Execution

This lesson is going to cover the second stage in the MITRE ATT&CK framework, Execution (TA0002). These techniques are used to describe ways that adversaries will execute malicious code for a number of purposes, and at the time of writing currently includes 10 top-level techniques. **We will be looking at the following:**

- Scheduled task/job (5 sub-techniques)
- Windows Management Instrumentation
- User Execution (2 sub-techniques)

MITRE Technique T1053

Using scheduled tasks or jobs, adversaries can set an action to complete at a specific time and even make it reoccurring. For example, a good way to get persistence would be to initiate a connection back to a command-and-control (c2) server every hour so that the attacker can retrieve remote access via a shell and then can execute commands on the system. We could do this by using Netcat (nc.exe) and create a scheduled task that will run a specific command connecting out to our IP every 60 minutes. We could also create or modify a scheduled task that runs when the system start up, so that it will always be run (a common location for key loggers and remote access trojans).

Let's take a look at the sub-techniques for scheduled tasks and jobs:

Sub-techniques (5)

ID	Name
T1053.001	At (Linux)
T1053.002	At (Windows)
T1053.003	Cron
T1053.004	Launchd
T1053.005	Scheduled Task

- At (Linux) –
- At (Windows) –
- Cron –
- Launchd –
- Scheduled Task –

MITRE Technique T1047

Windows Management Instrumentation (WMI) is an administration feature that facilitates the management of devices and applications in a network from a Windows system. WMI provides users with information about the status of local or remote computer systems and can be used to remotely execute code on other systems. It uses the WMI service for local and remote access and the Server Message Block (SMB) and Remote Procedure Call Service (RPCS).

WMI has a number of uses such as lateral movement and discovery, but considering we are focusing on Execution in this lesson, take a look at a few highlighted entries from the Procedure Examples table below (there's far too many to cover here!):

Procedure Examples

Name	Description
APT29	APT29 used WMI to steal credentials and execute backdoors at a future time.[97]
APT32	APT32 used WMI to deploy their tools on remote machines and to gather information about the Outlook process.[98]
APT41	APT41 used WMI in several ways, including for execution of commands via WMIEXEC as well as for persistence via PowerSploit.[67]
Astaroth	Astaroth uses WMIC to execute payloads. [99]
Emotet	Emotet has used WMI to execute powershell exe.[100]
Empire	Empire can use WMI to deliver a payload to a remote host.[101]
EvilBunny	EvilBunny has used WMI to gather information about the system.[67]
FELIXBOOT	FELIXBOOT uses WMI to query the Windows Registry.[102]
FIN6	FIN6 has used WMI to automate the remote execution of PowerShell scripts.[68]
FIN8	FIN8's malicious spearphishing payloads use WMI to launch malware and spawn cmd.exe execution. FIN8 has also used WMIC during and post compromise cleanup activities.[103][104]

MITRE offer a couple of Mitigations that can be used to prevent malicious abuse of WMI. They focus on properly managing privileged accounts by only issuing them to individuals that need those privileges (think principle of least privilege!). The other suggestion is to be very restrictive regarding who can use WMI to limit the number of accounts that could abuse this if they are compromised.

Mitigations

Mitigation	Description
Privileged Account Management	Prevent credential overlap across systems of administrator and privileged accounts. [9]
User Account Management	By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI.

MITRE states that organisations should monitor for WMI usage, which we can do using System Monitoring (Sysmon) from Sysinternals. We can monitor the following event IDs (click them for more information and log examples!):

- Sysmon Event ID 19 – WmiEventFilter activity detected
- Sysmon Event ID 20 – WmiEventConsumer activity detected
- Sysmon Event ID 21 – WmiEventConsumerToFilter activity detected

Detection

Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect. Perform process monitoring to capture command-line arguments of "wmic" and detect commands that are used to perform remote behavior. [9]

MITRE Technique T1204

This technique is related to a user interacting with a malicious URL (sub-technique 1) or a malicious file (sub-technique 2), and is very closely tied with Phishing as an Initial Access technique. By convincing a user to click a link or run an attachment using social engineering tactics (we covered these in the Phishing Analysis domain!) the adversary can achieve code execution on a system without first having initial access. Alternatively, this technique can be used from inside the network, sending phishing emails internally or uploading a malicious file to a shared drive or file sharing server, and then entice users to click on it, running malicious code.

Sub-techniques (2)

ID	Name
T1204.001	Malicious Link
T1204.002	Malicious File

MITRE offers 4 Mitigations that we could use. Application whitelisting is the process of preventing any unapproved executables from running, working to prevent malicious code execution. Network Intrusion Prevention systems (NIPS) can work to identify requests to malicious or suspicious web resources and block the connection before any files are downloaded. User awareness training is a huge part of an effective information security program. End users should be trained to spot phishing emails and not interact with them (replying, clicking links or attachments).

Mitigations

Mitigation	Description
Execution Prevention	Application control may be able to prevent the running of executables masquerading as other files.
Network Intrusion Prevention	If a link is being visited by a user, network intrusion prevention systems and systems designed to scan and remove malicious downloads can be used to block activity.
Restrict Web-Based Content	If a link is being visited by a user, block unknown or unused files in transit by default that should not be downloaded or by policy from suspicious sites as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some download scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious files.
User Training	Use user training as a way to bring awareness to common phishing and spearphishing techniques and how to raise suspicion for potentially malicious events.

In the Detection section we have two great recommendations for protecting against User Execution. The first is to monitor commands that are entered into processes such as CMD.exe (Windows Command Shell) and PowerShell.exe as well as monitoring applications that are used to compress payloads and then extract them (7Zip, WinRar, and others). The second suggestion is to use an up-to-date and commercial anti-virus solution that will detect malicious files using different techniques such as file analysis, pattern-based detection, and repetitional checks. An endpoint detection and response (EDR) solution should also be considered to monitor and report on malicious activity, providing analysts with a platform where they can investigate process activity, such as Winword.exe spawning a child process of CMD.exe (this is NOT normal activity, and will be a malicious macro inside a Microsoft Office document that is opening a command prompt to call back to the C2 server and download additional malware).

Detection

Monitor the execution of and command-line arguments for applications that may be used by an adversary to gain Initial Access that require user interaction. This includes compression applications, such as those for zip files, that can be used to Deobfuscate/Decode Files or Information in payloads.

Anti-virus can potentially detect malicious documents and files that are downloaded and executed on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the file is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning powershell.exe).