Terminating W3WP spawned processes

Scenario: After adversaries have implanted a webshell, it’s easy for them to gain a reverse shell via tools such as netcat, the Metasploit framework, or Powercat. The W3WP process itself is the IIS process running within Exchange. Whenever that process spawns the cmd.exe or powershell.exe processes, you have a match. The reverse shell allows the target machine to communicate back to the attacking machine. The attacking machine has a listener port on which it receives the connection, which by using, code or command execution is achieved. You would like an automated way to terminate W3WP spawned processes.

Prerequisites

To succeed in implementing this use case, you need the following dependencies, resources, and information.

- People: Threat hunter
- Technologies:
  - Splunk SOAR
  - Splunk Enterprise Security
  - Splunk Enterprise Security Content Updates
  - Playbook: Delete detected files
  - Splunk Add-on for Phantom
- Data: Windows event data

How to use Splunk software for this use case

This playbook terminates the process spawned by the W3WP process. Then, the format block creates SPL to search Splunk SOAR for any child processes of the process identified in the detection. Splunk SOAR runs the Splunk search to find the process IDs of child processes that were run. Finally, the playbook runs the terminate process command for any child processes that were found in the previous search. To use the playbook:

2. Enable the Send to Phantom Adaptive Response Action in the Enterprise Security correlation search. After a web shell is written, the detection sends the event to Splunk SOAR.
3. If you haven’t previously used this playbook, configure and activate it.
   1. Navigate to Home>Playbooks and search for terminate spawned processes. If it’s not there, click Update from Source Control and select Community to download new community playbooks.
   2. Click the playbook name to open it.
   3. Resolve the playbook import wizard and set the playbook to Active.
   4. Save the playbook and then run it.
4. Adjust the playbook as needed, according to your results.

The process that you want to terminate could actually be a child process of the one picked up in the detection. For example, the adversary could launch Powershell with a command like "cmd /c powershell.exe". The detection would pick up the execution of cmd, so you'd want the playbook to look for both that process and any child processes that have been run.

Results

The PowerShell command is reversed back to the listening instance, and is then terminated due to the playbook being run to terminate it. Adversaries will attempt to perform additional actions like create persistence, move laterally, and more, but automated processes in your kit bag can help deal with issues very quickly when they do pop up.

Additional resources

These additional Splunk resources might help you understand and implement this use case:

- Blog: [Automated clean-up of HAFNIUM shells and processes with Splunk SOAR](#)
- Blog: [Detecting HAFNIUM Exchange Server zero-day activity in Splunk](#)