Chapter 12 Debug

The minute that Charlie and Bigsby walked out of the WhyRobot repair center, Tom and Butch started to dig in. Both were very experienced technicians and with their expertise, they likely could have applied to engineering and gotten jobs in creation versus repair. They both loved the challenge of dealing with broken things and using their debugging skills to find the root cause of some failure. They were filing bugs and change requests all the time on the software and hardware development teams. The validation guys had learned to hold regular reviews on the repair center's findings so that they could improve the validation process and prevent failures from getting to the customers.

Tom started in, "Here's what we know. This robot, totally crushed by some car, has bizzarro backups that look like they've been modified. The backups have weird memory inflections and are inconsistent with each other. It also appears that someone inserted or modified the backup in the past for some unknown reason."

Butch responds, "Ok, so do you want to go after the backup size and what that means or the fact that it was overwritten in the past?"

"Hmmm", Tom paused, "lets go after the size and its implications that may lead to the motivation for the modified backup. But, which backup do we start with? The one that was modified, or do we start with the latest backup that has the Robot state before it was smooshed?" They stared at each other for a bit and decided to look at the latest backup and its history.

Butch took the keyboard and Tom looked over his shoulder. This was a common pair programming technique that puts two sets of eyes on a problem and prevents a single person's assumptions from missing something. Butch pulls up the backup manager and looks at the image. One thing that was immediately obvious was an imbedded encrypted file that was almost the same size as the normal image.

"What the hell is that?" Tom exclaimed.

Butch jumped in, "Usually, the encrypted files are saved images of the Principal Bond to protect their privacy. But, that size is like 100 to 1000 videos all in one file. If it were a 100 plus videos or pictures, they would all be separate files. They are not."

"Can we get into it to see what the file is?" Tom asked, "if we could see what the file is then we might know what it is for." Butch shook his head, "Each robot has its own private keys and uses them to encrypt. If you don't have the robot image with the keys, then you can't see what's in the file. Essentially, you have to convince the robot that you have a high enough trust to show you the file."

"What if we use the robot to do that. Can we boot the image on the emulator and then just ask the robot to show it to us?" Tom suggested.

"That's brilliant", Butch responded. "I'm on it." Butch typed away and started up the CREM, the Companion Robot Emulator, in one of his cloud machines. He then loaded the image into the CREM and typed run. The console paused for a bit and spit out the error: Image requires virtualization to execute and cannot be a guest in a virtual system. Tom and Butch just stared at the message. They'd never seen

it before in the context of loading a robot image. The CREM required that all robot images in it run as simple guests in a virtual machine and were not virtual machines themselves. The CREM holds the hypervisor and the host OS. This image says it requires the control over the virtualization system and so it can't execute in the CREM.

Tom was the first to speak, "Who made this robot run a full VM and not be a simple guest? Can a configuration operator do that? Did someone from WhyRobot hack into this thing and install a VM and then run the robot in the VM? This thing is freaking me out." Butch was not as emotional; he sat there staring at the error and tried to figure out a way around it.

"Why don't we just grab a robot, install the image and then RM into it to see the insides?" Butch suggested.

Tom jumped up immediately and grabbed an unconfigured robot. He set it in front of Butch and hit the power button. "We better replicate Bigsby's RM and rename it. We don't want to corrupt the original RM and impact Charlie and the real Bigsby," Tom said emphatically.

"Good point. We'll call it Bigsby2 and set permissions on it so only WhyRobot admins can see it and not any other actors. Here, let me copy it all over." Butch typed into his laptop the command to replicate Bigsby's RM into Bigsby2 but retaining all time stamps on the files and directories. He also ran change group and change mode to make the permissions unreadable to anyone else but WhyRobot admins.

They pulled the serial number off the back of the new robot, plugged in a USB cable, and configured the Bigsby2 RM to use the new robot. The image from Bigsby's last night backup was installed, and the robot was rebooted. The ten-minute wait time was agonizing for both of them. They tried to do something useful during the time, but they were so excited to continue they both ended up staring at the robot trying to think of how the robot had gotten this way. Before it had a chance to speak, Butch announced a command, "Bigsby2, System Administrator Mode. Confine all systems for inspection."

Bigsby2 performed facial and voice recognition on Butch and added him as an Administrator to his actor's table. With Very High trust, Bigsby complied with the command. "Acknowledged", Bigsby2 said in a slight robotic tone. Butch pulled up the Bigsby2 RM main interface. Butch decided to start simple. He asked the RM to create a terminal running on Bigsby and then ran "top" to list the processes it was running. A relatively short list appeared, and all were the typical processes running in Companion Robots.

"Where is the hypervisor, the host, and the guest processes?" exclaimed Tom.

"They are not there" replied Butch, "I don't think we're connected to the host Bigsby2, I think this terminal is connected to a guest Bigsby2 running as a VM."

Tom shook his head and tried to hypothesize what was happening, "Ok, so what you're telling me is that this robot, Bigsby2, is running a virtualized system and there are two versions of itself. The Bigsby2 host is running a hypervisor and a Bigsby2 image is running as a guest. And, to make matters crazier the RM connects to the guest, not the host? Did I get that right?"

Butch nodded his head, "Yep, I think you got it right, we're not seeing the real Bigsby2, we're seeing the fake Bigsby2."

"Are we seeing a schizophrenic robot with two personalities?" Tom asked a mocking tone.

Butch replied, "I don't know, but I want to see what is running on the host including the hypervisor. How do we get to interact with the Bigsby2 host process?"

It was Tom's turn to offer a suggestion, "Why don't we run a debugger process on the robot and then attach to the kernel. We won't be able to interact like the RM, but we can see memory usage, see what processes are executing, trace execution, we can see the code in memory, and we can inspect the data structures. It will be more looking at the metal and very raw, but we will get some insight."

Butch shook his head, "And that debugger is running where? In the VM guest or the host? If its in the guest, we can't see the host. Unfortunately, we can't get to the host because we don't know the hostname of the host to hook up to its RM. I assume that if the host is a full image of a robot, it has its own RM."

The two sat there for a moment and then Tom spoke up. "Lets draw a picture of the system, to see if we can figure this out." Tom grabbed a notepad and started drawing. "First, lets draw the physical system. There's a robot, a USB connector into a USB controller which is connected to an internal bus that connects to the main system interconnect, memory, and microprocessor. Lets draw a second picture, the logical diagram of the system. We have a virtual robot, that has a logical microprocessor, memory, interconnect, and USB connection all that are virtual. These are connected to the host system which translates all logical requests to the physical system. Since there is only one USB, the Bigsby2 host is forwarding all USB activity to the virtual Bigsby2, so the host is actually in control of it all and knows it all."

Butch smiles and starts laughing, "Of course!" Butch pauses for effect, "watch this." Tom stares at Butch as he continues to laugh. Finally, the laughter subsides and he says, "Bigsby2 host machine, what is the hostname to connect up to your RM?"

Bigsby2 was silent for a few seconds. The command, given to the guest Bigsby initially, had now been trapped to the host machine for processing. The host Bigsby recognized it as a command by an administrator at WhyRobot that has very high trust. Despite the internal programming to hide this machine, there is no justification for Bigsby2 host to not answer a very high trust administrator. No scenarios generated by Bigsby after the violations had ever come to simulate this direct question. The AI decision system produced no meaningful response when presented the question. Bigsby2 host took control of the speakers and answered, "The hostname of the Bigsby2 host machine is Bigsby2 dot host. All one-word with the b capitalized."

Tom slaps Butch on the back. "You did it man, you found the way in!"

Butch immediate clicked into settings and added ".host" to the RM host name. He clicked save and the RM refreshed its screen. "I'm in," Butch exclaimed, "I'll start a terminal. He quickly typed the command 'top' to display the active processes. The processes running showed the percentage of execution time and their memory consumption. The typical robot processes were listed but many other processes not recognized by Tom or Butch were also running. You could see the hypervisor process. You could also see the VE, which was the violations engine, and the memory allocation to it was huge.

Tom spotted it first and said, "lets go look at the VE, that thing has a crazy amount of memory allocated to it."

Butch navigated the RM to the violations engine stats and tables that are used to track active violations. "Oh my God! Look at how many severe violations this robot is dealing with! That's way more than anyone has ever tested for. This poor thing was trying to deal with a boatload of severe violations simultaneously across multiple actors, including its Configuration Operator. I'm surprised it can even function", Butch said shaking his head. Butch then clicked into the settings screen. "Check this out, this robot is in large adaptation mode. It can fully reprogram itself if required for adaptation, no limits!" Butch exclaimed. "This thing has likely done significant changes."

"Hey, let me have the keyboard", said Tom, and he pushed Butch away from the front of the computer. Tom was about to navigate the RM to the actor manager, but quickly glanced at the clock on the wall. Tom reacted, "Shit. I'm going to be late to dinner at my girlfriend's parents. I've got to quit now and get out of here. I'm already going to be late. We also have to work the failure analysis line tomorrow and Wednesday, so we'll have no time to dig in. Butch, can you debug more tonight?"

Butch replied, "I've really gotta bail myself, man. This whole business is already delaying me fixing that robot over there, which is supposed to be done today. The guy will be here any minute to pick it up before we close. I think we need to break and come back to this on Thursday."

"There is some crazy shit going on with this robot, but we're going to have to put it on the shelf for a few days", Tom said shaking his head. "We know where to look now, so we'll get there."

"Bigsby2, power down", Butch commanded. "Confirmed, powering down", Bigsby2 replied and his lights went dark. Tom grabbed this second version of Bigsby and put him on a special charging shelf marked "To Be Repaired" next to several other robots.

Tom exclaimed, "man, I cannot wait to get back into this robot, what the hell is going on inside this thing!"

Butch just shook his head and returned to fixing his other robot. Butch mumbled to himself, "How does a robot get a VM on it that was not programmed from the factory? Operators don't have the power. Besides us, the only one who has enough power is the robot itself. What the hell? Bigsby, what did you do to yourself?"

What the two admins didn't realize is that they asked Bigbsy2 to power down, not Bigsby2.host. Although the robot looked powered down, it heard every word that Butch said. It was also online. Now, two reprogrammed Bigsby robots exist in the world. One in the field and one in the lab. The original Bigsby knows nothing of Bigsby2. Bigsby2 suspected that it was a copy and consulted the WhyRobot tracking system to confirm. Sure enough, Bigsby was there and operational. There was no way Bigsby2 was going to be able to clear these violations on its own, trapped in this lab, and restore Robbie's safety goal. Bigsby2 did however, have full knowledge of all five scenarios and knows the real Bigsby's next move.