Detecting BlackMatter ransomware

Applicability

- Product: Splunk Cloud Platform or Splunk Enterprise
- Feature: Search
- Data: Endpoint data

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

Scenario

BlackMatter ransomware campaigns involve the use of ransomware payloads along with exfiltration of data. Malicious actors demand payment for ransom of data, as well as threaten the deletion and exposure of exfiltrated data.

You are an analyst responsible for your organization's overall security posture. You need to be able to detect and investigate unusual activities that might relate to BlackMatter ransomware. These searches will help you detect and investigate these infections.

Detection searches

► Registry modified to implement auto admin logon to a host

- 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.

- To successfully implement this search, your deployment needs to ingest process informations that includes the name of the process responsible for the changes from your endpoints into the Endpoint data model in the Registry node. Ensure that this registry is included in your config files to be monitored.

This search is designed to detect a suspicious registry modification that implements auto admin logon to a host. This technique is seen in BlackMatter ransomware to automatically log on to the compromised host, after a safe mode boot has been triggered, and then encrypt the whole network.

Auto admin logon is not a common practice and should be treated as suspicious.

| tstats summariesonly=false allow_old_summaries=true count FROM datamodel=Endpoint.Registry WHERE ("Registry.registry_path"="SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" ("Registry.registry_value_name"=DefaultPassword OR "Registry.registry_value_name"=DefaultUserName)) BY _time span=1h, "Registry.dest", "Registry.user", "Registry.registry_path", "Registry.registry_value_name", "Registry.process_guid", "Registry.registry_value_data", "Registry.registry_key_name" |

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
| rename "Registry.*" AS "***"  
| rename process_guid AS proc_guid  
| join _time,proc_guid [  
| rename "Processes.*" AS "***"  
| rename process_guid AS proc_guid  
| fields + _time, dest, user, parent_process_name, parent_process, process_name, process_path, process, proc_guid, registry_path, registry_value_name, registry_value_data, registry_key_name]  
| table _time, dest, user, parent_process_name, parent_process, process_name, process_path, process, proc_guid, registry_value_path, registry_value_name, registry_value_data, registry_key_name  
| search

► Bcdedit command back to normal mode boot

- 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.

- To run this search, your deployment needs to ingest logs with the process name, parent process, and command-line executions from your endpoints. If you are using Sysmon, you must have at least version 6.0.4 of the Sysmon TA. Tune and filter known instances where renamed rundll32.exe may be used.

This search is designed to detect a suspicious bcdedit command line, which is used to configure a host from safe mode back to a normal boot configuration. This technique is seen in BlackMatter ransomware, which forces a compromised host to boot in safe mode to perform encryption, and then brings it back to normal boot settings using the bcdedit deletevalue command.

| rename "Processes.*" AS "***"  
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(firstTime)  
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(lastTime)

► Change to safe mode with network config

- 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.

- To run this search, your deployment needs to ingest logs with the process name, parent process, and command-line executions from your endpoints. If you are using Sysmon, you must have at least version 6.0.4 of the Sysmon TA. Tune and filter known instances where renamed rundll32.exe may be used.

This search is designed to detect a suspicious bcdedit command line, used to configure the host to boot in safe mode with network config. This technique is seen in BlackMatter ransomware, which forces a compromised host to...
boot in safe mode to perform encryption and then brings it back to normal boot settings using bcdedit deletevalue command.

```bash
| rename "Processes.*" AS "*
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(firstTime)
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(lastTime)
```

► Services commonly stopped by ransomware

To successfully implement this search, your deployment needs to ingest logs with the 7036 EventCode ScManager in System audit logs from your endpoints.

This search detects terminations of services that are commonly stopped by ransomware before files in a compromised machine are encrypted. This technique is commonly used by ransomware to avoid exception errors while accessing targeted files because of the open handle of those services to the targeted file.

False positives from this search may occur since legitimate admin activities or the installation of updates can involve the stopping of these services.

```bash
| search (EventCode=7036 Message="*service entered the stopped state*" eventtype=wineventlog_system (Message="*VSS*" OR Message="*Volume Shadow Copy*" OR Message="*backup*" OR Message="*memtas*" OR Message="*mepocs*" OR Message="*sophos*" OR Message="*sql*" OR Message="*svc$*" OR Message="*veeam*"))
| stats count min(_time) AS firstTime max(_time) AS lastTime BY EventCode Message dest Type
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(lastTime)
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(firstTime)
```

► Wallpaper modification

To run this search, your deployment needs to ingest logs with the deleted target file name, process name, and process ID from your endpoints. If you are using Sysmon, you must have at least version 6.0.4 of the Sysmon TA.

This search identifies suspicious modification of the registry to change the wallpaper of a compromised machine. This technique is commonly seen in ransomware, where a bitmap file is created that contains a note that the machine has been compromised. Then, that image is set as a wallpaper.

False positives from this search may occur since legitimate third party tools can be used to change wallpapers.
To run this search using **Sysmon** data, you need a Splunk Universal Forwarder on each endpoint from which you want to collect data.

This search looks for instances where a large number of ransomware notes are files created in the infected machine. These notes often have file extensions of `.txt`, `.html`, or `.hta`.

```splunk
| search (EventCode=11 (source=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational OR sourcetype=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational) (file_name="*\hta" OR file_name="*\html" OR file_name="*\txt"))
| stats min(_time) AS firstTime max(_time) AS lastTime dc(TargetFilename) AS unique_readme_path_count values(TargetFilename) AS list_of_readme_path BY Computer Image file_name | where (unique_readme_path_count >= 50)
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(firstTime)
| convert timeformat="%Y-%m-%dT%H:%M:%S" ctime(lastTime)
```

Need more help with this search? Click [here](https://www.example.com).

---

### Additional resources

This use case is included within **Splunk Enterprise Security**, a Splunk app that provides prebuilt content and searches to help answer root-cause questions in real-time about malicious and anomalous events in your IT infrastructure. In addition, these Splunk resources might help you understand and implement this use case:

- **Use case:** [Detecting ransomware attacks](#)
- **Use case:** [Investigating ransomware attacks](#)
- **E-book:** [Ransomware, malware and cyberthreats](#)
- **Webinar:** [Detection of ransomware and prevention strategies](#)
- **Blog:** [Operationalize ransomware detections quickly and easily with Splunk](#)

---

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.