Detecting Ransomware's Stealthy Boot Configuration Edits

binarydefense.com/detecting-ransomwares-stealthy-boot-configuration-edits/

April 27, 2022

Written By: Binary Defense Threat Researcher @shade_vx

This blog post focuses on threat hunting methods and detections for a commonly observed technique used by Ransomware-as-a-Service (RaaS) operators. Such threat actors have often been observed altering boot loader configurations using the built-in Windows tool bcdedit.exe (Boot Configuration Data Edit) in order to:

- Modify Boot Status Policies
- Disable Recovery Mode
- Enable Safe Mode

The hypothesis that we are using to develop these hunting queries is that threat actors (such as Snatch and REvil) don't necessarily have to use bcdedit to modify boot loader configurations but could implement code that directly modifies the Windows registry keys that determine those configurations. Last year, the researcher am0nsec released proof of concept code demonstrating how to do exactly this on Windows 10 systems. We wanted to be sure that we were able to not only detect such activity, but also on Windows 7, 8.1, and 11 systems where the relevant registry key is stored under a different Globally Unique Identifier (GUID).

Our research is building upon prior work by the Specter Ops researcher Michael Barclay, who published an in-depth blog about hunting for such activity on Windows 10. The bcdedit.exe commands that attackers use to modify boot configuration are below. Please note that other utilities such as the Windows System Configuration Utility (msconfig.exe) can also be used to modify the boot configuration data. However, alternatives will not be covered in this paper as they are not command line applications and thus cannot be used out of user interface access.

Boot Status Policy

The normal way to edit the boot status policy is to use bcdedit with these command line arguments:

bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures

This will modify the “boot status policy” settings and force the system to boot normally, rather than into the Windows Recovery Environment (Windows RE), if there is a failed shutdown, failed boot, or other error during the startup process. Threat actors disable this in order to prevent system administrators from accessing the System Image Recovery feature in the Windows RE.

Recovery Mode

The usual method for disabling recovery mode with bcdedit is like this:

bcdedit.exe /set {default} recoveryenabled no

This command disables the Windows RE entirely. Changing the boot status policy with the previous command will stop the boot loader from loading the recovery environment when there are startup errors, but this setting will prevent system administrators from loading it manually.

Safeboot

To change the Safeboot options, bcdedit is used with these command line arguments:

bcdedit.exe /set {default} safeboot minimal

This command changes the configuration which determines whether the system will boot into Safe Mode the next time it is restarted. The reason that this is being modified is not to prevent recovery so much as prevent detection, as not all Endpoint Detection and Response (EDR) solutions and Anti-Virus (AV) software will be running in Safe Mode. For example, Windows Defender does not run in Safe Mode. Therefore, any actions conducted by a threat actor (e.g., encryption of files) will not be monitored and consequently will most likely not be prevented.

Prior research into these techniques mentioned that the registry keys storing these boot loader configuration items were Windows version specific, and only detailed detections that are valid for Windows 10. The way that we went about determining what those registry keys were for other Windows versions was to simply set up VMs running Windows 7, 8.1, and 11 respectively, and to run the three aforementioned bcdedit.exe commands while doing a capture with the Windows SysInternals tool Procmon. The logs generated by this tool are notoriously noisy, but it was easy to filter down to the relevant logs by adding two filters, one excluding any process not called bcdedit.exe, and the other excluding any operation that was not RegSetValue.

The following queries were tested across multiple enterprise environments with zero false positives in a 60-day time frame. Modifications of these settings are rare enough that all of these queries are suitable as detections surfaced to a SOC.

Detections

Carbon Black

Windows 7:

regmod_name:(*BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\250000e0* OR *BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\16000009* OR *BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\25000080*)

Windows 8.1:

regmod_name:(*BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\250000e0* OR *BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\16000009* OR *BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\25000080*)

Windows 10:

regmod_name:(*BCD00000000\\Objects\\\{9f83643f\-4a91\–11e9\–9501\-b252ac81e352\}\\Elements\\250000E0* OR *BCD00000000\\Objects\\\{9f83643f\-4a91\–11e9\–9501\-b252ac81e352\}\\Elements\\250000E0* OR *BCD00000000\\Objects\\\{9f83643f\-4a91\–11e9\–9501\-b252ac81e352\}\\Elements\\16000009*)

Windows 11:

regmod_name:(*BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\250000e0* OR *BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\16000009* OR *BCD00000000\Objects\{ea075dc0-83af-

11ec-9994-82f1525d1096}\Elements\25000080*)

CrowdStrike

Windows 7:

(event_simpleName=AsepValueUpdate OR event_simpleName=SuspiciousRegAsepUpdate OR event_simpleName=RegistryOperationDetectInfo) AND (RegObjectName="*BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\250000e0*" OR RegObjectName="*BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\16000009*" OR RegObjectName="*BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\25000080*")

Windows 8.1:

event_simpleName=AsepValueUpdate OR event_simpleName=SuspiciousRegAsepUpdate OR event_simpleName=RegistryOperationDetectInfo) AND (RegObjectName="*BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\250000e0*" OR RegObjectName="*BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\16000009*" OR RegObjectName="*BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\25000080*")

Windows 10:

event_simpleName=AsepValueUpdate OR event_simpleName=SuspiciousRegAsepUpdate OR event_simpleName=RegistryOperationDetectInfo) AND (RegObjectName="*BCD00000000\\Objects\\{9f83643f-4a91–11e9–9501-b252ac81e352}\\Elements\\25000080*" OR RegObjectName="*BCD00000000\\Objects\\{9f83643f-4a91–11e9–9501-b252ac81e352}\\Elements\\250000E0*" OR RegObjectName="*BCD00000000\\Objects\\{9f83643f-4a91–11e9–9501-b252ac81e352}\\Elements\\16000009*")

Windows 11:

event_simpleName=AsepValueUpdate OR event_simpleName=SuspiciousRegAsepUpdate OR event_simpleName=RegistryOperationDetectInfo) AND (RegObjectName="*BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\250000e0*" OR RegObjectName="*BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\16000009*" OR RegObjectName="*BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\25000080*")

Microsoft Sentinel and Defender for Endpoint

Windows 7:

```
DeviceRegistryEvents
| where TimeGenerated > ago(90d)
where ActionType == “RegistryValueSet”
| where RegistryKey has_any (@”BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\250000e0″, @”BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\16000009″, @”BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\25000080″)
```

Windows 8.1:

```
DeviceRegistryEvents
| where TimeGenerated > ago(90d)
| where ActionType == “RegistryValueSet”
| where RegistryKey has_any (@”BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\250000e0″, @”BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\16000009″, @”BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\25000080″)
```

Windows 10:

```
DeviceRegistryEvents
| where TimeGenerated > ago(90d)
| where ActionType == “RegistryValueSet”
| where RegistryKey has_any (@”BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\25000080″, @”BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\250000E0″, @”BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\16000009″)
```

Windows 11:

```
DeviceRegistryEvents
| where TimeGenerated > ago(90d)
| where ActionType == “RegistryValueSet”
| where RegistryKey has_any (@”BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\250000e0″, @”BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\16000009″, @”BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\25000080″)
```

SentinelOne

Windows 7:

EventType = “Registry Value Modified” and RegistryKeyPath In Contains Anycase (“BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\250000e0”, “BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\16000009”, “BCD00000000\Objects\{8c07be1f-21bb-11e8-9c5d-d181d62e5fbf}\Elements\25000080”)

Windows 8.1: {303a1187-f04f-11e7-ae97-d7affdbdc5e9}

EventType = “Registry Value Modified” and RegistryKeyPath In Contains Anycase (“BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\250000e0”, “BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\16000009”, “BCD00000000\Objects\{303a1187-f04f-11e7-ae97-d7affdbdc5e9}\Elements\25000080”)

Windows 10:

EventType = “Registry Value Modified” and RegistryKeyPath In Contains Anycase (“BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\25000080”, “BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\250000E0”, “BCD00000000\Objects\{9f83643f-4a91–11e9–9501-b252ac81e352}\Elements\16000009”)

Windows 11: {ea075dc0-83af-11ec-9994-82f1525d1096}

EventType = “Registry Value Modified” and RegistryKeyPath In Contains Anycase (“BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\250000e0”, “BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\16000009”, “BCD00000000\Objects\{ea075dc0-83af-11ec-9994-82f1525d1096}\Elements\25000080”)

References

https://posts.specterops.io/capability-abstraction-case-study-detecting-malicious-boot-configuration-modifications-1852e2098a65

https://github.com/vxunderground/VXUG-Papers/tree/6b5741ca4b1df25dcd453581954b92e083c054b8/Win64.VirTool.BCDEdit

https://github.com/am0nsec/vx