EMCS2600: The Future of Cybersecurity: Technology and Policy

Assignment: 1st Short Response Paper for Modules 1-3 (Draft)

The greatest foundational threat in cyberspace is the delta between the advances in malware deployment techniques and the slower advancements in-memory analysis. Most modern compilers and interpreters have no methods for detecting illicit behavior and malware defenses are most concerned with detecting malware by analyzing its signature. As malware becomes more advanced Operating Systems and the programming languages must become more sophisticated in their approach to defending against malware.

Modern computing uses less and less physical memory as the “thin client” approach to computing becomes the norm in personal and business computing. Systems with very small hard drives tend to shift the computing to the cloud, applications running in the browser and/or application that manipulate data in RAM. Some applications never save anything to the hard drive at all. Malware has been moving quickly to adapt to this new world while operating systems, hardware development, application development and antivirus has lagged behind.

Today malware has been defined as malicious software that usually falls into one of the following categories:

Backdoors

Botnets

Downloaders

Information Stealing Malware

Launchers

Rootkits

Scareware

Spam-sending malware

Worm or Virus

How can fight this new threat? As Jessica DeCianno argues in her article for Crowdstrike we need to start looking for different elements IOA vs IOC. “Unlike Indicators of Compromise (IOCs) used by legacy endpoint detection solutions, indicators of attack (IOA) focus on detecting the intent of what an attacker is trying to accomplish, regardless of the malware or exploit used in an attack.” This shift in perspective in not an easy one. While it has been embrace and celebrated by the hacker community as evidence by the number of talks about in memory exploits at DefCon in 2019. Every main stage talk at DefCon involved some sort of in memory attack, whereas at BlackHat in memory attacks were sidelined to smaller sessions. This is a clear indication that corporate America is not taking in-memory exploits as seriously as they should. There are only handful of security companies that like Cylance and CrowdStrike that can boast of full throated approaches to analysis of IOA. While I don't take for granted analysis of IOA is extremely hard, mainly because analysis of memory is extremely hard. If a hacker can do it, application developers can also defend against it, but it's going to take some serious collaboration from the programming community together with OS publishers and hardware manufacturers to make a safe system.

What does a collaborative effort from the programming community, operating system publishers and hardware manufacturers look like? The programming community needs to start taking responsibility for adding security to compilers and interpreters. Operating system publishers need to work with the hacking community to understand how attackers pivot into memory systems and how they can work with programming languages and hardware manufacturers to surface meaningful alerts to users when activity in-memory matches the elements of IOA. Third party software like Cylance and CrowdStrike can only do so much. Deep collaboration will create an opportunity for the programming language and the operation system to validate the intent of the users, maybe with hardware as a more immutable test for the authenticity of the communication between the two.

Malware doesn't have an inherit goal of having files on the hard drive of a machine. The inherent goal of attackers is to steal information and access. Shifting the perspective of defense to match the offensive attacks only makes sense. Furthermore failure to move faster could threaten the integrity of the entire ecosystem.