Corellium

Video Webinar

Episode 16: Revolutionize Mobile Pentesting with Corellium MATRIX™ Next-Gen Automotive Software

(00:13)
Brian Robison: Good morning, good afternoon, everybody. Welcome to our webinar today. Thank you all for taking the time out of your busy day to join us. We really appreciate you coming and hanging out. My name is Brian Robison, I'm Chief Evangelist here at Corellium, and I am joined with my co-speaker, Steven Smiley. You're going to hear from him a little bit later and today what we're going to be doing is giving you an update. We covered our new MATRIX capabilities a little bit back in March, but as we get closer to the product being finally released, we wanted to come back and share with you what it looks like today if you haven't had a chance to try it yet, and even give you some tips and tricks and even some updates of some of the latest and greatest changes that happened in our last release, which was actually last week.

(01:07)
So with that, we're just going to go ahead and jump right in and get started. So a little bit of housekeeping. Obviously everybody is muted except for us, the speakers, but we do encourage engagement with us during this webinar. So please feel free to use the Q&A widget tool that is available within the Zoom interface to ask us questions. We love to engage with you and if we can, we'll stop and answer questions throughout the webinar. So please help us do that and we'll be able to get in and help you. It is being recorded and it will be available on demand shortly after the session is over. So you can, and you'll get an email with the link to it as well, so you'll be able to view it and share it with colleagues who potentially missed it. And as with most of our presentations, we're going to have a few slides, but we're going to spend most of our time today in live demonstration.

(02:04)
So with that, we're just going to go ahead and jump right in. And I wanted to just kind of talk a little bit about where is this push for mobile app security, mobile app engineering quality, those kinds of things, coming in. And it's not necessarily all malicious vulnerabilities being discovered and malicious exploits being built. Yes, there is a good chunk of that, but the vast majority of what we're looking for from a pen testing side is misconfigurations of the application or intentionally leaving credentials hardcoded in the app just to get things up and running quickly, but then we forget to go back and remove

those things or potentially we're leaking information or maybe we're storing it in memory too long or we're exposing it in logs on disc and things like that. So really we're looking at engineering code cleanliness and things like that, but we have to do it from a security standpoint and from a pen testing standpoint because if these applications are leaking sensitive data, those could cause a breach in our organizations or even our customers' organizations.

(03:20)
So just a couple of fun little issues that have popped up. Symantec has found over 1800 mobile apps that are leaking AWS credentials. And I think that part of this is the most interesting piece of this to me is that nearly all of the applications containing the hardcoded credentials were actually developed for iOS. So we all feel the rumor is that iOS is a safer platform, it's harder to break out of the walled garden, et cetera than Android, but we also do not necessarily have the highest engineering quality sometimes or cleanliness on the iOS side. So again, it is not something necessarily malicious, but it is just maybe we forgot to remove those credentials or those API keys or whatever they are. And so you also get SDKs and APIs that have hardcoded keys as well that expose certain information. So for example, if you use Twilio, I'm not sure if you do, but a lot of people integrate their SDKs.

(04:30)
So you get instant chat voiceover, IEP communications and things like that through the Twilio SDK. Again, here we're looking at these tools and these capabilities actually leaking information that can be used to exploit our organization. So as pen testers and AppSec professionals, we have to absolutely look at these applications and we have to look at them at a level to make sure that we can guarantee our organizations that there are no data leakage issues with any of the data at rest on the device or maybe stored somewhere else in the device or data that is transiting over the air over the network. We need to make sure that we're not leaking any information in either of those spaces. So we have, as you know, Corellium has been utilized in vulnerability and exploit development research for a very long time. We are also involved in the AppSec community when we're looking at applications and because Corellium has the virtualization platform, but it also has a whole suite of tools of instrumentation capabilities, file browsers, network capture with disabling SSL and TLS built in, Frida, a bunch of other tools that can be used to do and perform pen testing.

(06:00)
But we've also found that what a lot of pen testers really want is to focus more on the really interesting vulnerabilities and less of the checkbox kind of things that have to be checked. Every time an application is tested, and this is both statically or dynamically, you're going to want to checkbox types of things. So if we look at what we're doing with MATRIX, we're taking all of those different tools that we have sitting on top of that virtualization platform and combining them together into essentially a one-click type of

report that you're going to be able to get out of the system that does run through a bunch of those automated and kind of mundane tasks that a pen tester would do. And we look through that for information and yield vulnerabilities or at least findings or artifacts that we're going to discover for you.

(07:04)
So if we look at what we're trying to design MATRIX to do, it's not designed to replace a pen tester. It is designed to augment what pen testers do and essentially making it a little bit easier to do that job, but also fitting into the world of shifting security left as far as possible, getting it closer to when the code is originally written. So a big chunk of what MATRIX is is true automation, being able to use MATRIX in your SDLC, within your CI/CD workflows and to be able to actually automate the testing of mobile applications with the security focused, not necessarily function focused automated testing, but more security testing focused tools that basically can run on every nightly build kind of thing. So being able to run mundane pen testing type of information gathering and assessments on a much more often basis versus maybe once or twice a year as we kind of currently do in our models of penetration testing.

(08:25)
So what I'm going to show you just as a quick diagram here, let's say that a normal pen test over here on the left is generally a two-week effort of both static analysis testing as well as dynamic testing. And in the world today, in most places, this is a 100% manual type of process. It's people running these things, they're clicking on devices or UIs and running through tests and statically decompiling code and pouring through it and looking for hard coded URLs or misconfigurations in your manifest or your PLIST files or whatever you're doing. You're just pouring through it and it's a lot of kind of mundane-ish type of work, but it has to be done so that we can guarantee our organizations that these apps are being developed correctly and aren't leaking any information out. So what we're designing MATRIX to do is to help you with about half of that effort.

(09:31)
We're basically combining things that Corellium can do traditionally in the dynamic space because Corellium is designed to run the application code. We have never been in the static analysis space where we're the code for you and helping you look through that. We have traditionally left that to other tools like Ghidra or Hopper or IDA Pro, things like that. So we're basically focused in that dynamic. Those are all the tools that we have are dynamic. However, with MATRIX, we're now picking up and are able to actually do some of that static analysis as well as the dynamic analysis. So we are actually now exploiting the IPAs and the APKs. We're pouring through them looking for things like hard-coded URLs, misconfigurations in your manifests, in your PLIST files, things like that. As well as at the same time as you run the code or your automation scripts, run the code and click through the UIs and enter information and send information, receive information, we're looking through that dynamically all at the same time.

(10:41)
So essentially you're kind of with one click kind of getting a static and a dynamic test of your application. So the goal is to essentially reduce the amount of manual effort that you would have to do to kind of complete a pen test down into the less of the mundane, but more of the more interesting vulnerability. So if you're now getting a checklist of those mundane things every time you run a pen test within a few minutes, then you have more time to spend looking at those network captures, poking at the backend APIs, making sure that those esoteric vulnerabilities that you might think are there but don't have enough time to get tested because you're running up against that two-week window. That's what this tool is designed to allow you to do—to spend less time in the mundane artifact gathering and assessment and more in the vulnerability discovery, potentially even some exploit development, those kind of areas of the application security tool.

(11:49)
So with that, like I said, there's just a couple of slides just to set level set base here, and then I'm going to switch over into the demo environment and then I will hand it over to Steven to also go through some demonstration stuff here in a few moments. So again, please utilize that Q&A widget. If you have any questions, Steven is on the line with us so he can address them or he can tell me to hit the brakes and we can address a question verbally with you at that time. OK, so because we're focusing on MATRIX, we're not going to do a huge demonstration of Corellium today. We're going to be focused mainly on some already created devices that we're going to go perform some MATRIX testing on. And again, today we're also focused on the interactive method of doing the testing.

(12:47)
We're not showing the API or CI/CD pipeline methods. We will do a future webinar on that, but for right now we're just going to focus on how the tool is used built into the traditional Corellium product. OK, so let's start on Android because we did a release last week and our Android content is quite significant at this time. So 6.4 is brand new, it was released last week and it's mainly focused on a lot more MATRIX improvements that we have. So for example, if we released a ton of new checks both for Android and for iOS, and it's all available in our support center here, you're able to actually view the tests and the checks that are here. So MATRIX is designed to essentially again align to that OWASP checklist of things that you would have to do. And so a lot of the categories and even some of the check names are going to align to those OWASP categories. So you'll be able to see here in our online documentation essentially what Android checks we have and what iOS tests are essentially being run through a MATRIX run. And this list is always updated with every release. So I do invite you to, if you're interested, maybe bookmark that page or whatever and come back to it with each release.

(14:28)
We have also released full regex support, and we did it in kind of a way that's really kind of neat. We did it in a way that allows you to have a keywords file. We're going to go through what this keywords file is, but this is your primary mechanism for reducing false positive. What you're going to do in that keywords file is you're going to store portions of or full pieces of what you would consider to be sensitive data that that application might have access to or your corporation uses or whatever. And you can do it the old way, which is you can just literally put literal strings in. So you would basically find an entire string that matches that keyword or that API key or whatever. But you also now have the ability to combine with that for pattern matching of strings. You can also now implement regex statements, and we did it in a way where you can actually have both regex and flat strings in exactly the same keywords file.

(15:39)
So what you're going to do is you're going to basically put, if you want to use regex, you're going to basically notate it with regex and then you're going to put your regular expression directly inside of those parentheses, and that will allow you to do searches for much more complex types of things, case insensitive versus case sensitive, all those kinds of fun things. And this is regular flat out compliant regex in the middle of these perens. So you can actually have some static strings as well as regular expressions in the same keywords file. So there's also quite a few other things. 6.3 actually brought out kind of the biggest change in a long time. We brought out some new checks, but we also built a whole new…this is when the CI/CD workflow became available as well as the entire new workflow and report environment that I'm going to show you here in a few moments.

(16:48)
So our release notes are always available on a support site, so keep an eye on those. We try to put those out very quickly and give you the ideas of what is happening with these releases. So let's go ahead and we're going to jump into an Android device here. And this is just an Android device I created just before our webinar today. And as you know, we've got all these different kind of tools over here on the left hand side, and what I'm going to do is I'm actually going to install the Corellium Cafe application, which is an intentionally vulnerable application that we wrote so that you can test your pen testing skills and/or run it through MATRIX and make sure that you're getting findings and things like that. So you can use our built-in app side loader that's been here forever as part of Corellium to install your app or even the MATRIX tool has the ability to do it when you create a new test. But I like to just go ahead and use our trusty app side loader that we have and we're going to drop that application onto the device. So now if we look at our device, we now have the Corellium Cafe application installed and ready to go.

(18:13)

As I said before, MATRIX is in beta currently and it is available to anybody who tries our product. It's also available to many of our enterprise customers in the beta. But as we go to our GA release, that is going to disappear unless you have a license product feature set license that does include the MATRIX capability. The great thing about it is it's not difficult to use, there's not a tremendous amount of knowledge required to use this. Steven's going to help us with some of the results that come out afterwards because that's where you're going to probably want to know how to use some of the other tools within the MATRIX environment to access some of those areas. I see there was a question that was asked, but I am unable to get to the question window for some reason on my screen.

(19:19)
I do apologize that for that. Maybe Steven can answer that.

Steven Smiley: I will answer it in chat. No problem.

Brian Robison: Alright. OK. I'll have to figure that out here in a few minutes. It pops up off my screen and I can't see, I don't know why. OK. Anyway, so it's very easy to use. Basically the process is you're going to create a test, which is a self-contained test with a report that comes out at the end. So you can have multiples of them. I've got some other devices that have some historical reports on them, but you could have lots of reports that will show up here in the historical standpoint. And so let's just go ahead and create our new test. So the first thing we have to do is we have to point it to what app we are going to test. And so we're going to choose our Corellium Cafe application.

(20:10)
Now you can go ahead and create the test and run it right now or you can upload that optional keywords file. And this is again, it's the critical piece that helps minimize false positives. So in the case of the keywords file, again, we are looking for sensitive information but not generic sensitive information. We're looking for sensitive information that you provide us and that way it's almost like a guided human guided test. You tell us what are sensitive strings in your organization or in this application specifically, and we're going to tell you if we find them anywhere on disc, anywhere over the air or in memory or other places. So providing those strings to the test allows you to have really strong information. If we turn up a finding that says, we found this string in memory or we found this string in a database, then you know that we found the actual string, not an encrypted or obfuscated string, but the actual string that you provided, whether you provided us with a full string or it's a regular expression formatted.

(21:27)
So it's a really good key way to reduce the number of false positives to essentially focus on what you really want us to test for. So once we have uploaded, chosen our application, it's on our device, we've uploaded our keywords file and these keywords files

can be pretty darn big. One of our engineers actually tested a full script, typewritten script of a movie or something like that. They threw it in there just as a big bulk test. So that was pretty cool to see happen. OK, power up your app and your keywords and then you can go ahead and create the test. Now creating the test essentially creates a container for what's going to happen next. And the way that you run this again right here on the screen, we put a lot of this information in, you're going to click on the start monitoring button and that's going to take Corellium and a bunch of the built-in tools over here on the left and turn them on and start like flight data recorder kind of mode.

(22:36)
Then we're going to interact with our application just as a normal user would. Again, this is the interactive version of it. If you're doing this through a CI/CD workflow, you do this with API calls and SDKs and things like that to make do those interactive type of things. So we're going to do some dynamic stuff. We're going, in our case, to order some coffee, enter some credit card information, et cetera. And then when we're done with that test case, we're going to stop the monitoring and then we can actually run the test. So the test is when you saw all of those checks that we've written, all the different ones, those are going to be then executed against all the evidence that was gathered during the monitoring phase of the MATRIX run. So we're going to hit start monitoring.

(23:30)
And so now basically, as I said before, Corellium is now in flight data recorder mode. It's capturing network, it's capturing logs, it's capturing all the fun stuff that you would want it to capture, and we can begin to interact with our application just as if we were doing kind of a normal functional test. So we're going to add a latte to our cart and we're going to go through the checkout procedure. We're going to enter data, we're going to enter sensitive information. So this would be a credit card number that looks like this would be something that would be considered sensitive to the organization, so I would want to know if I saw that credit card flying over the air somewhere or stored someplace or logged. So that is a good sensitive string there. You're also going to see as we go through this process, things like promo codes or potentially usernames and passwords and things like that that might have hardcoded values in the backend and need to compare what the user entered to, let's say a hardcoded value.

(24:42)
And so if that hardcoded value were to be exposed somewhere, let's say outside of memory where it needs to be temporarily, that might also lead us to a vulnerability. So we're just going to go ahead, submit our order. Now it is important for you to leave the app running. So there's a note here. Do not close the app until the testing is complete. And the reason for this is because we're actually doing some dynamic testing using Frida and some Frida scripts to actually inspect things like the memory and things like that of the application. So if you kill the application, those checks will not be able to run. So it's important that when your dynamic use case is complete, you want to make sure

you leave the application running so that those things are still resonant in memory. So we'll hit stop monitoring, and what it's doing now is it's basically gathering all of the evidence that was gathered during the monitoring portion.

(25:48)
It's putting them into an artifacts directory and it's beginning to do some of the work like exploiting the APKs and things like that, getting it ready to run the tests against the evidence. So now we're just going to hit run test, and this is going to take several minutes to actually run. So while this test is running, I'm going to go ahead and flip over to an iOS device and I hope it finished building. Let's see. All right. OK, so again, I started one right before I started the webinar building, so I was hoping it would be complete. So I'm just going to run through the same process on an iOS device. The great thing is that MATRIX can still be running in the background. So MATRIX is actually running on the Android device right now and it can stay running on that just fine. We're just going to repeat the same kind of procedure over here on iOS.

(26:56)
Installing the Corellium Cafe application is same exact process on iOS. This is what's nice about it, is that you don't necessarily have to be an expert on Android or an expert on iOS to execute MATRIX. We're going to use the exact same keywords file to run on both platforms. So again, it's exactly the same process. So we're going to start our monitoring, and once we see it monitoring the device, we're now going to go ahead and launch our cafe application, and we're just going to do pretty much the exact same procedure over here. We'll order some coffee, we'll add it to the cart, we'll test for a promo code. We'll go through the checkout again, we're entering what could potentially be considered as sensitive information, and we'll place our order. All right, again, we leave this test, we leave this up and running. We'll stop the monitoring here and we'll wait for the monitoring to stop. Again, we're gathering all that evidence and then we're going to go back and we're going to see if the Android one is done yet. The more checks that we add to the system, the longer the tests take to run, but OK, so iOS is now running its test.

(28:38)
Steven Smiley: Sorry, Brian, before you bring up the Android one, we just got a question for someone who wanted to see the keywords file. Can you show that?

Brian Robison: Yes, absolutely. So let me see here. Keywords file for you. So this is what my keywords file looks like. Let me blow this up. So I built my keywords file prior to when we had regular expression support. So I wanted to know if my name was in there. So this is basically without regular expression. This is what we kind of had to do. You had to put both permutations of the name because it is a case sensitive keywords file. There's the credit card information, some hardcoded passwords and keys and things like that, email addresses, things like that that are sensitive information for this application.

And we want to know if we're going to capture this information anywhere. So this is what my current keywords file looks like. Now that we have regex support in here, I'm going to go back and put in some of these same strings, but in regex format as well, so you can see them.

(29:49)
So you can see how you can collapse a credit card down to maybe just the last four digits or whatever your test data is going to be. If it's API keys, you're going to be able to do a lot there as well. So that's what my current keywords file looks like and that's what I was testing for. And I guess, yeah, so here the iOS report popped up while we were doing that and just a couple of things about the report, and Steven's going to go into this a little bit more in detail here in just a few minutes. But basically you've got some summary information on the report. You've got the number of checks that have passed. So pass in our world means we've run the test and we did not return any evidence that would yield a failure. We didn't return anything. Either the data was within the range or within the proper setting that is required.

(30:49)
And you can actually look at these. So you see a past one here, you can actually expand this out here in the UI and you can see kind of a description about what the vulnerability could be, how it potentially could impact you and your organization, and even information on how to remediate that issue if it is found. Now, right now, in this version of the report, you can see here that you can't change the status or the severity, however you can see the framework here for this. So in the final version due here, very, very soon, you're going to be able to modify these for a report as a finding. So you can change the severity, you can override our severity or override our status even if you want. So let's take a look. We did have six findings that resulted in an actual failed result.

(31:43)
So we can sort by just our or filter, by just our failed content, and you're going to be able to see things here, a combination of static testing as well as dynamic testing where we actually found sensitive values, stored insecurely, logged insecurely in memory, all those kind of things. So here's an example of a static. Let's see here we've got, OK, so here's a static ATS being disabled, et cetera. Again, what it is, the impact, the setting for you and the remediation of what it needs to be and where we actually found the evidence. So it was found in the info.plist file of the application. This is the current setting that was discovered from, so that you actually have the evidence here that says, yep, this is actually what was found in the PLIST file. Again, same with other ones that are static analysis.

(32:50)
We're finding things, these hardcoded strings, for example, that are in some of the settings files for the application. And then we can get into some of the more dynamic

ones. So, we can see here that we actually have sensitive data that is being stored in the SQLite database. We can see we actually found the credit card, we found the name because it's a binary file. The string matching and pattern matching doesn't work a hundred percent. And that's why Steven's going to show you here in a minute how to use your built-in file browser to actually go grab that database off the device and look at it in a real way. But again, you can see here where we are discovering some of these hardcoded sensitive information and it tells you exactly where that file actually is, what the application data vault is—essentially the full path to what that file is.

(33:49)
So a cool thing you can do here is come over here and grab this path, copy that path, pop into the interactive console, and you can then basically just go directly to that path right here and you can examine this file if you want in on the device's shell or better off, you'd be better off to use the file browser, download your SQLite database, and Steven's going to show you that here in a minute. But also the report gets archived and saved. So you can have different versions of the app show up here if you use it on the same device testing date, who it's done by, all that kind of stuff. And you can always call up the reports to look at them. Again, you can download them in human readable, HTML as well as JSON files. So if you're going to do something with the data after this, if it's going to fit into some sort of workflow, you can do that, whether you're doing it through the CI/CD automated method, or if you're doing it with the interactive method like we just did, you can still get that data file out that has all of the information in it, so you can use it to create Jira tickets or whatever you want kind of in an automated engine.

(35:11)
And let's go back and review the Android one. So Android finish, you can see there's quite a few more checks and also quite a few more findings in the Android app. So again, here it's the same kind of layout as the iOS. So you're not going to have to create two different types of reports based upon iOS versus Android. Again, you've got your ability to sort and filter so you can sort by your faileds. And again, it's a combination of static analysis. So here are some hardcoded URLs that we found in the Java code, hardcoded URLs, MinSDK versions for encryption, all kinds of fun stuff. And then again, we have the findings where we found sensitive values stored insecurely. So we've got, again, a customer payment database, credit card number, and the CVC code, name, all those kinds of things, even sensitive values in memory.

(36:17)
So this is where we get into some of the Frida checks that are returning if we can find this information in memory. All these kind of fun things are basically how we're returning these results. But as you can see, again, it's the goal here is to not necessarily replace the real art of pen testing, but to help a lot with some of the really kind of esoteric, mundane, kind of just check-the-checkbox kind of things that you have to do as a pen tester. And then you get again, all of the passes and things like that so you can see how

fast that was. I think most of you who've done a manual pen test would probably agree that that was probably a lot faster than decompiling and scouring through files for URLs or something like that. This whole set of reports is done within minutes versus hours or even days worth of that mundane type of work. So with that, I'm going to switch over and give this over to Steven and I will stay on here to answer any questions, but Steven, go ahead and take it away and we'll switch roles for a few minutes.

(37:41)
Steven Smiley: Awesome, thanks Brian. I am going to go ahead and share my screen super quick. Awesome. There we go. OK, so Brian did a fantastic job describing MATRIX and what we're doing and the main goal of all this and to go through this. But what I just want to talk about really quickly, I won't take too much time, is what do you do with the results once you get them? So you have this fantastic report. You can print the report as an HTML or as a JSON. You can put that into your own tools. You can triage some of the findings, that's great. But for anybody who actually wants to action these, wants to go validate them, check for exploits, things like that, what is the next step? And most people, most tools will just leave you at the spot where here are the findings.

(38:26)
You need to go procure a device and set up your environment and go through with the app and find the actual evidence or decompile the binary or all these things. That's going to take you a decent amount of time. So what I just wanted to show is obviously we have the whole platform here. We have the Corellium platform, and for anybody who hasn't seen the platform fully, we have other webinars or we can do other demos on that and talk about some of those features. But in general, we have the whole platform here with the virtual devices to be able to action these vulnerabilities kind of right away. So one thing I'll show, I already ran an assessment so I won't run another assessment. Brian already did a great job with that. I have one that's already ran. So if you look at some of the failed findings here, I'll just go through a couple just to show you how quickly some of this can be actioned.

(39:17)
So the example that Brian showed in his as well, contains hardcoded URLs. So great you have these hardcoded URLs. It's not always a problem, necessarily. We have it as an info severity, like I said, not always a problem. These could be generic URLs that are used for whatever purpose. It might not be any malicious intent to them, but you want to go investigate them. So we actually do provide you the path to these files to be able to go investigate. So just as an example, here is the same device where I actually follow that path using the files tab data, local temp artifacts, and I just followed the decompilation of this app into this WebView activity Java. And you can actually download that file if you want to look at it locally, but you can also just view it right in here. So you have the Java code and you can view it.

(40:06)
And this is for really quick analysis. So if you have something that's popping up and you see some code and you want to go look at the code block associated with it to understand if it's a potential vulnerability really quickly, or if you just want to investigate it a little bit further, this is a great way to do that. So as you look through this, this is where one of our URLs comes out with that blog URL. So if you actually investigate the code, and I won't go through it line by line, but you would notice that the way this is implemented is insecure. They essentially are just taking a URL from the user that can be manipulated using potentially Frida or another tool. It's just not done securely. So what you can do is now take that information that you've gained and you can go over to let's say the Frida tab for example, where you just run…I'm going to run the Cafe application.

(40:52)
I'm going to go in here. We actually have this blog connected here and we actually have a Frida script. So what I'll do is I'll actually hook this process, I'll attach to Cafe, so you'll see how I'm attached here now. And then I'll go over to the scripts and I have this script called WebView JSON. So you can actually view this and edit it right in here if you want to as well. But what this does is actually hook that class and that activity and it actually modifies the URL to something. It could be malicious. You could navigate somebody to another login screen where you're tricking them to enter their credit card or any number of things, but I'm going to go ahead and execute this. It's going to prompt me: do I want to do that? I can say yes. And then what's going to happen is if I actually navigate to this blog URL, if my app doesn't crash, it will load up the malicious URL. So for me it's just a YouTube video, but in general that could be anything. That could be a malicious login or something like that. And in mobile you can't really see whether it's HTTP or HTTPS without looking at the backend traffic. So you don't really know. Somebody could have redirected you to a malicious login that looks the exact same, they're going to get your credit card, something like that. That is entirely possible.

(42:13)
So let me go back to my report really quick here just to see some other findings and some other possibilities. So that's just one kind of option when you're looking at Java files, for example,

(42:30)
There was a question that came in for the remediation. I can share that after. So I can message that, Brian, if you just want to type my email in that as the answer. And then if you could just send me the question, I will respond to that. You got it. Awesome, thank you. So yeah, scrolling through. If you look down, there's obviously a ton of vulnerabilities we're going to go through. I'm just going to look at a few here. So intense vulnerable to redirection. You have something like this where you have a secret activity. Now again, there could be very common activities. Normal activities, you'll still want to

test them. That's as part of any pen test. You're going to be testing those activities to make sure that it doesn't get you to a secret part of the application. It just launches the main part of the application, what have you.

(43:14)
So in this case, you've got something called SecretActivity. Well that's interesting. So you definitely want to check that out. So that's something you can go in and obviously we have the console and we have some abilities to be able to do that. So you can actually use Activity Monitor to start an activity—com.corellium.cafe. And then I'm just going to paste in the SecretActivity path that we just found from the report as well. So what that'll do is that we'll actually start the intent and you'll be able to see in the application, I now have this secret admin portal, which by the way, in the application, there's actually no clickable link to get to this. The only way to get to this is via that activity, but you have a list of every credit card that was entered in that application for the history of that application until it essentially gets deleted. That all gets stored there as well.

(44:04)
Going back to the report, the other thing I will, that Brian briefly touched on as well was things around local storage. So if you've got value stored and securely on the device, and this is part of the reason we use the keyword files to get additional information as well. You can see there's databases where you see the credit card number that I entered when I ran this assessment. So if that's the case, you were going to go in and you can actually use the file tab, same thing, follow those directories, data data, I'm going to type it in. And then databases, it brings you to this path with a customer payment database, you can actually download this locally to view it. I already have it downloaded, so I'll just show it. This is essentially what you see. I know it's very small. DB Browser is not very friendly when it comes to zooming in, but what you would see is the credit card columns, every credit card that was entered in for the application was stored in that database including the CVV number. So you got a whole bunch of information, and again, my keyword file only looks for one credit card, but you get these databases and you start seeing more information. Everyone who's used the application and entered it without being deleted. So there's a lot more information for me. That's all I wanted to show. My whole goal was just to show how you can action this really quickly. It's not just a report. It's not just, here are some vulnerabilities. Go take those and do what you want with it.

(45:28)
But yes, in general, we have the entire platform here. You can take those vulnerabilities, you can follow up on them, you can exploit what you need to, you can download the evidence, you have full functionality to be able to do anything. You need to continue your testing beyond just what that scan provides.

(45:48)

Brian Robison: You can also take a snapshot of the device with the data in its current format and actually share that snapshot with others in your organization as well. So they can bring up the virtual device and look at it with the app and the evidence in the state that it was essentially found vulnerable. So there's a lot of different things that you can do now that you have these vulnerabilities being recreated in a virtual environment versus a physical phone. It's really hard to shut down your physical phone, ship it to an engineer, have them fire it up, look at it, jailbroken, all that kind of stuff, when in fact if you've got a virtual device like this, you can share with them a URL that they can click on and actually get access to that device. So again, it's not as point in time as other solutions that do you basically pay for a test and get a report back.

(46:52)
You might not action that report for months and months and months. And so you really need that much quicker access to the current state of vulnerability. So as engineering solves these vulnerabilities over time, you can prove it. The keywords thing again is a huge thing because again, the keywords are sensitive values that are pertinent to you and your organization and that gives you the ability to look for those essentially in an in-house solution. So this is not going out to a third party. This is your own product that you're going to be doing the testing in. It's your own tool. If you're using our cloud version of our product, Corellium has no access to those devices, so we don't see what's going on on your devices. So even in the cloud environment, your information is still your information and your network is still your network.

(47:57)
We also have on-prem solutions, which allow you to run the Corellium platform, including MATRIX fully on-prem, and those can be utilized to run apps or devices that can communicate with private backend-type of servers. We see that a lot in things like the telecommunication industry or banking where you have remote branches and the apps that run on the devices in those branches or satellite locations connect to a private backend service. They don't connect to the internet and other cloud services, they connect privately. So you want to test your apps and your devices on that same private network. So you can bring the appliances on-prem, run your devices in that private network and not have to worry about that information transiting the internet and other things. So really we wanted to make today's content relatively light. Steven and I are here to answer any questions, but if anybody has any questions, this is the time to address that.

(49:08)
If you have any questions, please feel free. We're here. Again, we can address typewritten questions. Otherwise, this is pretty much the end of our content today. We wanted to make it relatively light and easy to get through. I don't have Steven's information on the slide here, but if you have any further questions, feel free to reach out to me and we can get you more information. We can get you set up with our sales team

if you're interested in trying it, looking at it, playing with it, talking with them about what it takes to get it, all those kind of fun things. We'll be happy to direct you to the proper places. So with that, if there are no further questions, we're going to go ahead and adjourn today's webinar. Oh, good. A question did pop in. Steven, what about false negatives?

(50:11)
Steven Smiley:
Yeah, it's a great question. I'm just reading through. Will it miss some of the checks occasionally trying to…. Absolutely. It's a fantastic question. There's always I think going to be the case where there is going to be a small number of false positives or false negatives. Our goal right now, and part of this beta and what we're doing is we're just trying to put this MATRIX tool through its paces through a number of applications and we're constantly tweaking it and tweaking those checks to make sure that they're as robust as possible. And obviously the addition to things like the keywords file really help us out with potential false positives especially. But on the false negatives, there's always going to be cases where you're going to get it is potentially going to happen. But again, it's just for us, what we're trying to do is again, just run it through its paces, make sure that we can run as many apps through, get as much results through. We're reviewing those results to try to lower those numbers constantly and improve the checks that we do have. So more than happy to work with you guys as you guys run through some applications and get some of that data and figure out where you think some things are hitting and missing.

(51:20)
Brian Robison:
In most cases, I would echo what Steven is saying, but in most cases what I would say as well is that if it's a false negative and we actually are looking at it, I think that potentially would be a bug in the actual test itself. In a lot of cases, maybe we're doing something as we're looking through the evidence, not finding the full pattern that you put in or truncating it somehow or missing it. And so I would look at false negatives as potentially an actual product issue that we would want to work on and get fixed with you. So if you do run a test and you say, I know there's a vulnerability here and you've got a corresponding check that should return that, that's important for us to know. So please, if you do run into that situation, get with us, get with our support teams, and we'll debug that with you and figure out is it a check problem, is it something else?

(52:15)
And we'll see what we can do. But there will be a few of those false negatives that I would say that potentially would be on our part basically as we don't have every type of evidence that are going to return from that. So again, like Steven said, thousands of applications are going to have different ways of storing evidence in different places that we haven't been able to look through yet. So sometimes I think the false negative would

be a bug extending MATRIX with custom checks is on our roadmap. There's a few things and Steven might be able to kind of go through a couple of things as well that we're going to be doing. I did show you kind of what we have in there as kind of a container for modifying the severity and the status of, but that's only in the report. So we are looking at the ability for you to create your own custom checks for us to provide checks and content faster, maybe even someday a community of people who write the checks and content for it. But the ability to make and run tests with custom checks is on our roadmap. We're looking at it and sizing it and trying to figure out where it can fit in. We've got a few things that are going to come ahead of that, but it's definitely a known thing. So thanks for that question.

(53:54)
Steven, do you want to just talk just a few seconds about some of the other things that we have in the shorter term roadmap ideas?

(54:01)
Steven Smiley: Yeah, absolutely. Yeah, so we do have a very long roadmap for this obviously. I think in the shorter term we have some key things coming out. As Brian mentioned, some of the major things, the custom severities, being able to mark false positives for your checks, which is a key thing to be able to, we know every organization is not necessarily the same. Everyone has different,...the severities internally, different policies, maybe they have compensated controls for certain findings. So that's the sort of flexibility that we want to provide to the users just to be able to do for their reports. The other big thing that I think is targeted is our Appium support. So we're still looking at that right now. We didn't show the CI/CD workflow, but that is fully available. So you can actually do that. That will be shown in a different webinar, but you can fully automate this.

(54:54)
The whole piece that we walked through and that Brian walkthrough can be automated. We have action scripts that can be done through our API right now to click in certain spots on the device and actually walk through your application. So we're trying to extend that to include Appium support as well. That'll be a major target for the report. And then beyond that, those are the major things we're focused on right now. On top of obviously adding more checks and additional vulnerabilities and different report data. We also have in our next upcoming release, we're adding an artifact section to the report. So we will be able to see that, where it's artifacts that we have gathered from our testing but might not directly have a security concern yet, but they could be valuable for you as you guys continue testing or you guys are performing a pen test or just a security scan. That's good information to have so that you continue your testing without spending additional time to get those yourself. That is coming as well. But lots of unique stuff coming in the next little while.

(56:03)
Brian Robison : I had a question pop in just reading through it really quick. So a couple of things there. Setting up with MobSF is somewhat difficult with virtual device versus local device. We actually aren't using MobSF at all in any way, shape or form. This is a homegrown capability that we built into our backend. So yeah, MobSF can be somewhat difficult, but we're not using anybody else's products. I mean obviously we have Frida and things like that built in, but we're not using anybody else's static or dynamic analysis test scripts, et cetera. So you can still use MobSF if you wish, with Corellium there's a support guide on how to make it do that and work that way. Yeah, and in fact you should probably just as best practice, you want to make sure that potentially you're using multiple tools to look at the same evidence because each tool kind of works differently.

(57:17)
Just like each decompiler works differently, might turn up different results. So thank you for the comments on that and just wanted to point that out that yeah, we've written this thing from scratch, which is pretty cool. And Steven here has been instrumental in actually creating a lot of the content and the checks as he's a professional pen tester by trade and part of our team here. So I want to thank Steven for working so hard on basically putting his brain into those checks, especially in the remediation and the impact. Those sections are essentially written by Steven and other pen testers to be used by pen testers. So it's good stuff. Alright, we are at the top of the hour. We're out of our allotted time. Do really appreciate all of you hanging out and watching today. Thank you for attending our live webinars. We do these live so you can see us and you can see us do all the mess ups on screen, which is always fun. But we do thank you for taking the time out of your day. Again, it'll be recorded or it is recorded and will be posted for follow-up on our events page, under our webinars. So look there if you want to share it with the rest of your team. Otherwise, have a fantastic day and rest of your week and take care everybody. Thank you very much for attending today.

##