NSA's first Trusted Computing Conference and Exposition

Narrator: Hi, I'm Ari Singer. Today, we're going to show that Trusted Computing technologies can protect enterprise applications and data against cyber threats far more effectively than other technologies in use today. To do that, we'll launch a series of attacks against a well-protected standard workstation and a High Assurance Platform - or HAP - workstation, and compare how well their defenses work. Our HAP workstation was built from standards developed by the National Security Agency's High Assurance Platform Program, to show how Trusted Computing technologies can be used to secure the workstation and the networks it's connected to. Two of the key Trusted Computing benefits we'll focus on are: Secure domain separation and attestation. Secure domain separation limits the impact of attacks, and Attestation verifies the integrity of devices to prevent compromised devices from getting access to network resources. We'll start by testing anti-virus defenses, then move on to desktop virtualization, which we've used to secure separate information domains, and lastly we'll show how attestation can dramatically improve the security of an enterprise's network. The vulnerabilities we exploit in this demonstration are publicly known, and software patches are available to protect against them. But, the intrusions you are about to see are real and similar attacks could successfully infiltrate current-day systems. Now, let's have a look at what we're demonstrating.

Narrator: Here are the systems we're using for the demo. Over here we have the two machines that are being targeted for attack: the standard workstation and the HAP workstation. These machines are connected by a switch to a shared network -- like the Internet. An intrusion server, where the attacks come from, is also connected to the network. Each workstation is running secure virtualization software, and each one is running two guest virtual machines. This lets us separate resources for improved security. In both workstations, one of the guest VMs is the "untrusted" or "Low-side" VM-it's used for web surfing, personal e-mail and similar activities. The other VM connects to our "secure network." - In government, this might be called a Secret, or "high-side network." In business, this could be a distinct network for the most sensitive corporate data. A secure switch controls access to the systems on the secure network behind it. On the HAP network, we have a firewall that encrypts network traffic, and a network authentication server, that verifies the integrity of machines looking to connect to the network.

Narrator: NOW, let's get a little more detail on each machine and introduce our team:

Narrator: Bruce is running our standard workstation. He could be working in government or in business - but either way, this is an enterprise-grade machine that is well secured. Bruce: Ok -- I've got a standard Dell machine running Ubuntu Linux. On top of that host OS, I have two guest virtual machines -- an "Untrusted VM" running Windows XP, some current anti-virus software and a PDF reader, and a "Trusted VM" running Fedora Linux.

Narrator: Our focus isn't on the specific software on Bruce's machine, but we have made sure everything we attack on his machine was released in the last year or so. So the exploits we'll show are relevant to real systems that are in use today. Now here's our HAP workstation. Alyssa will be running it for us. She could be in any branch of the military or government, or in any commercial business committed to improved IT security. Alyssa, tell us about your system.

Alyssa: Thanks, Ari. This workstation is actually the same Dell hardware as Bruce's machine, But what makes it a HAP workstation is that it meets the trusted computing hardware, software and configuration requirements set by the High Assurance Platform Program. This particular workstation was built by General Dynamics and is commercially available from the Dell website.

Narrator: Our intrusion server is being operated by Brian. Brian is our resident hacker - a great guy who in real life would never do this kind of stuff... we hope. Brian, can you tell us a little about your system?

Brian: Sure. This is just an ordinary workstation. I actually could have used any hardware or operating system. I have, however, installed a commercial attack tool called CANVAS, as well as several other open source exploit tools.

Narrator: Okay. Lastly, behind me we have a few machines that establish a "HAP-enabled" network. We have an 802.1x secure switch, a firewall and a server we use for network authentication and attestation.

Narrator: NOW let's get to the demo!!

Narrator: In the first part of our demo, we're going to show how typical anti-virus software protects our workstations. Brian is going to send an e-mail from his attack server with known malware in an attachment and both machines will block it. Okay, Bruce, let's start with a look at your standard workstation-- tell us what's going on there.

Bruce: Sure. You can see my Ubuntu desktop with two virtual machines open.

Bruce: The virtual machine on the left is my "trusted" Fedora VM and the one on the right where I have my e-mail open is my "untrusted" Windows XP VM.

Narrator: Ok - and now we'll switch to Brian, our attacker. Brian, what are you up to?

Brian: I've just finished using the CANVAS tool to alter an innocent-looking PDF so it will deliver my malicious code to the machines I target.

Brian: Now I'm writing an e-mail to Bruce and Alyssa with the PDF attachment. If they open it, it'll execute a "heap spray" attack that will install an administrative application on their machines. Brian: Okay. I just clicked send on the email.

Bruce: Okay-

Bruce: So over here on my untrusted XP virtual machine I am checking my e-mail inbox and there's Brian's email.

Bruce: I'm downloading Brian's e-mail and as you can see, my anti-virus detected the malware and prevented it from infecting me.

Narrator: Okay, great - anti-virus blocked the intrusion attempt on the standard workstation. Alyssa, how about on your machine?

Alyssa: Here is my HAP workstation-

Note: this cell intentionally left blank

Alyssa: Ok, you can see there's a green border around my "low-side" or unclassified VM, and a red border around my "high-side" or Secret VM.

Alyssa: My e-mail client and the OS on my "low-side" VM are just the same as Bruce's.

Alyssa: When I try to download the infected file, my anti-virus protects me just like it protected Bruce.

Narrator: Ok, we've shown that anti-virus stopped the intrusion attempt on both machines. It did this by using a "black list" - a large list of code signatures for known malware threats. Incoming emails are scanned, and if code is found that matches anything on the blacklist, it's blocked. Using anti-virus to protect your systems is a good idea. It stops attacks every day and provides a good first line of defense.

Narrator: Unfortunately, a large percentage of machines with up-to-date anti-virus software still get infected. Now we'll show you an example of how this happens.

Narrator: Brian is going to modify the exploit that just failed and try it again. This time, he is going to get through the anti-virus defenses on Bruce's low-side VM and compromise his system.

Brian: I've changed the exploit code just enough so the anti-virus software won't recognize it.

Note: this cell intentionally left blank

Brian: I was able to do this pretty quickly without impacting any of the attack capabilities of my malware. And while I made these changes manually, there are actually mutation engines available that can do it automatically.

Brian: Before I send my new PDF exploit to Bruce and Alyssa, I'll bring up CANVAS to start a "listener" process. This will accept a "dial-back" connection from an infected machine. All right, now that my attack is ready, I'll bring my email client back up and send this new exploit -.done.

Narrator: Ok, now we're going to show Brian's intrusion server and Bruce's workstation so you can see how they interact.

Brian: Right now, I'm just waiting for one of my targets to fall into my trap. This red dot in my attack window represents my intrusion server. If either Bruce or Alyssa gets infected with my malware, their system will automatically dial back to mine and I'll see them show up here as a new dot on the screen.

Bruce: So here's Brian's e-mail. My anti-virus worked last time, so I'm going to assume it's safe to open the PDF. Hmmm. Looks like the attachment didn't launch correctly, but otherwise I don't see anything wrong. Brian: Bruce thinks he's safe - but, you can see a blue dot just showed up on my screen. That means Bruce's machine is infected with my malware. The window that just popped up over here is a command and control interface that I can use to remotely access Bruce's machine.

Brian: Now I can do pretty much anything I want in Bruce's "low-side" or "untrusted VM" and he won't have a clue. So, let's see what's in his My Documents folder - I've got financial documents, itineraries, and other private stuff-, Hmmm - what's this?

Brian: And I can do more than just steal Bruce's files. I can remotely execute programs, change his settings -- whatever I want - on this VM. To prove that I'm in full control of his "low-side" VM, let me just change his desktop background-.And now, for the sake of remaining undetected, I'll change it back to his normal desktop.

Narrator: Brian was able to get past the anti-virus protection and get control of the "low-side" VM on Bruce's machine. That's not good, but it could be worse. Bruce's organization is using virtualization to keep sensitive information in a separate, more secure "high-side" or "trusted" VM. And, for now at least, the low-side VM is as far as Brian is able to get.

Narrator: Now let's try the same attack on the HAP workstation. Just like on the standard machine, the attack will come down and infiltrate the "low side" HAP VM.

Note: this cell intentionally left blank

Alyssa: Here you can see my HAP machine again and I've gotten the same e-mail on my "low-side" VM that Bruce did. When I open it, it behaves the same way it did on Bruce's machine. To me, it just looks like the attachment didn't open properly for some reason. I can't tell if there's anything wrong with my machine.

Brian: Looking at my intrusion server, you can see another blue dot just popped up. That means I now have control of the low side VM on Alyssa's machine too. I can mess with Alyssa's VM like I did with Bruce's.

Narrator: This is a common malware attack. In many organizations, this would have been enough to expose sensitive enterprise data and applications.

Narrator: As we've mentioned earlier, both Bruce and Alyssa's organizations use virtualization to keep more sensitive systems and data safe from attacks like this one. This is a good second line of defense. But it too is vulnerable.

Narrator: What Brian really wants is to get access to the data and applications on the "high-side" VM and on the enterprise network. He'll do this by breaking out of the low-side VM, into the Host OS, and from there get access to the "high-side" VM and the secure network.

Brian: Here's my intrusion server and Bruce's standard workstation again. I'm going to use the command and control window in my CANVAS tool to launch an exploit called Cloudburst against Bruce's machine to try to break out of that low side VM.

Brian: Now the sobering thing is that while the exploit I'm using is sophisticated, I was able to buy it off the shelf pretty inexpensively, and I've had to do very little to make it work. Ok, I'm launching Cloudburst now in Bruce's machine- it's working away- and THERE! In about 10 seconds, I'm in. The purple dot that shows up in my attack window is Bruce's host OS. The great thing from the hacker's point of view is that this is completely invisible to Bruce -- all I needed was for him to open that one PDF in his untrusted VM and my software's done the rest.

Brian: Now that I'm in the host OS, I can do whatever I want there, just like on the guest VM, including seeing that he has a second VM running. - I'd like to see what's in there. So, I'll just install a keylogger on the host OS, and start to capture all the keystrokes made on that system. First, I'll copy over the files, and then launch the keylogger. With this keylogger I'm taking advantage of the fact that all keystrokes go through the host OS before going to any guest VM, so even though I haven't installed anything on the "trusted VM," I can see everything Bruce types there.

Bruce: So, now I want to access my secure network to do some work. I have no idea that there's anything wrong with my system. I'm confident that even if I did pick up something on the low side, my High-side VM should be secure. Overall, I feel pretty safe. Brian: Unfortunately for Bruce, my keylogger doesn't care how secure his virtual machine is because it catches keystrokes before they even get to that VM. It looks like he's launching his VPN client... And now he appears to be entering his private key password. [[Read the password - lets use something fun]]

Brian: - And now he's mapping a network drive. At this point, I just need to grab his VPN key and I'll have everything I need to impersonate him.

Bruce: From my point of view, I've done everything right. I used the VPN to log into my enterprise server, so I think everything is safe and secure. I've had no indication that my passwords were just stolen. Brian: Since I discovered which VPN Bruce is using, I have a pretty good idea of where his VPN files are stored. So I'll just pull the files down from the trusted VM to the host OS, and then from there to my intrusion server. Now, I can log in as Bruce on his "Secure" enterprise network any time I want.

Brian: Bruce has no idea that right now I'm connecting to his network using his VPN credentials. I can map the same network drive he just did. And I've just hit the jackpot! I've used Bruce's "trusted VM" to get access to sensitive enterprise data. Stealing the personal information on the "low-side" VM was nothing compared to this. Here I have access to design specs, product roadmaps, and other critical pieces of intellectual property. You can imagine the implications for Bruce's organization.

Narrator: The lesson to be learned here is that determined attackers can penetrate even the best defenses in common use today. And while this was a pretty sophisticated attack, Brian was able to pull it off with commercially available attack tools.

Narrator: Now that all of Bruce's organization's secrets are easy pickings for Brian, Brian is going to try the same thing with Alyssa's machine.

Brian: Okay, now I'm launching on Alyssa's HAP workstation the same virtualization attack that I used to break into the host OS on Bruce's machine- -- Ok - launched - And now I'll look for the purple dot to show up in my attack window to show that I've penetrated through to Alyssa's host OS.

Brian: Hmmm- I'm looking at the CANVAS output, but Cloudburst doesn't seem to be getting through.

Narrator: What's happening here? This worked great on Bruce's machine, but he doesn't seem to be able to break out of Alyssa's low side VM.

Narrator: Brian - still no luck breaking into the Host OS on the HAP workstation? Brian: Nope - I have no access to anything but the low-side VM. I don't know what kind of protection Alyssa's machine has, but I can't get access to anything useful here.

Narrator: What prevented Brian from breaking into the Host OS? One of the key benefits of trusted computing that we mentioned at the beginning of the demo is secure domain separation. Cloudburst and similar exploits take advantage of security vulnerabilities like heap or stack-based memory buffer overflows, direct memory access weaknesses, and insufficient process isolation to gain unauthorized access to system resources and other processes. [[To block these attacks and ensure secure domain separation, HAP leverages several commercial security technologies. These include hardware security components that protect memory and execution space and that isolate input/output devices. They also include a robust Host OS with strict security policies and commercial virtualization software that has been configured to eliminate vulnerabilities. [[OR]] To block these attacks and ensure secure domain separation, HAP leverages several commercial security technologies. These include hardware security components such as Intel TxT and Vt-d that protect memory and execution space and that isolate input/output devices. They also include Red Hat Enterprise Linux with strict SE/Linux security policies as the host OS and VMWare virtualization software that has been configured to eliminate vulnerabilities.]] These are just a few of the trusted computing technologies and techniques used in the HAP workstation.

Narrator: Stopping attacks like the ones we have shown is critical, but since it's unrealistic to assume that we can be 100% successful preventing all attacks, it's even more important to know with confidence whether machines are safe or have been compromised. As we've seen, Bruce had no idea his machine was compromised. Narrator: But Alyssa's HAP workstation uses Measured Boot and Remote Attestation technologies to verify her machine's integrity.

Narrator: Each time a HAP workstation boots, it goes through a Measured Boot process, in which critical software measurements are stored on the Trusted Platform Module, an embedded hardware security chip. The TPM, already a core hardware component on most enterprise PCs, is vital because it provides a secure "root of trust" for the workstation. It stores system measurements and cryptographic keys where they are safe from software-based attacks. When a

HAP workstation tries to connect to the network, a Network Authentication Server verifies the measurements and machine identity. This process is known as Remote Attestation. If the new measurement of the software matches the stored "safe" measurement; and if the machine identity is verified, network connection is allowed. But if the software has changed, or if the machine identity is wrong, network connection can be denied. Let's have a look at remote attestation in action.

Alyssa: When my machine booted, it tried to attest to the network. Since remote attestation happens automatically and invisibly, there isn't much to look at other than to show that our workstation passed or failed attestation when it tried to connect to the network. So, over here we are showing a view of the attestation server log that shows that when we booted up our HAP workstation, it successfully authenticated.

Alyssa: We've highlighted some of the key output messages that come through. If all the cryptographic checks and verifications succeed, the system is allowed on the network.

Alyssa: You can see on the Network Authentication Server that when the HAP workstation successfully attested, the light turned green to show the new connection.

Narrator: Attacks from unauthorized or compromised machines on the network are a tremendous threat. Although measured boot and remote attestation may not be visually stunning, they are very powerful cyber defense tools, because they can be used to limit network access ONLY to KNOWN systems running software that has been verified.

Narrator: For this stage, we've changed things up a little bit. We are going to modify our HAP workstation to simulate a compromise.

Narrator: However, we actually know of no way to remotely compromise the host OS on a HAP workstation using an intrusion server, so we've allowed Brian to sneak into Alyssa's office where he'll attack her system locally. Brian will modify the host OS on the HAP workstation to simulate a successful attack on the host OS.

Narrator: Then, when the compromised HAP workstation tries to connect to the network, the TPM on the HAP machine will report different software measurement values than those expected, and the Network Authentication Server can deny access. So, the compromised host OS can't be used as a jumping point into the rest of the enterprise, and the attacker is isolated to the host OS. The trusted computing technologies in the HAP workstation successfully contain the attack.

Narrator: Brian, can you show us what you need to do to modify the HAP workstation?

Brian: Sure. I had to physically come over here and reboot the HAP workstation so that I could get access to the host OS. I also needed the password so that I could log on as root.

Brian: I entered into single user mode by modifying the HAP's boot menu, and now I'm at the command prompt. At this point, since I'm logged on as root, I can make the modification. I'm now running a script that is making this "malicious" change to the host OS. Now that it's done, I'll reboot the system, and then it will try to re-attest to the network. Narrator: Alyssa, can you show us what happens on the server side after Brian did this?

Alyssa: Sure. During the process, the HAP workstation attested with its new - modified -- measurement values. This caused the authentication server to reject the network access request. It didn't matter what the change was, any change would have caused attestation to fail. Here is the server log that shows my modified machine failed attestation, and Brian has been prevented from getting on the network.

Alyssa: We've highlighted the part where it shows that the attestation failed.

Alyssa: And you can see, there's an orange light that shows my machine is not authorized to access the network because attestation failed.

Alyssa: Depending on the policies set by the administrator, a compromised HAP workstation could be connected to a separate network for remediation or it could simply be blocked from network access altogether. As you can see on my HAP workstation, attestation failure can be flagged so that appropriate action can be taken - it could be shut down, or sent to a Remediation Server for troubleshooting, or it could be allowed to continue to function in some limited capacity. This is all configurable by system administrators.

Narrator: The remote attestation process we've illustrated here is far more reliable than any other technique for protecting network resources. If Bruce's organization had been using HAP, Brian would never have been able to impersonate him using his intrusion server, and neither Bruce nor Brian could have used Bruce's compromised machine to get network access. That's because we're not just looking for known threats in a race against malware writers. With attestation, any change to the measured environment is detected, and workstations with unauthorized changes are prevented from getting on the network. The clear benefit: No untrusted workstations on the network.

Narrator: So what does all this mean? It means it's time to change the way we build enterprise IT infrastructure. Modern-day workstations and networks remain highly vulnerable to attack. We've shown how both anti-virus and virtualization defenses can be circumvented with relative ease. Trusted computing technologies like those in HAP reduce the risk of compromise, improve our ability to detect compromises quickly, and minimize the damage cyber attacks can cause to both endpoint devices and the networks they connect to. The trusted computing technologies in the HAP workstation that you've seen here are real, they're tested, they're Department of Defense-certified, and they're all commercially available. Today's cybersecurity challenges require more than incremental enhancements to traditional solutions. What's needed is nothing short of a revolutionary new foundation for secure computing-- the foundation that you've seen demonstrated today. If YOU would like to know how to build trusted computing technologies into your solutions or your enterprise, contact the HAP program office to learn more. Thanks for watching.