Monitoring for Windows updates

Ensuring your endpoints are patched and updated regularly is a best practice to reduce risk of compromise to your environment. Attackers can exploit known vulnerabilities that could otherwise be mitigated by applying routine security patches.

This search is designed to help you monitor your endpoints for system patches or updates so you can identify any systems that are not successfully updated in a timely matter.

Required data

- To complete this process, your deployment needs to ingest normalized endpoint data, populating the Updates data model in the Common Information Model (CIM). For information on installing and using the CIM, see the Common Information Model documentation.
- Your deployment also needs to ingest Windows event logs or the Windows Update log via a Universal Forwarder on the Windows endpoints you want to monitor. The Splunk Add-on for Microsoft Windows should be also be installed and configured to properly parse Windows events in Splunk.
- Instead of using Windows events, you can also run this process with other data sources which can populate the Updates data model, including vulnerability management systems.

How to use Splunk software for this use case

This search looks for Windows endpoints that have not generated an event indicating a successful Windows update in the last 60 days. Windows updates are typically released monthly and applied shortly afterwards. An endpoint that has not successfully applied an update in this time frame indicates the endpoint is not regularly being patched.

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

```
| tstats summariesonly=false allow_old_summaries=true max(_time) AS lastTime FROM datamodel=Updates
WHERE ("Updates.status"="Installed" "Updates.vendor_product"="Microsoft Windows") BY "Updates.dest",
"Updates.status", "Updates.vendor_product"
| rename "Updates.dest" AS Host
| rename "Updates.status" AS "Update Status"
| rename "Updates.vendor_product" AS Product
| eval isOutlier=if((lastTime <= relative_time(now(),"-60d@d")),1,0)
| search isOutlier=1
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(lastTime)
| rename lastTime AS "Last Update Time"
| table Host, "Update Status", Product, "Last Update Time"
```
Next steps

The content in this article comes from Splunk Enterprise Security (ES). As a Splunk premium security solution, ES solves a wide range of security analytics and operations use cases including continuous security monitoring, advanced threat detection, compliance, incident investigation, forensics and incident response. Splunk ES delivers an end-to-end view of an organization’s security posture with flexible investigations, unmatched performance, and the most flexible deployment options offered in the cloud, on-premises, or hybrid deployment models. If you have questions about this use case, see the Security Research team’s support options on GitHub.

In addition, these Splunk resources might help you understand and implement this use case:

- E-book: Ransomware, malware and cyberthreats
- Blog: Hunting with Splunk: The basics