Web shell present in web traffic events

Your company uses SolarWinds Orion business software, which is vulnerable to the Supernova in-memory web shell attack. You know that because it runs in-memory, detection and forensic analysis post-breach are difficult. You want to search your web data to see if the web shell exists in memory. This is done by looking for calls to the known to be implemented and called in the trojanized versions of the API. Finding these in your web logs would be an indicator of compromise.

Required data

- Web server data

Option 1 - Normalized data

1. Ensure that your deployment is ingesting endpoint logs from your various systems and populating the Web data model.

2. Run the following search. You can optimize it by specifying an index and adjusting the time range.

   ```splunk
   | tstats count FROM datamodel=Web.Web WHERE
   web.url=*logoimagehandler.ashx*codes* OR
   Web.url=*logoimagehandler.ashx*clazz* OR
   Web.url=*logoimagehandler.ashx*method* OR
   Web.url=*logoimagehandler.ashx*args* groupby Web.src Web.dest Web.url
   Web.vendor_product Web.user Web.http_user_agent _time span=1s
   ```

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>
</table>
Option 2 - Stream data

This sample search uses Stream HTTP data. You can replace this source with Bro/Zeek or any other data set that contains information around http and URI filenames and parameters used in your organization.

1. Ensure that you have installed and configured the Splunk App for Stream.
2. Run the following search. You can optimize it by specifying an index and adjusting the time range.

```
sourcetype=stream:http Dest_ip=< IP of SolarWinds system you are investigating>
(url=*logoimagehandler.ashx*codes* OR Web.url=*logoimagehandler.ashx*clazz* OR
Web.url=*logoimagehandler.ashx*method* OR
Web.url=*logoimagehandler.ashx*args*)
| table _time src_ip src_port dest_ip dest_port url transport status
```

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>sourcetype=stream:http</td>
<td>Search only Stream HTTP data.</td>
</tr>
<tr>
<td>Dest_ip =&lt; IP of SolarWinds system you are investigating&gt;</td>
<td>Search the system you want to investigate for this inbound traffic.</td>
</tr>
<tr>
<td>(url=<em>logoimagehandler.ashx</em>codes* OR Web.url=<em>logoimagehandler.ashx</em>clazz* OR Web.url=<em>logoimagehandler.ashx</em>method* OR Web.url=<em>logoimagehandler.ashx</em>args*)</td>
<td>Search for these URLs being requested from the web client.</td>
</tr>
<tr>
<td></td>
<td>Display the results in a table with columns in the order shown.</td>
</tr>
</tbody>
</table>

Next steps

If these URLs found in the web server logs are called successfully, you can assume that the web server is infected with the webshell. Investigate further and then run your incident response plan to clean the web server of the malware.

Finally, you might be interested in other processes associated with the Detecting Supernova web shell malware use case.