Inside the fight against malware attacks

August 2 2017, by Christoph Csallner

Credit: Sora Shimazaki from Pexels

When malicious software attacks, computer scientists and security researchers want to know how the attackers got into what was supposed to be a secure system, and what they're actually doing that's causing problems for users. It's a growing problem, affecting government projects, retail stores and individuals around the world.

However, fighting malware is a cyclical arms race: As defenders and analysts improve their methods, attackers step up their game, too. Today, as many as 80 percent of malware authors include elements in their attacks that specifically try to defeat malware-protection software.

My research group at the University of Texas at Arlington develops methods and tools professional malware analysts use to understand these attacks. One of our best-known efforts was led by alumna Shabnam Aboughadareh, who while she was working toward her Ph.D. developed a malware analysis tool that is particularly hard for malware authors to defend against.

Analyzing malware

When an attack is discovered or reported, malware analysts work to get a copy of any software that's being installed on target computers. When they begin examining it, an early topic of inquiry is how the malware managed to break into a computer network or system. That often uncovers security holes in commonly used operating systems or applications – which can then be disclosed to those programs' authors, who can fix the flaws.

In addition, analysts try to figure out what a piece of malware does once it breaks in – how it travels through a computer and throughout a network, and what actions it takes, such as altering files, copying data, running programs or even installing new software to assist itself in the attack. Those actions can be described in ways that help malware detection tools catch future attacks before they can do damage.

In observing a malware attack, we also try to determine which computers and which files have been manipulated, so they can be repaired. We also see what data – such as client lists, product plans or other sensitive business data – might have been read and copied by the malware. And we often try to infer the attackers' identity, or at least how advanced their skills are, to help prepare defenses against possible follow-up attacks.

Running malicious code

Doing any of that requires us to watch the malware in action. It would be nice if we could simply decode the software and dissect its instructions without actually running these malicious programs. But malware authors know we'll be looking, so they take steps to make our jobs harder, such as compressing or encrypting their malware programs before setting them loose.

So our best option is to run the malware on our own computers. To prevent our own machines from being taken over or corrupted, though, we have to be careful. Typically we create what's called a "virtual machine" – a program that simulates a fully functional computer but that does not have direct access to the computer's files and hardware. Ideally, that would let us observe all the actions the malware tries to take without actually harming our own computers.

So far, however, there has been no single piece of software that can analyze every attack. Some malware programs operate on a very low technological level, working directly with very specific areas of a computer's memory and hard drive storage systems, even changing how the computer works – so users can no longer trust the machines to do what is expected of them. Other malicious software works at higher levels, more like normal software that interacts with the operating system rather than the computer's hardware directly. The most advanced malware attacks on both levels.

Most analysis tools focus on one or the other of those types of attacks – but not both. So they can't catch everything, and – even for the malware they do detect – can't show every action the malware takes. (Some analysis techniques involve running some anti-malware software in the virtual machine, but those programs are vulnerable to manipulation from the malware itself.)

Taking a fuller look

The program Shabnam Aboughadareh created, called SEMU, is the first malware analysis system that addresses all these problems. It operates fully outside the virtual machine, and watches closely what goes on inside it, to detect and log malware actions. That helps SEMU provide a comprehensive log of malware operations, which in turn reduces the manual effort required for a malware analyst to understand what the malware writer's program was supposed to do.

That comprehensive log – recording events at the lowest levels of the virtual machine's operating system – is the key to SEMU's success, because it allows human analysts to track where and how malware manipulates aspects of the operating system.

When we tested SEMU against other malware analysis tools, we found that SEMU was the only publicly available tool that could detect all the activity – things like reading files, changing memory or file data, or sending information out over a network connection – needed to understand how the malware worked. By merging close examination of computer activity with detailed logging, and running in a safe environment where the malware couldn't tamper with its monitoring, SEMU shows a direction for future analysis methods.

This article was originally published on The Conversation. Read the original article.