some of our other devices that we have, but that's probably one of the most popular features in the platform, is the ability to just take a device, restore it to any snapshot and be able to get back to using it again. And while that

is restoring, there's a couple of questions that came in. Do we emulate Mac devices? Not yet. Not at this time, but we do because we're ARM virtualization, we actually do emulate other ARM-based platforms. And if you're interested in looking at those, definitely reach out to us. We can talk about some of the other things that we can do.

(31:47)
How long does it take to release new OSs? A lot of times we actually can release OSs very, very close to day zero because we have engineers that are working on the betas and things like that in the background and testing. And so we actually have access to beta versions as well. Okay, let's move into some of these other areas very quickly. There's lots of great questions coming in. Keep them coming in. We'll try to get to as many of them as we can. So let's see here. Alright, so here is an Android device. Again, it looks very similar to the iOS device as far as the tools. And while I'm not going to go through all of the tools today, we do actually offer what we call a quick start training where we actually do go through all of these tools and keep an eye on our website for those. It's offered the first Tuesday of each month. And if you're interested in coming and looking at that, we've got a lot of information about that.

(32:58)
This is a mobile malware demonstration. So if you're in the malware research field, this is going to be kind of right up your alley. And the most popular type of malware on Android devices is adware or this kind of scenario where basically the app doesn't really do anything hugely malicious, but it is doing something in the background usually to make money for the advertisers or their platforms by automatically clicking and signing up for certain things on your behalf. And so this application CamScanner was actually a legitimate program. It was installed on over a hundred million devices on Google Play. This is a relatively older one. If you want to look up Camscanner mobile malware, you can, but I thought as just as a kind of a starting ground, drop in and take a look at CamScanner. This specific version of CamScanner had a malicious advertising library in it that actually clicked on ads in the background.

(34:08)
So really it wasn't doing anything malicious for you, the end user, but basically it was utilizing your device’s resources to click on ads for the advertising framework and making them money on the back end. And I thought what I would do is just kind of show you some of the built-in tools. So we have a built-in tool called Network Monitor. Some of you in the application space might be using Burp Suite or Charles Proxy or something like that to do this, but we also have our own built-in network monitoring tool that basically strips out SSL and TLS and even disabled certificate pinning and things like that. So we can actually see information in clear text. So when I launch the CamScanner program, it's going to start hitting the network or at least it should be starting to hit the network.

(35:13)
Maybe I'm not, but it should start hitting the network and displaying actual information in my app. It is not talking on the network right now. I don't know why I've had this problem with this device

before. So let me just reset its snapshot, we'll come back to CamScanner or I'll actually look at an iOS device where we can actually look at the network traffic. For some reason that one is not online right now. Okay, so let's fast forward a little bit. Here is another device, another Android device. On this device we actually are using a program that is a sample of malware, actually a little bit more malicious. And these are organizations that have essentially begun to migrate away from the Wintel world with ransomware, have actually begun to migrate into mobile malware and mobile ransomware. Now, and this one is interesting, I like this one because a lot of the ransomware up to this point in time have really focused on creating these screen overlays that trick the user into thinking that they have ransomware running but actually don't do anything malicious in the background.

(36:51)
However, this one, Black Rose Lucy, came out earlier this year actually in 2022 with a new version of their program that actually does do a little bit more bad things with files in the background. So I've got the .apk already kind of on the device and I've got this .txt file sitting here. And this .txt file, this is our cannon fodder, right? This is our dummy .txt file, and the reason I use this file is to show you this is what it looks like before we actually run the malware. Now this one is an interesting one. So the first thing I'm going to do is go ahead and install it.

(37:35)
And this doesn't actually execute it or anything yet, but it's actually installed into the device. I think I have had this issue before with Zoom. I think there's a Zoom sharing problem that kills the web RTC on these devices. So here is the app that gets installed. It purports itself as being a video player and so if I click on this video player, it's going to ask me to go ahead and install, but I'm going to click okay. And what's going to happen here is the app's going to go away and you can see here, that video player actually disappeared. However, it's now asking me again if I wish to go ahead and install this streaming video service thing. So I'm going to click okay, it's going to say it doesn't need any sort of capabilities or anything like that, and this is one of these malware that work like a lot of other malware does where it basically tricks the user into installing it because they think you're going to get better video performance or whatever.

(38:56)
But if you just look at this and read some of this text, Android optimization must be enabled to play video on your phone. It sounds kind of shady, it's not really worded well, but you know what? I need this thing in my phone so I'm going to go ahead and enable this service. And then here's the traditional popup that happens on Android when an application is asking for more information or access to the operating system of your device. And you can see here, well there's nothing here actually if I scroll down inside this window, there is some information here, but the authors of this malware have placed enough white space to actually get these things down past where the user would normally experience and see these things. So again, I can be socially engineered to click okay. Now the interesting thing is once I click okay, I'm actually giving the malware access to my device and it's going to go ahead and do the rest of its configuration all

by itself, I'm going to hit okay, I'm going to take my hands off of the keyboard and mouse and you're going to see what's going to happen on the device.

(40:11)
It's going to automatically pop up and give itself all the rest of the permissions that it needs because it's using the accessibility services to pre-program and do that.

(40:24)
Now we do get a traditional kind of malware ransomware popup. It says, we've scanned your device and you've been visiting some bad sites, we want you to go and buy us a gift card. There's information here, all that kind of fun stuff. And if you look at these files, they've actually changed the extension on them, kind of similar to what happens in the iOS world or in the Wintel world. And if we actually open this file now, this file, instead of saying that I'm a dummy file, is actually encrypted. So this is an interesting sample of malware that we actually can have some fun with. And again, once I'm done compromising my device, I can just hit restore on my snapshot and put that device back to its mode. Let's try our kernel hack again and see if we can get that exploit to function.

(41:30)
And again, see here, these exploits are not a hundred percent reliable. So you try and try and try on this one. This one, we actually got the correct entry into that controller where we can actually basically say, okay, your next execution is going to be from a different memory address and this is actually showing that exploit actually getting in there. Now we have a working exploit that we can then take a little further and weaponize, and again, the device is panicking. The screen doesn't work, anything like that. So I can go ahead and restore that snapshot back to its clean state and you can see ransomware is already restored and the device is back up and running in its clean state. Again, that is way faster than having to reboot physical devices, reflash, reset them up, all those kinds of things. So we looked, let's see—let me see if this device has network access. Sometimes we get a device that for some reason or another just doesn't quite get its network access running and it looks like we're in that situation again on this device.

(42:59)
Strange. I'm just going to go ahead and reboot it instead of restoring the snapshot, I'll go ahead and reboot that one. Okay, so pen testing. So we talked a little bit about kernel exploit research, fun stuff there. We did a ransomware demonstration. Now let's look at pen testing. So here I've got an iPhone 7 Plus with 15.7, which is the latest version of OS that it supports and it is in a jailbroken state. On jailbroken devices you already get Cydia pre-installed when the device boots up, so you don't actually have to do anything at all. It's already pre-installed and the devices, because again, because we control that boot process. We're not relying on a vulnerability to exploit to do the jailbreak. They're essentially permanently jailbroken. This device will always boot up in a jailbroken state. Now let's see here. IHopefully we're not having network issues with this device as well. I wonder, I might've set this device up to do some burp suite

stuff. Let me see if I have my wifi going to a proxy. Nope, proxy is off so I'm not redirecting the traffic locally.

(44:34)
I might be having network issues this morning with my devices. Let's just drop into the console and see what we can do here. Okay, we have network. This one's talking out on the internet. That's good. All right, so this one is communicating on the internet. And so what we're going to do is, you're probably familiar with DVIA for iOS. We're going to use this to do a couple of different things, a couple of different tests for vulnerabilities. I'm going to show the network monitor because again, I wasn't able to show it with the adware program on the Android box, but on this one I should be able to use it. And if we use DVIA, at least I'm hoping this is going to work, we can test different vulnerabilities within this intentionally vulnerable app. So we can enter some credit card data here. Let's just send some credit card data and we're going to send it over HTTPS as an example.

(45:43)
And you can see here immediately this post became available and I'm able to look at this in our UI and actually see that even though this was over an HTTPS connection, the actual information being transmitted was being sent in clear tech. So I could say even though this was over HTTPSs, the data that was inside that—because our network monitor tool automatically strips out the SSL and TLS certificates and pinning and things like that—we can actually see that this information was still being transmitted in clear text and that's a bad no-no vulnerability for applications that do that. The second thing I want to do is I want to test for some vulnerable data storage. So we can go look at the data container for this application. Let's see here. VAR/mobile/containers/data/applications. And if I sort by modification time, the newest app that was installed was DVIA.

(46:52)
You see, I created this device yesterday. DVIA was installed shortly after. So this is the unique identifier of the DVIA application Data Vault. And I can come in here and look at documents because the next thing I'm going to do is I'm going to attempt to do some local data storage. So I started the presentation with those AWS credentials that get hard coded or something like that. But let's say we're storing our user data. Our users create a username and a password and we want them to be able to reuse that information.

(47:29)
So we want to put that information, we want to test it and save it in a PLIST or maybe save it in the device key chain or something like that. And once that file gets saved, it should show up here. I may be looking at the wrong data container. I think I am. There it is. I was looking at the wrong data container. So here's the info PLIST file that I actually just created by saving this data. In fact, I'm just going to do it again. I can actually delete the file directly off the device with my browser here, but I'll go ahead and save the file in the PLIST. You'll see it pop up right away.

I can click on the file and then actually launch that and look at it locally again, and I can see that we're actually storing that secure data.

(48:20)
We're storing it unencrypted on the device even though we think we're storing it encrypted. So again, this is another good example of locating a vulnerability in an application when we're doing app pen testing, and because I have access to the root file system with Corellium, I can verify that those applications are installing and storing these files in a secure manner or a non-secure manner. We can go back to development and get that fixed. But I also, again, like I said, we have access to the file system so I can download this file, I can download whole folders or I can even just delete the file and be ready to do this again the next time. So that is a very kind of quick overview. I'm going to go ahead and jump back over to our deck to finish up our session this morning. We have a few minutes left as promised. I don't have it available in my demo environment yet.

(49:22)
It is under construction right now, but here is what the iPhone 14 pro looks like running inside Corellium on a 16.0 release. We posted this to Twitter I believe last week or two weeks ago or whatever when the iPhones came out or were available. So these are very, very quickly available to our Corellium users. Okay, so lots of questions coming in. I don't know if I'll be able to get to all of them or even have the answers to some of these, but let me just go through these. Do we emulate air tags? No, not at this point. We don't have virtualization for air tags, but we do have things that test for Bluetooth availability, wifi ability, things like that.

(50:20)
Yes. So there's a question also around Bluetooth or wifi. Yes, we do. So on Android, we emulate the baseband. We actually emulate the cell signal, and if you're going to use a local proxy to do testing like Burp Suite, you actually go in and set an APN on that device’s cell connection to actually route your traffic through your local Burp Suite on iOS. You do the same. I kind of showed you, I thought maybe I had my proxy set. You do it on the wifi side to do that. Let's see here. If you're doing testing, do we have any way of showing that a specific vulnerability matches an OWASP top 10 number or something like that? No, we don't. It's something we're looking at in the future, but right now we don't really do automated testing. We have, I didn't really show it, but we have Frida in there.

(51:27)
We have those kinds of tools. So we are in the data gathering platform, not the reporting platform. So we're not going to report to you that we found 10 vulnerabilities here and here and here. That is to the actual tester's responsibility, who's actually doing the testing themselves, the kernel exploit? Yes, the kernel panic demo. The exploit tool was an executable that we had written to demonstrate that we could take control over and put in our own desired memory address into the controller and tell it where to go next. So it was a proof of concept of an exploit.

It's not weaponized or anything, but it does demonstrate the ability for you to actually test your exploits on a virtual device and then be able to reset them.

(52:35)
DVIA, people aren't familiar with it. It's basically a go-to application, although it has known issues with newer versions of phones. It's an application. There's several of these that are out there that are intentionally written vulnerable. OWASP has one as well where you can basically go through the testing guide and it's essentially written in a vulnerable manner, so it's overly vulnerable, but it basically allows you to test or to demonstrate the most common vulnerabilities that appear in mobile applications so that you can hone your skills, you can train yourself, you can do those kind of things. So it is an app that is intentionally vulnerable. We're actually working on our own right now. We actually are going to have our own. I was hoping they would be available for today's webinar, but they should be available by next month's webinar. Then we can actually show you our own kind of vulnerable versions and that will allow us to run them on much newer devices as well, like the iPhone 14 or whatever.

(53:44)
There was a question on how many cores does the iPhone 14 run? All of the devices iPhone 8 and newer are all running six core resources, loading basically six cores and iPhone 6s and 7s are on two cores when you use them in the platform. So let's see, we're getting close to the end. Do we emulate fingerprint scanners or other biometrics? No, we don't emulate them, but we do have the ability to essentially bypass them. So giving your application a pass essentially without actually having that access. So you can test your app in a biometric pass or a biometric fail scenario to do that secure enclave on iOS? Yes, we do for on-premises appliances, you do get the ability to debug and look at SEP as well as the iBootinfrastructure. That's only available in the on-prem premium capabilities that we offer there. That's not available for folks that are in the Cloud.

(55:06)
Yes, sometimes I do confuse the emulation with virtualization, but this is truly virtualization. These are ARM-based OSs running on ARM hardware through our hypervisor virtualization layer, and so in different ways we sometimes talk about emulating different devices or capabilities or trap and respond with, let's say there's a physical device that the device, a physical chipset or something like that, that doesn't exist in our model. Those would be kind of emulated inside of the virtualization layers, but all of the good stuff around actually running the iOS and the Android run natively on ARM, not on x86.

(56:03)
So there's a lot of questions coming in. I really appreciate it. Yes, this webinar is recorded. We'll send it out available for everybody afterwards. If you want to get in touch with me, my email is right there on the screen. I would love to hear from you and I really wish I had some more information to give you. We do offer training. We offer, as I said before, that quick start course. It's a two-hour course that gives you a much deeper dive into the platform as well as a deep

dive into, well, a limited deep dive into iOS kernel research as well as mobile application testing, OWASP guide, those kinds of things. And then we do offer much more in-depth training as well. But those quick start trainings, we offer on the first Tuesdays of the month and they are free to attend. Keep looking at Corellium.com for the next one that's coming up in October, and we'll be able to help you guys out with that.

(57:04)
Again, I want to go ahead and end as close to on time as I can. I thank you all very, very much for attending today, taking the time out of your schedule and hanging out with us for a little bit, and hopefully we have shown you something that at least interests you in helping resolve those three core issues that limit what we can do with mobility, security research today, and that's access to devices, access to operating systems at the proper level and the time it takes for you to manage all of that stuff. Hopefully the virtualization platform makes that a little bit better. Thank you all very much. I will see you next month for our next episode. We are going to spend the next three months deep diving into application pen testing, and so come back, look for those in our events area. You'll see emails about them as well, but we're going to be out there again with a webinar in October to talk about application pen testing. Thank you very much. See you next time.

#####