Defense Evasion

T1055.004 Asynchronous Procedure Call

Atomics: T1055.004

SentinelOne isn't great at detecting all 5 injection methods, only 1 indicator of **RemoteInjection** is caught (Agent v. 4.3.2.86, Liberty SP2). In the future you could probably look for unsigned processes with some sort of combination of **Cross Process** event types > ##.

Reviewing process execution data for T1055.exe, I noted 4 child calc.exe processes and 2 notepad.exe child processes with their own calc.exe children; both notepad.exe instances had 2 **Process** events despite only having one child (most with **CrossProcess** entries in_storyline but only 1 storyline_child).

T1197 BITS Jobs

Atomics: T1197

The below query will find and remote content downloads from DesktopImgDownldr or BitsAdmin processes, Start-BitsTransfer cmdlet downloads, and excludes system processes and noise with

SrcProcParentName Not In ().

```
(( TgtProcName In Contains Anycase ("bitsadmin.exe","desktopimgdownldr.exe") AND (
```

T1548.002 Bypass User Access Control

Atomics: T1548.002

Detection of UAC bypass through tampering with Shell Open for .ms-settings or .msc file types. Beyond this Atomic test, and to further UAC bypass detection, the below query includes detection for CMSTPLUA COM interface abuse by GUID. See Security-in-bits for more info about CMSTPLUA COM abuse.

Noted issues with Sentinel Agent 4.3.2.86 detecting by registry key. All registry key paths were ControlSet001\Service\bam\State\UserSettings\GUID...

```
(SrcProcCmdLine ContainsCIS "ms-settings\shell\open\command" OR SrcProcCmdLine Con
```

T1218.003 CMSTP

Atomics: T1218.003

CMSTP is rarely used within my environment, so the below detection has low false positives without filtering, though you may want to limit query to inf files located in personal/writeable directories.

```
SrcProcName = "cmstp.exe" AND SrcProcCmdLine RegExp "^.*\.(inf)"
```

T1574.012 COR_PROFILER

Atomics: T1574.012

Detection of unmanaged COR profiler hooking of .NET CLR through registry or process command.

```
(SrcProcCmdScript Contains "COR_" AND SrcProcCmdScript Contains "\Environment") OR
```

T1070.001 Clear Windows Event Logs

Atomics: T1070.001

Detects the clearing of EventLogs through wevtutil (concise) as well as Clear-EventLog through CommandLine and CommandScript objects. Powershell cmdlet detection returns a lot of noise for the CommandScripts object, so filtering out *SrcProcParentName* may be required.

```
(TgtProcName  = "wevtutil.exe" AND TgtProcCmdLine ContainsCIS "cl ") OR ((SrcProcCr
```

T1027.004 Compile After Delivery

Atomics: T1027.004

Both Atomic tests for this technique leverage csc.exe for compilation of code. The below will detect specific compilation of executables as well as dynamic compilation through detection of csc.exe creating executable files (both dll and exe). Filter noise from later portion of query using *SrcProcParentName Not In ()*.

```
(TgtProcName = "csc.exe" AND SrcProcCmdLine Contains "/target:exe") OR (SrcProcNam
```

T1218.001 Compiled HTML File

Atomics: T1218.001

Breaking down the below query, the first section will detect Atomic Test 1 where a malicious chm file spawns a process, whereas the second half of the query detects hh.exe loading a remote payloads.

```
(SrcProcName = "hh.exe" AND EventType = "Open Remote Process Handle") OR (SrcProcNa
```

T1218.002 Control Panel

Atomics: T1218.002

The below query will find all cpl files outside standard directories and all cpl files executed outside of Windows directories. First portion of query may need to be dropped if there's too much noise in your environment.

```
(TgtFileExtension = "cpl" AND TgtFilePath Does Not ContainCIS "C:\Windows" AND Tgt
```

In the future, when Process type counts are working, it may be more accurate to detect execution of cpl files where EventType **Open Remote Process Handle** or **Duplicate Process Handle** exists, though that can be added to above for filtering but would exclude Process type data.

```
SrcProcName = "rundll32.exe" AND SrcProcCmdLine ContainsCIS "Shell32.dll,Control_Ru
```

T1574.001 DLL Search Order Hijacking

Atomics: T1574.001

Detection of DLL search order hijack for AMSI bypass. Search order bypasses can target more than AMSI, so this can be expanded upon greatly by switching the `ContainsCIS` to `In Contains Anycase(dll list)`.

```
(FileFullName ContainsCIS "amsi.dll" AND FileFullName Does Not ContainCIS "System3
```

T1574.002 DLL Side-Loading of Notepad++ GUP.exe

Atomics: T1574.002

Detection for GUP.exe side-loading a dll, where executable has a display name of "WinGup for Notepad++" and has non-standard source process. Keep an eye on Cross Process events or add `AND EventType = "Open Remote Process Handle"` to the query to narrow down target (child) process.

```
TgtProcDisplayName ContainsCIS "WinGup" and SrcProcName Not In ("notepad++.exe","e
```

T1078.001 Enable Guest account with RDP and Admin

Atomics: T1078.001

Detects enabling of Guest account, adding Guest account to groups, as well as changing of Deny/Allow of Terminal Server connections through Registry changes.

```
(SrcProcCmdLine ContainsCIS "net localgroup" AND SrcProcCmdLine ContainsCIS "guest
```

T1140 Deobfuscate/Decode Files or Information

Atomics: T1140

This Atomic tests detections of certutil encoding and decoding of executables, and the replication of certutil for bypassing detection of executable encoding. Our query below will detected renamed certutil through matching of DisplayName, as well as encoding or decoding of exe files.

```
(TgtProcName != "certutil.exe" AND TgtProcDisplayName = "CertUtil.exe") OR ( TgtPr
```

T1562.002 Disable Windows Event Logging

Atomics: T1562.002

Atomic #1 - Disable IIS Logging

```
TgtProcName = "appcmd.exe" AND TgtProcCmdLine ContainsCIS "/dontLog:true" AND TgtP
```

Atomic #2 - Kill Eventlog Service Threads

Detection is specific to Invoke-Phant0m strings as the test uses it, and we're hoping to catch renamed and obfuscated versions by catching the TerminateThread call.

```
SrcProcCmdLine ContainsCIS "Invoke-Phant0m" OR SrcProcCmdScript ContainsCIS "$Kern
```

T1562.004 Disable or Modify System Firewall

Atomics: T1562.004

Atomic #1 - Linux

```
(SrcProcName In Contains ("service","chkconfig") AND SrcProcCmdLine In Contains ("
```

Atomic #2 - Disable Defender Firewall

```
TgtProcName = "netsh.exe" AND TgtProcCmdLine ContainsCIS "state off"
```

Atomic #3 - Allow SMB and RDP on Defender Firewall

```
(TgtProcName = "netsh.exe" AND TgtProcCmdLine ContainsCIS "remote desktop" AND Tgt
```

Atomic #4 AND #5 - Open Local Port on Defender Firewall

```
TgtProcName = "netsh.exe" AND TgtProcCmdLine ContainsCIS "add rule" AND TgtProcCmd
```

Atomic #6 - Allow Executable Through Defender Firewall

```
TgtProcName = "netsh.exe" AND TgtProcCmdLine ContainsCIS "add rule" AND TgtProcCmd
```

T1562.001 Disable or Modify Tools

Atomics: T1562.001

Atomic #1 - Disable Syslog

```
TgtProcName In Contains ("service","chkconfig","systemctl") AND TgtProcCmdLine In (
```

Atomic #9 AND #10 - Disable Sysmon

```
(TgtProcName = "fltmc.exe" AND TgtProcCmdLine ContainsCIS "unload SysmonDrv") OR (
```

Atomic #11 - AMSI Bypass - AMSI InitFailed

```
TgtProcCmdLine ContainsCIS "[Ref].Assembly.GetType('System.Management.Automation.A
```

Atomic #12 - AMSI Bypass - Remove AMSI Provider Reg Key

```
RegistryPath ContainsCIS "\Microsoft\AMSI\Providers" AND EventType In ("Registry K
```

Atomic #17 - Disable Microsoft Office Security Features

```
(RegistryKeyPath ContainsCIS "Excel\Security" OR RegistryKeyPath ContainsCIS "Excel
```

T1564.001 Hidden Files and Directories

Atomics: T1564.001

T1564.003 Hidden Window

Atomics: T1564.003

T1070 Indicator Removal on Host

Atomics: T1070

T1202 Indirect Command Execution

Atomics: T1202

T1553.004 Install Root Certificate

Atomics: T1553.004

T1218.004 InstallUtil

Atomics: T1218.004

T1127.001 MSBuild

Atomics: T1127.001

T1112 Modify Registry

Atomics: T1112

T1218.005 Mshta

Atomics: T1218.005

SentinelOne happens to be pretty good at detecting MSHTA attacks, and *IndicatorName* = "*SuspiciousScript*" specifically picks out these javascript based attacks very well. The below query will detect mshta.exe spawning processes as well as URLs for remote payloads to be loaded by mshta.

```
(SrcProcName = "mshta.exe" and EventType = "Open Remote Process Handle") OR (SrcPr
```

T1218.007 Msiexec

Atomics: T1218.007

The below query will accurately detect execution of remote msi files by msiexec.exe. The second half of the query aims to detect processes spawned by msi files instead of dll files in the CommandLine (as that is very noisy) and may return a bit of noise within for the CrossProcess Object as some auto-update processes may be collected by this query.

```
( SrcProcName = "msiexec.exe" AND SrcProcCmdLine RegExp "https?:\/\/(www\.)?[-a-zA
```

T1564.004 NTFS File Attributes

Atomics: T1564.004

T1070.005 Network Share Connection Removal

Atomics: T1070.005

T1027 Obfuscated Files or Information

Atomics: T1027

T1218.008 Odbcconf

Atomics: T1218.008

T1134.004 Parent PID Spoofing

Atomics: T1134.004

Detects parent PID spoofing through Cross Process indicators (SrcProcParentName limits scope heavily) as well as detecting the use of PPID-Spoof powershell script through Command Scripts indicators. Update the `TgtProcName` list to filter noise.

```
(TgtProcRelation = "not_in_storyline" AND EventType = "Open Remote Process Handle"
```

T1550.002 Pass the Hash

Atomics: T1550.002

T1550.003 Pass the Ticket

Atomics: T1550.003

T1556.002 Password Filter DLL

Atomics: T1556.002

T1574.009 Unquoted Service Path for program.exe

Atomics: T1574.009

Detects creation or modification of the file at `C:\program.exe` for exploiting unquoted services paths of Program Files folder.

```
(FileFullName = "C:\program.exe" AND EventType In ("File Creation","File Modificat
```

T1055.012 Process Hollowing

Atomics: T1055.012

Detect Process Hollowing using the Start-Hollow powershell script, through CommandLine and CommandScript indicators.

The `IndicatorCategory = "Injection"` has a lot of noise, but in the future a combination of `EventType = "Duplicate Process Handle" AND TgtProcRelation = "storyline_child"` joined with some `ChildProcCount` or `CrossProcCount` > 0 may help filter the noise.

```
--- Detect Start-Hollow.ps1 by command or content
(SrcProcCmdScript ContainsCIS "Start-Hollow" AND SrcProcCmdScript ContainsCIS "[Ho
```

T1055 Process Injection

Atomics: T1055

Detects Process Injection through execution of MavInject, filtering out noisy/expected activity. `SrcProcParentName` filter narrows Cross Process items to HQ results.

```
(TgtProcName = "mavinject.exe" AND TgtProcCmdLine ContainsCIS "/injectrunning") ANI
```

T1218.009 PubPrn

Atomics: T1218.009

T1218.009 Regsvcs/Regasm

Atomics: T1218.009

T1218.010 Regsvr32

Atomics: T1218.010

T1036.003 Rename System Utilities

Atomics: T1036.003

T1207 Rogue Domain Controller

Atomics: T1207

T1014 Rootkit

Atomics: T1014

T1218.011 Rundll32

Atomics: T1218.011

T1574.010 Services File Permissions Weakness

Atomics: T1574.010

T1574.011 Services Registry Permissions Weakness

Atomics: T1574.011

T1218 Signed Binary Proxy Execution

Atomics: T1218

T1216 Signed Script Proxy Execution

Atomics: T1216

T1070.006 Timestomp

Atomics: T1070.006

T1222.001 Windows File and Directory Permissions Modification

Atomics: T1222.001

T1220 XSL Script Processing

Atomics: T1220