cybereason

Lab Analysis

Fileless Malware: An Evolving Threat on the Horizon

By: Ian Muller, Senior Security Writer
Yonatan Striem-Amit, CTO
Amit Serper, Senior Security Researcher

cybereason

Malware is usually a piece of software, designed to perform a specific malicious task. It's written by an individual and installed by one. We're accustomed to seeing these programs attached to benign files or installed via backdoors and hidden from anti-virus' through a fairly standard series of techniques.

However, lately we're seeing a new breed of malicious operations utilizing inherent "features" built into Windows that allow a clever hacker to turn the operating system against itself and compromise entire networks without needing a single piece of software to be installed. In order to better protect your environment from these types of threats, it is critical to understand how attackers are manipulating these tools, which are native parts of the Windows operating system itself, and using them to gain access and create a launching pad for their malicious operation.

There are two different tools that hackers are beginning to utilize as a foundation for fileless malware: Windows Management Instrumentation and Powershell. What's particularly new about the use of WMI and Powershell today is that attackers are developing their entire operations without installing a single file on the targeted machine. Because of this, it's impossible for traditional antivirus tools to detect the attack, and amplifies the difficulty of identifying the threat for other security solutions. When the entire malicious operations resides within the WMI/Powershell framework, we have to change the way we think about security in order to protect ourselves.

In the following sections we explore the underlying technology behind these services and how it can be subverted for malicious uses, as well as highlight the challenges these technologies pose for classical security approaches.

Windows Management Instrumentation

WMI allows an admin to perform a variety of actions, such as gather metrics, install software and updates, or self-query the OS itself. WMI has access to every resource on the machine, and is divided into classes for different tasks, such as executing files, deleting and copying files, changing reg values, etc. This tool is built directly into every modern version of Windows, and is the backbone of the “agentless agent.” These inherent features make WMI a blessing because it allows admins to perform tasks very quickly, but when used for malicious operations it very quickly because an even bigger curse.

The same way an admin uses WMI to query metrics and execute code, a hacker can use it to run malicious code across an entire network of machines. This can be done silently, instantly, and is undetectable by standard anti-virus as an undesired action. It also allows for persistence through auto-running programs stealthily on startup or based on specific events. Furthermore, WMI *cannot be uninstalled*, it can only be disabled, but doing so impairs/limits admin and software capabilities, such as disabling Windows updates.

Powershell

Similar to BASH or Python, Powershell is an internal scripting language inherent to the Windows OS that allows access to Windows API and is used for automation of tasks. It can be considered the big brother of WMI, allowing admin access to tasks, etc. Powershell has full access to .NET, so entire programs can be written and executed directly through it. This allows attackers to easily hide themselves by dropping Base64 scrambled or encrypted code directly into Powershell, automatically decoding it and running it from memory. This isn't a new tactic, but furthers the ability of Powershell-based attacks to avoid most real-time detection and popular security programs.

What makes Powershell even scarier as a tool for hackers is that it can be run remotely through WinRM, legitimately, allowing attackers to instantly punch a hole through the Windows Firewall on that compromised endpoint. This allows the attacker to run Powershell scripts remotely, or simply drop into an interactive Powershell session, providing complete admin control over the machine, without a single piece of malware or exploit being run. And, if WinRM is turned off… it can be turned on remotely through WMI using a single line of code.

Ignoring traditional lines of attack

What makes WMI and Powershell such amazing threats is that they completely blur the line between compromising a single machine and compromising the entire enterprise. The moment a hacker has a username and password on one machine (which can be easily obtained in PtH and PtT scenarios), the path to complete compromise is laid wide open.

In a traditional attack strategy two of the most important steps are installation and communication with the command & control server. Typically these two stages are thought to be outside of the scope of WMI. However, recent explorations of WMI's capabilities have shown that it's ability to change system configurations, as well as store and load additional binaries, allows an attacker to easily communicate both inside and outside of the victim's network.

One reason that Powershell is such a devious tool for an attacker to utilize is that it's a trusted scripting language in Windows. However, scripting languages, like Bash, Python, and Powershell, are significantly more powerful and provide unprecedented access to a machine's "inner core" because of their unrestricted access to Windows APIs. Because Powershell is an inherent part of Windows, it is completely trusted by the machine, ignored by antivirus software and other defense mechanisms. All the attacker needs is access to the machine, which can be achieved using WMI as shown above.

What Can These Services Do?

We’re already seeing a plethora of tools that utilize these services for malicious intent. At Black Hat USA 2015 and Defcon 23, Matt Graeber, a reverse engineer at FLARE Labs, FireEye’s R&D branch, discussed some of the new tools and the trend of utilizing WMI and Powershell. There are several known fileless malware, such as Poweliks, Powersploit, Empire and WMIGhost, which all utilize these service in various fashions to provide unprecedented access to a machine.

Poweliks

A Powershell-based tool that uses Reg Keys to autorun scripts that load other keys and dynamic libraries to gain control over the system, without installing a single piece of software, loading encrypted malware directly to memory on the machine.

Powersploit

A suite of tools developed by Graeber that act as “addon modules” to Powershell itself, amplifying its potential for malicious operations. These tools allow for a variety of malicious tasks to be performed such as DLL, shellcode, and Reflective PE injection; keylogging; timed screenshots; data exfiltration; taking a full memory dump of a process; port scanning; and running Mimikatz. All of these tools are loaded directly to memory, so antivirus solutions can’t detect them.

Empire

Similar to Powersploit, Empire is a post-exploitation agent that provides the attacker with a platform to stage their operation from. Whether it’s running scripts or commands, or leveraging other Powershell functionality, Empire includes 90+ modules for advanced Powershell-based attacks.

WMIGhost

This tool has WMI listen to events on the machine, with specific triggers that occur when a designated event happens. This tool can be used to host an entire malicious operation across a network simply by combining triggered WMI actions, hiding the attack within the normal Windows processes occurring on the endpoint. For example, when users save files to a specified directory, WMI would then automatically upload them to the C&C server, exfiltrating the data as part of the seemingly benign task.

The Bottom Line

WMI and Powershell are critical tools for Windows administration, but are being subverted by hackers as powerful infiltration solutions. Traditional approaches to security are rendered useless in the face of these attacks because WMI and Powershell are both highly reputable, have trusted signatures, load directly through system memory (which cannot be scanned using heuristics) and have unrestricted access to the operating system because they are integral parts of Windows.

It’s incredibly difficult to tell regular WMI and Powershell operations apart from malicious ones using traditional security methods. Graeber noted in his talks that he is unaware of any security solution that can detect WMI persistence, for example, in real time. However, there is a way. We can begin identifying and tracking such attacks through large-scale behavioral analysis. By utilizing the same approach that the hackers are to subvert these services, we can gain visibility across WMI and Powershell and start detecting these new attack techniques. We have to ask critical behavioral questions, such as “which machines invoked remote WMI/Powershell access on the other endpoints,” “what is the scripts actual activity,” and “how is the script interacting with the remote system?” By asking these questions, we can better identify potentially malicious use of these services and start working toward a better solution.

cybereason

222 Berkeley St., 13th Floor
Boston, MA 02116 USA
www.cybereason.com

About Cybereason

Cybereason was founded in 2012 by a team of ex-military cybersecurity experts to revolutionize detection and response to cyber attacks. The Cybereason Malop Hunting Engine identifies signature and non-signature based attacks using big data, behavioral analytics, and machine learning. The Incident Response console provides security teams with an at-your-fingertip view of the complete attack story, including the attack's timeline, root cause, adversarial activity and tools, inbound and outbound communication used by the hackers, as well as affected endpoints and users. This eliminates the need for manual investigation and radically reduces response time for security teams. The platform is available as an on premise solution or a cloud-based service. Cybereason is privately held and headquartered in Boston, MA with offices in Tel Aviv, Israel.