Characterizing and Prioritizing Malicious Code Transcript

Part 1: Identify Characteristics of Destructive Behavior

Julia Allen: Welcome to CERT's Podcast Series: Security for Business Leaders. The CERT Division is part of the Software Engineering Institute. We are a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. You can find out more about us at cert.org. Show notes for today's conversation are available at the podcast website.

My name is Julia Allen. I'm a principal researcher at CERT working on operational resilience. I'm very pleased today to welcome Jose Morales. Jose is a senior member of the technical staff at CERT, working in malicious software research with the Forensics, Operations, and Investigations group.

And I think you'll find today's subject pretty compelling. We are going to be talking about results that Jose and his research team have produced that describe an automated approach that can help malicious code or malware analysts determine which malware is the most severe, the most malicious, and thus should be the highest priority when it comes to analysis and action.

This is particularly critical today given the growth in new malware strains that are released on a daily basis, estimated by some as much as 150,000. And Jose's team has also recently published two blog posts on the SEI website, and we'll include links to these in the show notes for more details.

So enough about all of this, Jose, in terms of tee-up. Welcome to the series. Glad to have you.

Jose Morales: Thank you, Julia. I'm excited to be here.

Julia Allen: Well, this is a great topic and I think will be of high interest.

So to get us all started, if you want to say a little bit about what motivated you to get into this research area. But most specifically, how did you begin to start to tease out an initial approach for both analyzing and prioritizing malware? So if you could get the ball rolling for us, that would be great.

Jose Morales: Sure, no problem. So, my motivation started when I read an article about a virus called Flame, which was a very -- got a lot of news and it got a lot of press coverage. And one of the articles that I read about this virus was that it had been sitting in an anti-malware company's repository for two years before it became known to the public. And I thought to myself, "Well, why wasn't it discovered over those two years? What was going on there? Why didn't it get analyzed sooner, given how bad it was?"

So I started talking to some analysts around here. And it turns out that they receive on a daily basis a huge amount of malware into their repositories and it's more than what they can cover. And they always have a hard time deciding, "Well, I have this huge pile of malware. What do I start with? What do I look at first? Which sample should I analyze? I need some guidance."

So I decided it would make it easier for them if I could come up with a way to analyze all of these incoming samples in an automated fashion, and then based on some criteria prioritize them in a queue, saying, "You should start with this one, you should start with this one, you should start with this one," giving the analyst the chance that, unless they have a request for a specific sample, they can just, without thinking about it, just look at the queue and say, "Okay, I'll start with this one," the first one, and then the second one, and then the third one. And what we try to do is try to get up on top the most malicious ones -- and malicious meaning whatever that group of analysts consider to be most malicious to them or the ones that they are most interested in seeing first.

Julia Allen: So when you say "most malicious," that conjures up all kinds of possibilities. So did you start to see some emerging criteria or characteristics or heuristics that helped you identify what you call most malicious?

Jose Morales: Yes. So, in this research, I applied previous work that I had done, where I analyzed large numbers of known malware. And I was able to enumerate abstract malicious behaviors -- things like self- replication, code injection, process execution, killing anti-malware-related execution processes, modifying the operating system, reaching out to various remote hosts, and doing all of this behind the scenes.

I've noticed that a lot of analysis that's done on malware is usually very objective; it's more like a binary analysis. You run it through a system and it tells you what the binary does but it doesn't really tell you what the malware does. So I see a difference between what a malware does and what a binary does.

A binary does things like create files, open sockets, do a DNS lookup -- things that are very granular, very low-level. The malware, on the other hand, they do things like replicate themselves, remove themselves from the process list, search for anti-malware on your system, log your keystrokes, upload information, set themselves up to run on reboot. You see, those are behaviors at a more abstract level.

Once I understood that, I found that by enumerating those behaviors, given any operating system, it only becomes a matter of figuring out how those behaviors can be implemented on that operating system.

So with the samples that we looked at for this research -- it was all based on Windows -- I already knew what behaviors I wanted to look for and those are the ones that I focused on. I focused on identifying their implementation with the analysis tools that we have here.

Julia Allen: That is really fascinating. I mean, what you're saying about the difference between the binary, which is really more just, "It does this, it does this, and it does that." When you start looking at the malware behavior, the malicious code behavior, you're really talking about impact. What is the nasty thing that is the result of that malware executing, correct?

Jose Morales: Exactly, exactly. When you look at the binary, it's very easy, and there's a lot of tools that will tell you what a binary did when it executed on a system. But that's not what the malware does. The malware uses what the binary did to implement malicious deeds, to carry out nefarious behaviors.

And once you understand the malicious behaviors, then you can run it through an analysis system, set up a suspicion assessment, and based on the behaviors you can say, "Okay, this is doing things. This is implementing known malicious behaviors." And as each one occurs, “I am becoming rather suspicious about it.”

Julia Allen: Got it, got it. So with respect to this approach -- so you've teed this up a little bit, you've talked about some of the characteristics and some of the things that you look for -- were there other aspects of the approach that you'd like to talk about in terms of categorization or characterization before we get into a discussion of your classification and clustering algorithms?

Jose Morales: Well, the behaviors that I used for this work were just very similar to what I've used in the past, and there's a whole list of them. They're all listed in a blog post, so anyone can go there and look; I've got every single one of them.

The main thing is that all of these behaviors only occur when you run the binary on an actual operating system and you watch what it's doing in the system. The one behavior that can be done without running the binary was the digital signature.

A lot of times malware authors are not really interested in having a valid digital signature, although there have been cases where they do have valid digital signatures. But usually checking for the absence of a digital signature, or an unverified signature, indicates a lack of attribution, a lack of provenance. You should be a little wary about that running on your system.

Julia Allen: And you mentioned the topic of provenance. Do you analyze for that? Is that germane? Do you care about where the malware actually came from or is that a secondary concern?

Jose Morales: That's one of the key behaviors. It's at the same level of interest as all the other ones. I want to know that the binary that's going to run in my system has a valid digital signature because that assigns it to someone. You can actually pinpoint a person, an entity that is saying, "This binary is what we say it is." If it does something bad, there's some culpability with the associated company behind the digital signature.

But we've seen more recently that some malware authors create companies for the sole purpose of getting digital signatures under that company. Instead of faking the digital signature, they just legitimize it.

Part 2: Steps to Detect and Prioritize

Julia Allen: Got it, got it. Well, let's get into a little bit more about your method and approach. So in your blog post, you talk about actually training algorithms, in two Categories -- classification algorithms and clustering algorithms -- where you actually train them to test for their ability to both recognize Malware -- I know you talk about not generating false-positives -- so that you can have confidence that when a malware analyst actually uses these algorithms to prioritize their malware, they can be confident that they're identifying the code that they should be examining the most.

So can you describe to us a little bit about how you went about developing those algorithms and training them?

Jose Morales: Oh, sure, sure. So the first step was to create a large set of known malware and known benign. So I had a very large set of several files -- and I think it was around 11,000 something samples that I used -- and that was a very diverse set of malware. You had everything from viruses, worms, password stealers, keyloggers, botnets, backdoors, droppers, downloaders. And we also had in there a subset which is called APT, Advanced Persistent Threats.

We were able to get a subset of the APT malware mentioned in the Mandiant report and that was important to us because those are of key interest to a lot of the people that we work with. The benign set was built by getting -- I think we got five laptops and a couple of desktops, and we copied out -- these are desktops and laptops that are currently in use. We ran several virus scans on them, so they were all clean. And then we copied every .exe that was inside those machines and we put them together to create our set.

So the benign set was a mix of everything from third-party applications that the users installed, third-party applications that IT would install, the .exes that come with Windows. It was very broad. It wasn't just -- some people would think, "Well, you just took Windows executables from the Windows operating system, from the System32 folder, or Microsoft Office products." No, this was way beyond that. And one of the machines was Windows XP, one was Windows 7. I think two of them were XP; the rest were Windows 7.

So it was very diverse, and we had a lot of third-party applications. The users of these machines are designers, developers, a home user, an office user (two office users), and one more developer. So the tools were very broad. And we did that on purpose so we don't have any bias. And we made the two sets to be about equal size.

So we took all that, we ran it through our analysis system called MCARTA, which does dynamic analysis, and it generates a report of what the binary did within the analysis. And from that report, I already determined what behaviors I wanted, so I had to figure out how to identify their implementation in the reports.

Once I did that, we wrote a script. We ran a vast majority -- I think we ran about a smaller set of the malware and of the benign -- through MCARTA, collected their behaviors, and then we used that as a training set for our machine learning algorithm. So what we do in the algorithms is we tell it: "These are the behaviors that we want you to recognize as being malicious, and these are the things that we want you to recognize as being benign.

And then we're going to give you another set, a large set of malware and benign, and you're going to look at them, and you're going to say, based on what we gave you to train on, you'll decide if this is malicious or benign." And that's the general approach that we took.

Julia Allen: So you talk about malicious versus benign. I know you also talk about making sure you didn't generate a flurry of false- positives. So that was another factor, as I'm reading your blog. And then you also talk about this idea of prioritization. Can you say a little bit about those aspects?

Jose Morales: Sure. So what I just talked about was the first part of the work, which was to assure detection accuracy.

Julia Allen: Got it.

Jose Morales: And then once we knew that we were minimizing false- positives and false-negatives, we took the malware that came out of the machine learning classification using Random Forest and AdaBoost.

Those two algorithms, when they tell you, "This sample, based on the training that you gave me, I believe to be malicious," it'll give you a little score about how confident it is that it's malicious. So if you have a sample that's 100 percent -- one, it's from zero to one -- it's a fraction -- if it's a one, that means that according to Random Forest, this sample -- Random

Forest is 100 percent confident that it is malicious, based on the training that you gave us. And the same thing for benign. So given that confidence score, you can sort everything, based on the score.

Julia Allen: Ah, yes, yes.

Jose Morales: And it goes from one all the way down to zero. So we took -- we did two approaches. We did individual samples, just all the individual malware samples. We sorted them, based on their confidence score. And then the ones that were at the top, we considered to be the most malicious because they were the most related to the behaviors that we trained the classifiers on.

But then we started looking at the ones that were at the bottom, anything that was 10 percent or lower. So 10 percent means, "I'm saying this sample is malicious, but I'm only 10 percent confident about it." Or if it's a zero, it means, "I have no confidence in this sample being malicious at all. I have nothing."

But we already knew that all the samples that we used were malicious to begin with. So if you have zero percent, we started wondering -- if you had 10 percent or less, you started wondering, "Well, it could be one of several factors. Maybe it didn't execute correctly. Maybe it identified itself as being in an analysis environment and it acted in a benign manner. Maybe it knows how to undermine everything that we're doing and it runs in a very stealthy way. Or maybe this wasn't the environment that it needed to run all of its malicious events.

What we decided was, after looking at the ones that were at the bottom, we realized that a lot of these were important pieces of malware. And we looked at it a little deeper and we realized that they possessed the abilities to run stealthily, to target only certain systems, and to avoid analysis under certain conditions.

And our conclusion was, “Well, if they have the ability to do that, then they're just as dangerous as the ones at the top who are at the top showing malicious behaviors.” So when it came to prioritizing, we recommend that you should look at the ones at the top of the queue at the ones at the bottom of the queue -- 10 percent or less -- which is something we didn't expect at first, but after looking at the results we realized that it does make sense, that these are more sophisticated malware that, for the various reasons I just talked about, will not give you the true behavior that they're going to carry out when they're in an environment that they feel comfortable in.

Julia Allen: You know, as I'm listening to you speak, I feel like I'm in the middle of a murder mystery, where you have the obvious clues, the obvious discriminators, the characteristics that you identified up front that result in high confidence. But I know when I was reading your blog post, it was, for me initially, counterintuitive to consider the ones with the lowest or zero confidence. And so I'm so glad we're discussing that, because, as you said, those may end up being some of the most dangerous malware, correct?

Jose Morales: Yes. You have to question, well -- see, the key thing was, with the set that we used, we already knew it was malware. Someone else already told us. It had already been vetted. So we knew it was malware. So if you have a sample that you know is malware but you're analyzing and you're classifying it, and it's telling you zero percent, “Well, why is it zero percent if it's malware to begin with? Well, it could be stealthy. It could be smart and be able to avoid analysis. Or simply the analysis environment that we have wasn't the one that it needs to run.”

And for any one of those reasons, it makes it highly suspicious. And so you think, "Well, if it didn't run in our analysis environment, and we know it's a Windows malware and we're in a Windows environment, then what environment does it need?" So is this a specialized piece of malware going after a certain OS with certain features, right, highly specialized? Or it's stealthy and it knows how to detect that it's being analyzed so it takes a different execution path. So what is it hiding? What can it really do?

Part 3: 96-98 Percent Detection Accuracy

Julia Allen: Got it. So if I were a malware analyst at CERT or at another organization, how would I go about actually using the results of the tooling environment and the algorithms and the various approaches that you've just described? How would I get started?

Jose Morales: What you would do is you could take the list of features that we provided, which is the key thing in this research, was the features, the malicious behaviors. You take that list of malicious behaviors and you identify when they occur in your analysis system. So everyone has their own form of an analysis system.

I'm telling you what behaviors you need to look for. Now you just have to find them in your analysis system, in your environment. Once you find them, you can collect the information, you can train the algorithms that we use -- Random Forest, AdaBoost-- and then you use that to classify an unknown set of malware.

Once you're confident with that, actually putting it into use is just programming to look for the occurrence of the various behaviors in an automated fashion. So you take the pipeline that brings in all the samples and you plug into it an analyzer that looks for the behaviors that I've given you. And once you've seen enough to hit a threshold, you throw them at the top of the list, the bottom of the list -- not the list -- the queue.

Julia Allen: Got it, got it. And have you folks actually -- your research team -- have you actually worked with some of the malware analysts at CERT to take this for a trial run or not yet?

Jose Morales: Not yet. What we've done so far was we wanted to find the set of features that would work the best. And that was mostly what this research did -- what are the features that we need to have accurate detection and to have a good prioritization? And our detection accuracy was actually 98 percent, which is really, really good.

Julia Allen: When I saw those numbers, I just couldn't believe it. I mean, I thought that was outstanding.

Jose Morales: And for the Advanced Persistent Threats for that set -- we tested that set by itself -- it was 96 percent. So those are really, really good detection accuracies. That shows the strength of the features that we have and their usability in telling you, for some binary you know nothing about, if it is highly suspicious of being malware or not.

Julia Allen: Got it. Well, before we come to our close, Jose, are there any other key points about the method or the approach or your findings that you would like to highlight that we haven't discussed?

Jose Morales: I think the key thing is to realize that when you're dealing with malware, it's not a matter of describing what a binary does when it executes on a system but being able to identify more abstract malicious behaviors that malware do, and then identify their implementation. Those high-level behaviors, like the ones I give, carry across all operating systems. So you can take them into any OS and just look for how they can be implemented.

Of course, one OS might introduce some new behaviors that we haven't seen before. That's very true in the mobile world. We've been working with Android malware for some time now and they have slightly different behaviors. But a lot of the ones that we've seen in the past carry over and are true; they just implement differently.

Julia Allen: Fantastic.

Jose Morales: One last thing I would talk about is I wrote a paper called "Building Malware Infections Trees." And the key thing that we realized from that work is, at least in Windows and in some other cases, it's not -- in malware in general, it's not just one -- it's not always the case of one binary doing all the malicious behavior. You can have one binary that starts but then it could induce other processes and other files to carry out malicious behaviors on its behalf.

So you need to have the ability to capture all that and keep track of everything so that when some other process aside from the original is detected as doing something highly suspicious, you have the ability to say, "Okay, but he's part of all these other ones. They're all interconnected. All these processes are part of the same tree." So it's not just the one process that's become suspicious; it's all the processes linked to it.

Julia Allen: So you basically have a community of -- if you'll allow me -- cooperating malware all to some malicious objective, right?

Jose Morales: Yes. You start with one and it sort of replicates itself and infuses malicious code into other processes. So if you don't make that link between the original one and a replication of itself that's now a process that's doing the bad stuff that you've identified, you'll only be aware of partial -- partially of the malware infection. The original one might stay behind; some other ones might stay behind, and they could continue to run.

Julia Allen: Very challenging for someone in your position from a research perspective but I also think of the analysts on the firing line trying to combat this every day. So Jose, before we go, do you have some places -- I mean, we've barely scratched the top level of this very challenging topic -- but do you have some places to point our listeners for more information?

Jose Morales: They could go to the SEI blog. I have a two-part series on prioritizing malware analysis. They could look up my paper entitled "Building Malware Infection Trees". And they could also check out the SEI website describing MCARTA, which is our analysis tool.

Julia Allen: Fantastic. Well, I cannot thank you enough for this fascinating conversation and for the body of work that you and your research team are doing, which I think is critical to trying to tackle this escalating problem. So thank you so much for your time today.

Jose Morales: Thank you for having me. It's been great.