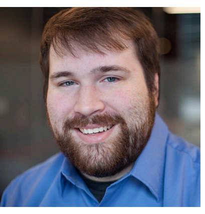
red canary

RESOURCES • BLOG
THREAT DETECTION



Lateral Movement Using WinRM and WMI

TONY LAMBERT

Originally published November 20, 2017. Last modified October 1, 2024.

Many organizations invest millions of dollars to bolster their systems and prevent attackers from gaining entry. Much less attention is given to the concept of lateral movement within an organization. Yet we've seen time and time again that once an adversary breaks through the crunchy outer layer of the network, the gooey center quickly becomes trivial to move about.

Stopping lateral movement is just as important as preventing a breach. Attackers frequently move laterally with tools included in Windows, and this tactic has also been observed within commodity malware samples. This article will outline a threat detection in which Windows Remote Management (WinRM) spawned a process via Windows Management Instrumentation (WMI).

First, let's take a look at normal execution of WinRM and why it's important to secure it.

How WinRM Usually Works

Normal execution of WinRM involves the configuration and inspection of the Windows computers across the network. Within minutes, an incident can grow from a single compromised system to hundreds using built-in tools in Windows.

We usually see simple commands such as “winrm get config” or “winrm quickconfig.” In this case, we noticed something very different…

What are we seeing and why isn’t it normal?

Functionality of the WinRM command is provided through a Visual Basic Script, so it’s natural to see WinRM configuration performed from the CScript utility. The command of concern used with WinRM here is the “invoke” command. This command allows WinRM to work with management resources defined by the Windows operating system, primarily through WMI. After looking into the structure of a WinRM command, we discovered that whatever comes after “invoke” is a method defined per management resource or WMI class.

In this case, the Win32_Process WMI class has a “Create” method. This allows the user of WinRM to execute a process via WMI.

Now we can interpret the rest of the command.

The screenshot below enumerates three things to note.

The “-r” switch (1) signifies the WinRM Invoke statement is being executed on a remote host specified at the “HTTPS” address. This is significant because if we look at the host this command was executed from, we won’t find evidence of what happened after this command. To get visibility from here, we’d need to jump over to the remote host and observe what happened. The “-a” (2) and “-c” (3) switches signify the attacker authenticated to the remote host using a certificate identified by the specified thumbprint.

At this point it should be becoming apparent this isn’t normal. There are much better ways to run simple applications when authorized to do so. For example, legitimate system administrators can use PowerShell Remoting or PsExec commands to run applications on remote computers. And when on a local computer, users can simply double-click on applications or launch them through the Command Shell or PowerShell. Quite simply, processes started in this fashion are an anomaly. But, hey… at least the attacker used encryption when connecting to the host!

How to Detect This Threat

A detection engineer originally found this event due to other bad behavior on the endpoint and realized we needed a way to better detect it. To get more information about its detection, we jumped into a test lab to look at different permutations of this attack. We realized the attack would be slightly different if the attacker had left out the remote connectivity options, so we began there.

In most cases when a command launches another command, we expect to see the second one spawn as a child process of the first. So in theory we would expect WinRM to spawn a process as a child to “CScript.exe.” Once we got into the lab, we found a different reality:

CScript had no child processes! Not only that, it didn’t modify any files or leave any other telltale signs that a process had been spawned. So we needed to dive deeper to find how Notepad executed…

Notepad spawned as a child process of “wmiprvse.exe,” a binary whose function allows WMI to interface with the rest of the Windows operating system. Our WinRM command simply submitted an operation to WMI, and WMI used its own interfaces to execute that operation and spawn a process. Our initial detection specified a remote host, so our next round of testing needed a remote host. This time around, we raised the stakes by attempting to execute “vssadmin.exe” instead of Notepad since we’ve observed lateral movement used with vssadmin during ransomware attacks. When we examined the execution of vssadmin, we discovered that it spawned from “wmiprvse.exe” just like our first test (which did not involve a remote host). This time around there was another catch: we didn’t see a network connection from “wmiprvse.exe”. In fact, we had to search around to find the connection, and it was shown as established by one of the “svchost” processes. This is due to the nature of WinRM, since it executes as a Windows Service and makes all the relevant network connections on behalf of the processes that use it.

To expand your detection functions for this attack, you’ll need to monitor processes spawning from “wmiprvse.exe” and suspicious network connections to “svchost.” For those interested in monitoring processes spawning from WMI, be warned! It gets noisy, and you’ll need to establish a baseline of what looks normal in your environment. Once you can outline legitimate activity from your admins, you can focus on spotting evil.

How to Mitigate This Threat

WinRM can be secured in a few different ways. First, system admins should establish a management network architecture that lends itself to increased security. This involves establishing a jumpbox that is only used for remote administration functions. This strategy helps secure a network by forcing this privileged activity through a single controlled and hardened system, therefore not exposing sensitive credentials to attack. It also helps secure WinRM directly because we can limit the number of hosts trusted by the WinRM subsystem. In an ideal environment, client computers in the organization should not trust one another, and they should only trust the jumpbox systems. To configure what trusted hosts are allowed to contact WinRM, we can execute the following command:

```
winrm set winrm/config/client '@{TrustedHosts="JumpBox1,JumpBox2"}'
```

This configuration can also be enforced using Group Policy objects in an Active Directory environment. This can be accomplished via a policy with the “Allow remote server management through WinRM” setting, and specifying a list of hosts to trust.

For authentication to WinRM for management, keep the defaults when possible as they don’t allow the less secure methods of authentication (Kerberos is default). Finally, WinRM default configurations establish both an HTTP and HTTPS listener. If you can, endeavor to disable

Key Takeaways

Preventing malicious lateral movement is just as important as preventing the initial breach. Limiting this movement helps security teams “stop the bleeding” during an incident and prevent it from becoming a full scale breach. Detection of this threat is difficult as WMI processes are noisy, but a solid understanding of your network makes it much easier. Taking action against this threat is a great way to defend your organization and stop a breach in its tracks!

RELATED ARTICLES

THREAT DETECTION

Artificial authentication: Understanding and observing Azure OpenAI abuse

THREAT DETECTION

Apple picking: Bobbing for Atomic Stealer & other macOS malware

THREAT DETECTION

Keep track of AWS user activity with SourceIdentity attribute

THREAT DETECTION

Subscribe to our blog

You'll receive a weekly email with our new blog posts.

First Name

Last Name

Email Address

SUBSCRIBE ›

See Red Canary in action

Schedule your demo now

Get a Demo

→

Search

PRODUCTS

- Managed Detection and Response (MDR)
- Readiness Exercises
- Linux EDR
- Atomic Red Team™
- Mac Monitor
- What's New?
- Plans

SOLUTIONS

- Deliver Enterprise Security Across Your IT Environment
- Get a 24×7 SOC Instantly
- Protect Your Corporate Endpoints and Network
- Protect Your Users' Email, Identities, and SaaS Apps
- Protect Your Cloud
- Protect Critical Production Linux and Kubernetes
- Stop Business Email Compromise
- Replace Your MSSP or MDR
- Run More Effective Tabletops
- Train Continuously for Real-World Scenarios
- Operationalize Your Microsoft Security Stack
- Minimize Downtime with After-Hours Support

RESOURCES

- View all Resources
- Blog
- Integrations
- Guides & Overviews
- Cybersecurity 101
- Case Studies
- Videos
- Webinars
- Events
- Customer Help Center
- Newsletter

PARTNERS

- Overview
- Incident Response
- Insurance & Risk
- Managed Service Providers
- Solution Providers
- Technology Partners
- Apply to Become a Partner

COMPANY

- About Us
- The Red Canary Difference
- News & Press
- Careers – We're Hiring!
- Contact Us
- Trust Center and Security