Defense Evasion

We're currently adding this new lesson, please be patient while we add the content.

This lesson is going to cover the fifth stage in the MITRE ATT&CK framework, Defense Evasion. These techniques are used to describe ways that adversaries will work to evade or disable security defenses such as antivirus, endpoint detection and response, logging, and human analysts to ensure they can remain in the network for as long as possible. At the time of writing this stage currently includes a crazy 38 top-level techniques! **We will be looking at the following:**

- Impair Defenses, T1562 (6 sub-techniques)
- Indicator Removal on Host, T1070 (6 sub-techniques)
- Rootkit, T1014

MITRE Technique T1562

The Impair Defenses technique is all about disruption the normal operation of security tools, from firewalls and antivirus to actually targeting logging and aggregation tools to prevent or disrupt the flow of events likely into a SIEM platform to make it harder for both the SIEM correlation engine and human analysts to detect the malicious activity. Let's take a look at the sub-techniques:

Sub-techniques (6)

ID	Name
T1562.001	Disable or Modify Tools
T1562.002	Disable Windows Event Logging
T1562.003	HISTCONTROL
T1562.004	Disable or Modify System Firewall
T1562.006	Indicator Blocking
T1562.007	Disable or Modify Cloud Firewall

- **Disable of Modify Tools –** "Adversaries may disable security tools to avoid detection. This can take the form of killing security software or event logging processes, or other methods to interfere with security tools scanning or reporting information."
- **Disable Windows Event Logging –** "Adversaries may disable Windows event logging to limit data that can be leveraged for detections and audits. This data is used by security tools and analysts to generate detections. By disabling Windows event logging, adversaries can operate while leaving less evidence of a compromise behind."
- **HISTCONTROL –** "Adversaries may configure `HISTCONTROL` to not log all command history. The `HISTCONTROL` environment variable keeps track of what should be saved by the `history` command and eventually into the `~/.bash_history` file when a user logs out."
- **Disable or Modify System Firewall –** "Adversaries may disable or modify system firewalls in order to bypass controls limiting network usage. Changes could be disabling the entire mechanism as well as adding, deleting, or modifying particular rules."
- **Indicator Blocking –** "An adversary may attempt to block indicators or events typically captured by sensors from being gathered and analyzed. These settings may be stored on the system in configuration files and/or in the Registry as well as being accessible via administrative utilities such as PowerShell or Windows Management Instrumentation."
- **Disable or Modify Cloud Firewall –** "Adversaries may disable or modify a firewall within a cloud environment to bypass controls that limit access to cloud resources."

MITRE have a number of good mitigations to prevent adversaries using this technique. Having proper file and process permissions will prevent an attacker from making changes or disrupting tools or logging. If they are able to obtain administrative access or domain administrator access this could likely be bypassed, which is why it's also important to limit the number of administrator accounts and ensure they are appropriately secured. The suggestions also include ensuring the the Registry is locked down, and only specific users are able to make changes because this is such an important part of any Windows system, and if an attacker gains full access they can take a number of actions, from disabling logging or security tools to ensuring persistence through reboots.

Mitigations

Mitigation	Description
Restrict File and Directory Permissions	Ensure proper process and file permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
Restrict Registry Permissions	Ensure proper Registry permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
User Account Management	Ensure proper user permissions are in place to prevent adversaries from disabling or interfering with security/logging services.

We also have some pretty standard detection suggestions. Depending on the tools that an organisation is using they will need to adapt their detections based on the registry keys or core files created. As mentioned in the screenshot below logging should be enabled for processes (using Sysmon) and command-line (CMD and PowerShell) to detect any processes being killed.

Detection

Monitor processes and command-line arguments to see if security tools or logging services are killed or stop running. Monitor Registry edits for modifications to services and startup programs that correspond to security tools. Lack of log events may be suspicious.

Monitor environment variables and APIs that can be leveraged to disable security measures.

MITRE Technique T1070

When adversaries take actions on a system, they're going to be leaving behind a number of artefacts that defenders can discover and use to create a timeline of actions that were conducted. From file times being modified when they're opened to system logs and open ports, if adversaries want to survive in the network for extended periods of time they will need to remove these artefacts before they're discovered. Some examples include:

- **Deleting bash history**
- **Deleting files** (such as malicious files downloaded to the system by the adversary)
- **Deleting raw log files** (provided the adversary has SYSTEM or SUDO privileges)
- **Timestomping** (Changing file timestamps so its not immediately clear files were accessed)
- **And more**

Before we jump into some real-world examples, let's take a look at the sub-techniques:

Sub-techniques (6)

ID	Name
T1070.001	Clear Windows Event Logs
T1070.002	Clear Linux or Mac System Logs
T1070.003	Clear Command History
T1070.004	File Deletion
T1070.005	Network Share Connection Removal
T1070.006	Timestomp

- **Clear Windows Event Logs –**
- **Clear Linux or Mac System Logs –**
- **Clear Command History –**
- **File Deletion –**
- **Network Share Connection Removal –**
- **Timestomp –**

The below screenshot shows part of the Procedure Examples table, and we have some really interesting cases here! For the malware named 'Goopy' we can see that it will use emails for command-and-control, and then delete them

so they're not present on the infected system. PoetRAT at the bottom is clever, and can detect when it is ran in a sandbox and will delete itself to prevent researchers or security from conducting analysis.

Procedure Examples

Name	Description
Bankshot	Bankshot deletes all artifacts associated with the malware from the infected machine.[3]
Goopy	Goopy has the ability to delete emails used for C2 once the content has been copied.[10]
MAZE	MAZE has used the "Wow64RevertWow64FsRedirection" function following attempts to delete the shadow volumes, in order to leave the system in the same state as it was prior to redirection.[8]
Misdat	Misdat is capable of deleting Registry keys used for persistence.[4]
Orz	Orz can overwrite Registry settings to reduce its visibility on the victim.[1]
PoetRAT	PoetRAT has the ability to overwrite scripts and delete itself if a sandbox environment is detected.[6]

For mitigations, MITRE keeps it simple. Hide your logs, ensure no one can tamper with them, and minimalise the time between a log being generated and it being forwarded to an aggregation point so it can be ingested by the SIEM. Once it's stored there, the chance of it being modified or deleted by an adversary is extremely low.

Mitigations

Mitigation	Description
Encrypt Sensitive Information	Obfuscate/encrypt event files locally and in transit to avoid giving feedback to an adversary.
Remote Data Storage	Automatically forward events to a log server or data repository to prevent conditions in which the adversary can locate and manipulate data on the local system. When possible, minimize time delay on event reporting to avoid prolonged storage on the local system.
Restrict File and Directory Permissions	Protect generated event files that are stored locally with proper permissions and authentication and limit opportunities for adversaries to increase privileges by preventing Privilege Escalation opportunities.

For detection its straightforward, monitor file access and changes to the location where logs are stored locally on a system before they're sent off to a SIEM or central repository.

Detection

File system monitoring may be used to detect improper deletion or modification of indicator files. Events not stored on the file system may require different detection mechanisms.

MITRE Technique T1014

Rootkits are malicious programs that work to hide the existence of malware by intercepting and changing operating system API calls that supply system information so that malicious elements are not reported. Rootkits can be found at the user or kernel level in the operating system or lower, to include a hypervisor and the Master Boot Record (often referred to as Bootkits),

Let's take a look at some real-world examples where rootkits have been used! In the below screenshot of the Procedure Examples table we can see that APT28 have previously used a UEFI Rootkit known as LoJax. We can also see that APT41 have built and deployed rootkits for Linux systems (adversaries have also created rootkits for Windows, Linux, and Mac OS X!).

Procedure Examples

Name	Description
APT28	APT28 has used a UEFI (Unified Extensible Firmware Interface) rootkit known as LoJax.[20][13]
APT41	APT41 deployed rootkits on Linux systems.[21]
Hacking Team UEFI Rootkit	Hacking Team UEFI Rootkit is a UEFI BIOS rootkit developed by the company Hacking Team to persist remote access software on some targeted systems.[11]
HiddenWasp	HiddenWasp uses a rootkit to hook and implement functions on the system.[12]
HIDEDRV	HIDEDRV is a rootkit that hides certain operating system artifacts.[13]

Previous Topic
Mark Complete
Next Topic
Back to Lesson