Sc.exe manipulating Windows services

You might want to look for arguments to sc.exe that indicate the creation or modification of a Windows service when doing the following:

- Detecting techniques in the Orangeworm attack group
- Recognizing improper use of system administration tools
- Detecting lateral movement with Active Directory data

Prerequisites

Content developed by the Splunk Security Research team requires the use of consistent, normalized data provided by the Common Information Model (CIM). This search requires the Endpoint data model. For information on installing and using the CIM, see the Common Information Model documentation.

Example

Attackers often create a new service to host their malicious code, or they may take a non-critical service or one that is disabled and modify it to point to their malware, enabling the service if necessary. It is unusual for a service to be created or modified using the sc.exe utility, so you want to look for instances of this occurring so you can investigate further.

To optimize the search shown below, you should specify an index and a time range.

1. Ensure that your deployment is ingesting records process activity from your hosts to populate the Endpoint data model in the Processes node. You must also be ingesting logs with both the process name and command line from your endpoints. The command-line arguments are mapped to the process field in the Endpoint data model.

2. Run the following search:

   |tstats summariesonly=true allow_old_summaries=true values(Processes.process) AS process min(_time) AS firstTime max(_time) AS lastTime FROM datamodel=Endpoint.Processes WHERE Processes.process_name = sc.exe (Processes.process="* create *" OR Processes.process="* config ") BY Processes.process_name Processes.parent_process_name Processes.dest Processes.user
   |rename "Processes." as "*"
   |convert timeformat="%m/%d/%Y %H:%M:%S" ctime(firstTime)
   |convert timeformat="%m/%d/%Y %H:%M:%S" ctime(lastTime)
Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.

<table>
<thead>
<tr>
<th>Splunk Search</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tstats summariesonly=true allow_old_summaries=true values(Processes.process) AS process min(_time) AS firstTime max(_time) AS lastTime FROM datamodel=Endpoint.Processes WHERE Processes.process_name = sc.exe (Processes.process=&quot;* create &quot; OR Processes.process=&quot;* config *&quot;) BY Processes.process_name Processes.parent_process_name Processes.dest Processes.user</td>
</tr>
<tr>
<td>rename &quot;Processes.<em>&quot; as &quot;</em>&quot;</td>
<td>Rename the data model object for better readability.</td>
</tr>
<tr>
<td></td>
<td>convert timeformat=&quot;%m/%d/%Y %H:%M:%S&quot; ctime(firstTime)</td>
</tr>
<tr>
<td></td>
<td>convert timeformat=&quot;%m/%d/%Y %H:%M:%S&quot; ctime(lastTime)</td>
</tr>
</tbody>
</table>

Result

Using sc.exe to create or configure Windows services is uncommon. However, there may be legitimate instances of this behavior. It is important to validate and investigate as appropriate.

Investigate web and authentication activity on the destination. If you have the Splunk Enterprise Security app, you can leverage the Threat Intel Framework to watch for traffic from known malicious IP addresses.

For additional information about this search, such as its applicability to common frameworks and standards, see this project on GitHub.