Investigating Lateral Movement — WMI and Scheduled Tasks

blog.gigamon.com/2022/02/03/investigating-lateral-movement-wmi-and-scheduled-tasks

February 3, 2022

Threat Research / February 3, 2022

Michael Lazic

Lateral movement is a necessary step early in the attack lifecycle of any successful breach. Once an adversary gains access to a victim environment, their natural progression will be to move laterally to other hosts of interest, getting ever closer to the ultimate objectives. Attackers use a variety of methods for lateral movement, but this post will focus on those using Windows Management Instrumentation (WMI) and Scheduled Tasks.

Both WMI and Scheduled Tasks play the same role in a lateral movement operation: They provide a means of execution on a remote system over the network. We'll look at WMI and Scheduled Tasks individually to discuss why, how, and what type of adversaries use them for lateral movement. Finally, we'll explore questions to consider when detecting and investigating these intrusions over the network.

As with several other tools, WMI (wmic.exe) and Scheduled Tasks (schtasks.exe) executables are part of a collection of binaries often referred to as LOLBINS, or Living off the Land Binaries and Scripts. These are programs or scripts that an adversary can, with relatively high reliability, expect to be present on a Windows-based system.

WMI and Scheduled Tasks are also commonly used by administrators and third-party vendor solutions (such as configuration management or system monitoring) that generate significant traffic in the environment and allow adversary activity to blend in with "normal" operations. To demonstrate the volume of such activity, Gigamon ATR observes hundreds of thousands (for Scheduled Tasks) or millions (for WMI) of monitored events per day across environments.

Finally, both tools offer flexibility of use, allowing for credentials to be passed, to be run under specific user contexts, or executed with user-defined commands. Given their accessibility, ability to blend in with benign activity, and flexibility, WMI and Scheduled Tasks make for effective go-to tools for lateral movement by adversaries.

We can see from existing industry research that the use of WMI and Scheduled Tasks is common across a range of threat actors, from ransomware to advanced persistent threats (APTs). In the case of ransomware, recent leaks from the Conti ransomware affiliates identify both WMI and Scheduled Tasks used for lateral movement. Industry reporting on APT41 and

Dark Halo indicate use of WMI and Scheduled Tasks, respectively, for lateral movement. When looking at how threat actors are using these tools in lateral movement, we see that it's primarily an execution method, with the item being executed varying.

WMI is often used by calling the wmic.exe binary directly; however, several tools exist for offensive use of WMI, such as Invoke-WMIexec, WMIexec.py, SharpWMI, and the built-in WMI functions for frameworks like Cobalt Strike. ATR research finds that regardless of the tool, the functionality over the network tends to remain the same. Irrespective of the tool used to execute WMI, adversaries seek to execute a payload on the remote host, typically an executable or script created by the attacker that initiates a connection to command and control (C2) infrastructure. In some instances, instead of a payload, a command such as a PowerShell "one-liner" is used to initiate the connection to attacker C2. Below is an example of WMI use with wmic.exe:

```
wmic /node:<remote IP/Hostname> process call create "<command>"
```

Figure 1: Remote command execution wmic usage example.

The **/node** switch allows targeting of a remote host, and the process class with the **call** verb for the **create** method allows for the execution of the passed command. Wmic will run under the current user context unless the **/USER** and **/PASSWORD** switches are used.

Scheduled Tasks use is similar but more elaborate due to the nature of Scheduled Tasks. Specifically, a task needs to be created or registered, then run and "cleaned up" (deleted) if the attacker wants to remove obvious traces of activity. While a limited selection of third-party tools exists for Scheduled Tasks, calling the schtasks.exe application directly is the predominant method for adversaries. Example usage of creating or registering, running, then deleting a task is provided below:

```
schtasks /s <hostname/IP> /RU "SYSTEM" /create /tn "<task name>" /tr "
<command/payload to execute>" /sc ONCE /sd 01/01/1970 /st 00:00
schtasks /s <hostname/IP> /run /TN "<task name>"
schtasks /S <hostname/IP> /TN "<task name>" /DELETE /F
schtasks /S <hostname/IP> /TN "<task name>" /DELETE /F
```

Figure 2: Sanitized schtasks example found in Conti leaks.

Schtasks is often run with the **/RU** switch to define what context to run the task under. This can be used to run under the system context, which is a high privilege on the host. However, the option to run as a particular user is also available with the **/U** and **/P** switches for user and password respectively.

When used for lateral movement, Scheduled Tasks creates a common pattern of execution (create, run, delete). Note that the delete action doesn't necessarily need to be done remotely; it can be done locally on the newly compromised host once it establishes the connection to the C2.

Detection and Investigation

While many might immediately look to host-based detection methods to identify the use of WMI and Scheduled Tasks, the network should not be overlooked. Lateral movement naturally occurs over the network from one host to another and therefore exhibits network activity from both hosts for defenders and investigators to consider.

WMI

When detecting the use of WMI, as described above, over the network we need to understand what traffic to focus on. WMI primarily leverages the DCE RPC protocol and, from ATR's research, uses IWbemServices endpoint for the ExecMethod operation when used for remote command execution as described. Unfortunately, the presence of the ExecMethod operation on its own is not an indication of malicious activity, as this operation is conducted by several benign sources, such as administrator activity and configuration management solutions. The context surrounding the ExecMethod is therefore important in determining if lateral movement could have occurred.

As previously mentioned, we see adversaries executing a payload of some kind when laterally moving with WMI. This payload would need to be present on the remote target host prior to the execution for it to be successful. Therefore, the adversary must upload a payload via SMB from an already compromised host to the target host before the execution of WMI. Once the adversary executes the payload with WMI, we expect to see a connection to a likely external C2 destination. This host may be the same destination that the host initiating the lateral movement was already communicating with, a common C2 server. When forming this context around the ExecMethod, a sequential procedure becomes clear:

1. Host A uses SMB to upload a payload to Host B, which is the target of the lateral movement
2. Host A then executes the payload remotely via WMI, issuing an RPC ExecMethod operation
3. Host B initiates a connection to an external destination that Host A was also communicating with (a shared destination between the two hosts)

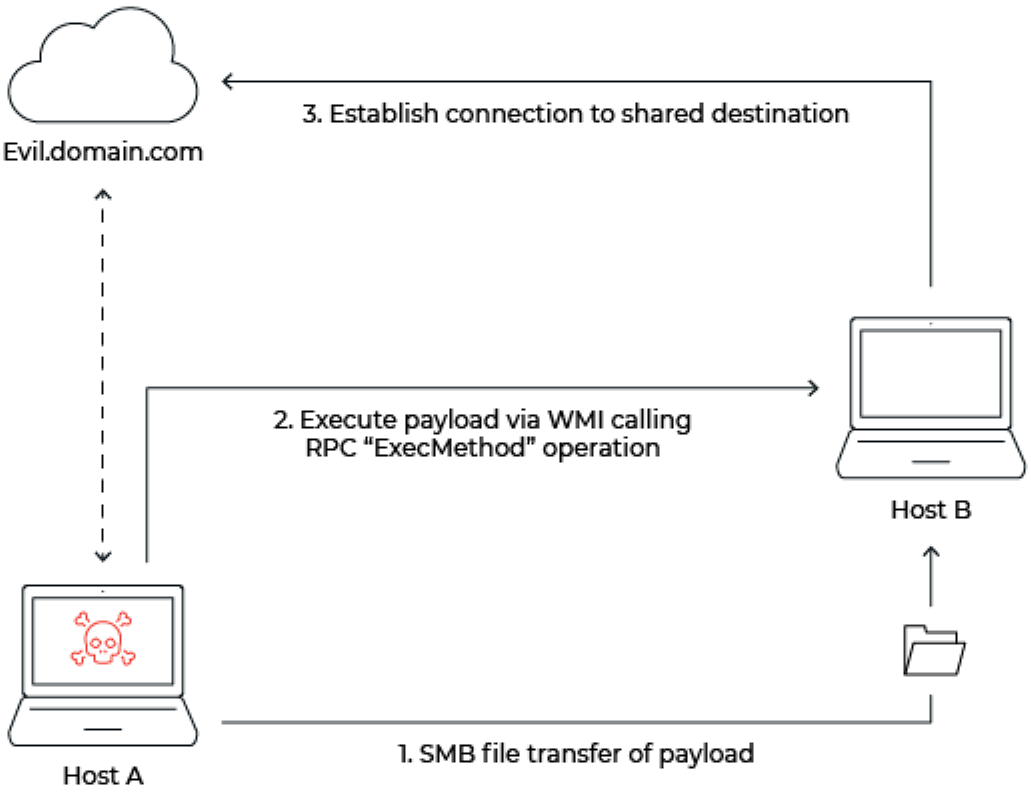
Evil.domain.com
3. Establish connection to shared destination
2. Execute payload via WMI calling RPC "ExecMethod" operation
Host B
Host A
1. SMB file transfer of payload

Figure 3: WMI lateral movement procedural steps.

A common variation would be to forgo the use of a payload and execute a command, using WMI, to establish a connection to the C2 server. In this case, we still expect to see a shared destination following the WMI execution, as seen in Figure 3:

1. Host A executes a command via WMI on Host B, issuing an RPC ExecMethod operation
2. Host B initiates a connection to an external destination that Host A was also communicating with (a shared destination between the two hosts)

Scheduled Tasks

Scheduled Tasks-based lateral movement can be easier to identify compared to WMI because it's less commonly used in a remote communication setting and requires a greater number of operations to execute the movement. As indicated previously, when using Scheduled Tasks for lateral movement, a series of operations are conducted in a pattern for typical scenarios. Like WMI, Scheduled Tasks uses DCE RPC for remote execution with the ITaskSchedulerService endpoint called along with the following operations:

- SchRpcRegisterTask (Create/Register/Modify task)
- SchRpcRun (Run task)

In addition to these essential operations, the following might also be present but are not required to successfully complete a movement:

- SchRpcGetTaskInfo (Get a particular task's details)
- SchRpcRetrieveTask (Retrieve a list of tasks)
- SchRpcDelete (Delete a task)

Attackers might query tasks before or after a movement attempt and, as part of cleaning up their tracks, delete the task they created. However, the task deletion could also occur locally on the newly compromised host and doesn't need to be done remotely.

The execution of a particular pattern of Scheduled Task operations is not a sure sign of lateral movement. To better determine if lateral movement occurred, we would want to look at the context surrounding the Scheduled Task operations. As with WMI, Scheduled Tasks is used to execute a payload or command on a remote system. The adversary must upload the payload to the host prior to execution. As noted earlier, we determined SMB is the most common method to accomplish file transfer. Once the payload is executed, we expect to see the same "shared destination" artifact between the two hosts involved to be present in the network activity. Given this, Scheduled Task-based lateral movement follows these steps:

1. Attacker on Host A uploads a payload to Host B via SMB, which will be the target of the lateral movement
2. Attacker on Host A remotely creates a task on Host B to execute their payload or command, calling the SchRpcRegisterTask operation
3. Attacker on Host A remotely runs the newly created task on Host B calling the SchRpcRun operation
4. Host B initiates a connection to an address/host that Host A was also communicating with (a shared C2 destination between the two hosts)
5. (Optional) Attacker on Host A remotely deletes the created task on Host B calling the SchRpcDelete operation

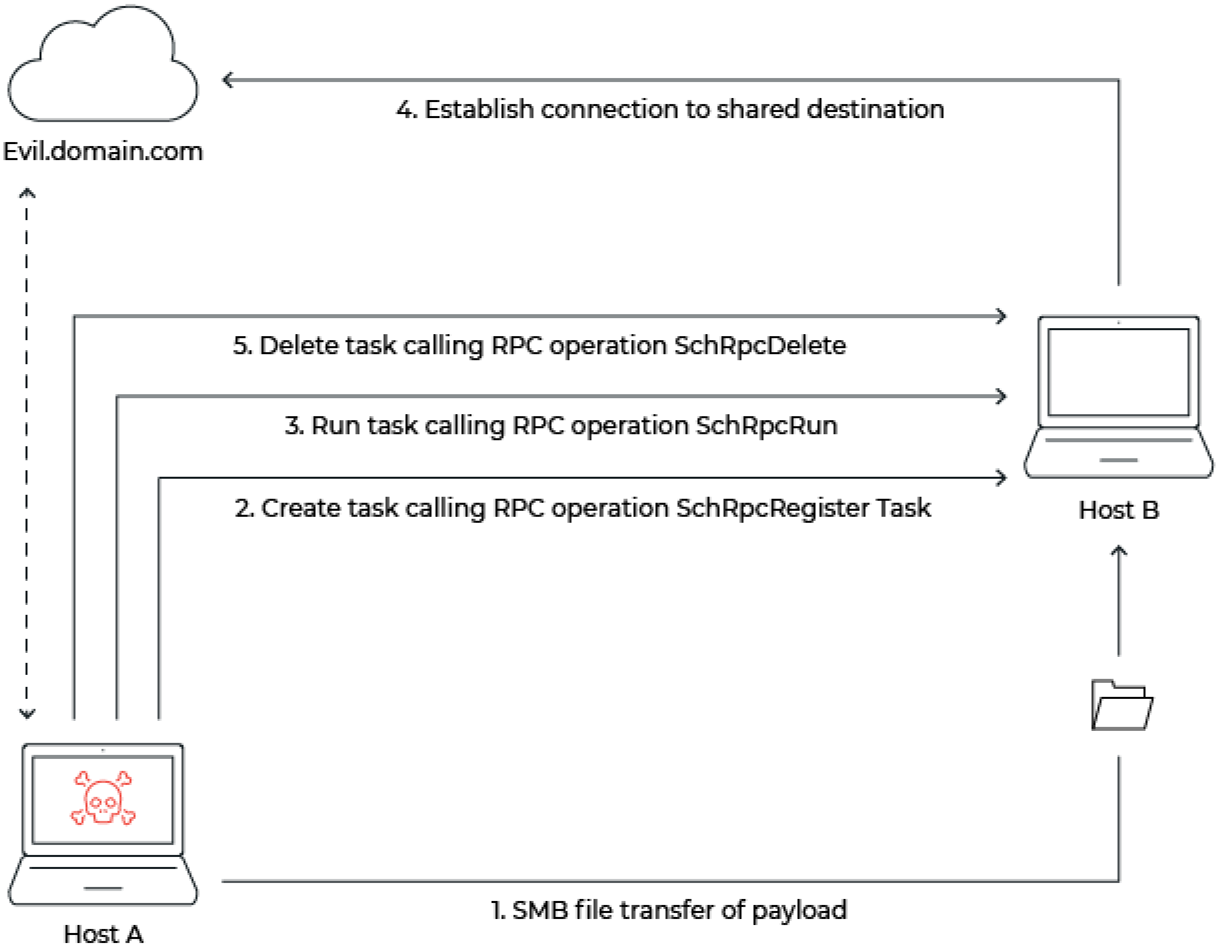
Evil.domain.com
4. Establish connection to shared destination
5. Delete task calling RPC operation SchRpcDelete
3. Run task calling RPC operation SchRpcRun
2. Create task calling RPC operation SchRpcRegister Task
Host B
Host A
1. SMB file transfer of payload

Figure 4: Scheduled Tasks lateral movement procedural steps.

While, as the name implies, Scheduled Tasks are usually scheduled for a certain time, ATR research finds that it's common for adversaries to run the task immediately to avoid waiting for the configured schedule.

A less common but potential variation would be to modify an existing task to run the payload or command of the adversary's choice, then revert the modification to avoid detection.

1. Adversary on Host A remotely lists and checks the details of various tasks already registered on Host B, calling the SchRpcRetrieveTask and SchRpcGetTaskInfo operations.
2. Adversary on Host A remotely modifies an identified viable task to execute their payload or command on Host B, calling the SchRpcRegisterTask operation
3. Adversary on Host A remotely runs the modified task on Host B calling the SchRpcRun operation
4. Host B initiates a connection to an address/host that Host A was also communicating with (a shared destination between the two hosts)
5. (Optional) Adversary on Host A remotely reverts the modified task on Host B calling the SchRpcRegisterTask operation again

Follow-Up

When investigating either WMI or Scheduled Task lateral movement, some follow-up questions should be asked:

- What type of system and role does Host A have in the environment, and should it be conducting this activity?
- What file was copied over? Was it an executable file type?
- Was the shared destination an IP address or a hostname? What intelligence do you have on it?
- What account was used to authenticate to Host B? What privileges does it hold? Should it be used in this context?
- What type of authentication was observed? NTLM? Kerberos? (Could imply a particular way the tooling was used.)
- What time of day was the activity, and is that normal for the hosts and users involved?

Additionally, there is the historical context of the involved hosts to consider, if possible:

- Has Host A been executing the WMI/Scheduled Task-related operations regularly in the past? With other hosts?
- Does the user that authenticated to Host B have a history with that host?
- Have the two hosts previously interacted in the past? Was/is there an expectation for them to interact?

Conclusion

Once an adversary gains access to an environment, lateral movement becomes a necessary and frequent activity in the attack lifecycle. Every lateral movement execution becomes an opportunity for the Blue Team to detect, investigate, and respond to the threat. Within lateral movement, various adversaries use Scheduled Tasks and WMI, from ransomware affiliates to APT actors.

These entities will continue to leverage such techniques because of their availability and the flexibility they provide. When it comes to detecting malicious use of WMI or Scheduled Tasks, it’s not as straightforward as identifying the right RPC call. However, if we take into consideration how adversaries use these methods and the context around their behavior, we begin to identify clearer detection opportunities and natural investigation follow-up.

ThreatINSIGHT: Eliminating Adversaries’ Dwell Time Advantage

SANS 2022 Cloud Security Survey