



**AND THEN THERE WERE NONE**  
**(MORE FALSE POSITIVES)**

**DAN BANKER**  
**THREAT RESPONSE TEAM LEAD**  
**MOTOROLA SOLUTIONS**

*Belinda Fewings © 2016*

**dan@localhost~\$whoami**

dan@localhost~\$list jobs

Secureworks, Motorola Solutions

dan@localhost~\$list certs

GCIA, GCIH, GCFE, GNFA, GWAPT

dan@localhost~\$list hobbies

Guitar, Metal, iRacing

dan@localhost~\$echo "Don't try to race anyone from  
Finland. You'll lose."

# A TALE OF TWO MOTOROLAS



YES



# WE'RE NOT ROCK STARS

- Rock Stars:
  - Emotionally stunted
  - Bad with money and people
  - Substance abusers
  - Die young
- They shouldn't be role models for IT



# EDR is Grate

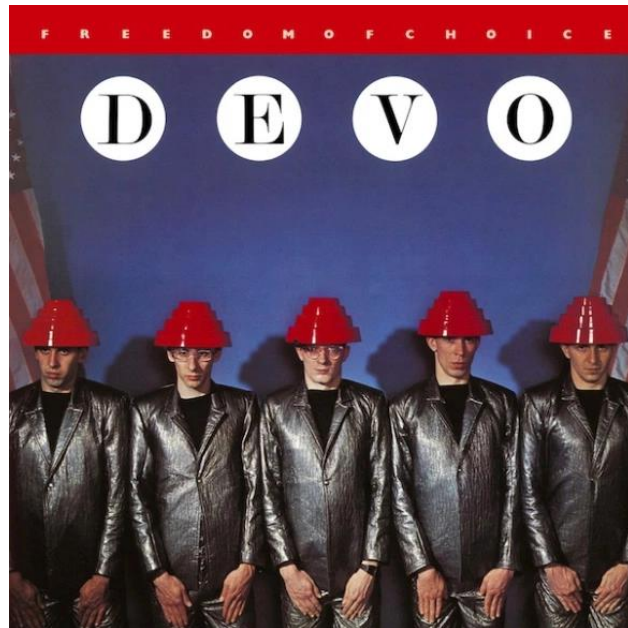
- Telemetry, telemetry, telemetry!
- Great threat hunting tool
- More Visibility - program starts/stops, file mods, registry mods, network connections, cross-process, etc.
- Instant Remediation - quarantine hosts, kill rogue processes, grab memory





# FREEDOM OF CHOICE

- All examples in VMWare Carbon Black Enterprise EDR syntax
  - Formerly CB ThreatHunter
  - Similar to CB Response
- Theory should work in your EDR of choice
  - YMMV



# SEARCHING

Searching for answers to my problems  
like

- <field>:<value>
- process\_name:cmd.exe
- Chain statements with logic
  - AND, OR, NOT
- Whitespace between statements = logical AND
- process\_name:cmd.exe netconn\_count:1
- process\_name:cmd.exe NOT process\_name:C:\\Windows\\system32\\cmd.exe



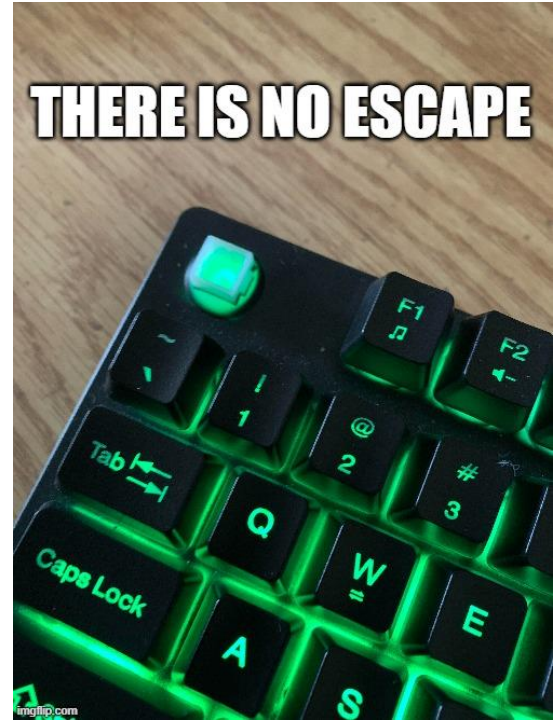
# ESCAPING

Quick note: escape EVERYTHING in CBTH

C:\Program Files (x86)\Kodak\AiO\StatusMonitor\EKPrinterSDK.exe

Becomes

process\_name:c:\\program\\ files\\ \\(x86\\)\\kodak\\aio\\statusmonitor\\ekprintersdk.exe





# TOKENIZATION

Process path: C:\Users\grateuser\AppData\LegitProgram\legitprogram.exe

Find it with:

- process\_name:"C:\Users\grateuser\AppData\LegitProgram\legitprogram.exe"
- process\_name:exe (along with every other exe in your network)
- process\_name:legitprogram.exe
- process\_name:\\legitprogram\\legitprogram.exe
- process\_name:\\AppData\\legitprogram\\\* (finds anything executing from that folder)

# NOW WITH AMSI!

- New for Carbon Black Threat Hunter, Crowdstrike
- Fileless script loads (no obfuscation)
  - Powershell, VBA, etc.
- We can look for command line strings, length, etc.
- Lots of FPs
  - (fileless\_scriptload\_cmdline:\*RunOnce\* OR fileless\_scriptload\_cmdline:\*Run\* OR fileless\_scriptload\_cmdline:\*Shell\Folders\*)

# WATCH THE SPLATS!

### Formatted PowerShell Script

```
1 \r\n#requires -version 3.0\r\n\r\ntry { Microsoft.PowerShell.Core\Set-StrictMode -Off } catch { }\r\n$MyInvocation.MyCommand.ScriptBlock.Module\r\n\r\n$script:ClassName = 'ROOT/StandardCimv2/MSFT_NetI\r\n'1.0.0'\r\n$script:ModuleVersion = '1.0.0'\r\n$script:ObjectModelWrapper =\r\n[Microsoft.PowerShell.Cmdletization.Cim.CimCmdletAdapter]\r\n\r\n$script:PrivateData =\r\n[System.Collections.Generic.Dictionary[string,string]]::new()\r\n\r\nMicrosoft.PowerShell.Core\Expo\r\n__cmdletization_BindCommonParameters\r\n\r\n\r\nparam(\r\n\r\n    $__cmdletization_objectModelWrap\r\n\r\n\r\n    if ($myPSBoundParameters.ContainsKey('CimSession')) { \r\n\r\n        $__cmdletization_objectModelWrapper.PSObject.Properties['CimSession'].Value = $myPSBoundParameters[\r\n        if ($myPSBoundParameters.ContainsKey('ThrottleLimit')) { \r\n\r\n            $__cmdletization_objectMod\r\n            $myPSBoundParameters['ThrottleLimit'] \r\n\r\n            }\r\n\r\n            \r\n\r\n            if ($myP\r\n            $__cmdletization_objectModelWrapper.PSObject.Properties['AsJob'].Value = $myPSBoundParameters['AsJo\r\n\r\n\r\n\r\nfunction Get-NetIPInterface\r\n\r\n\r\n    [CmdletBinding(DefaultParameterSetName='ByName', Po\r\n    [OutputType([Microsoft.Management.Infrastructure.CimInstance])]\r\n    [OutputType('Microsoft.Managemen\r\n    nterface')]\r\n    \r\n    param(\r\n\r\n        \r\n\r\n        [Parameter(ParameterSetName='ByName', ValueFromPipel\r\n        [ValidateNotNull()]\r\n        [uint32[]]\r\n        ${InterfaceIndex},\r\n\r\n\r\n        [Parameter(ParameterSetN\r\n        ValueFromPipelineByPropertyName=$true)]\r\n        [Alias('ifAlias')]\r\n        [ValidateNotNull()]\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [Microsoft.PowerShell.Cmdletization.GeneratedTypes.NetIPInterface.AddressFamily[]]\r\n        ${Address\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [Microsoft.PowerShell.Cmdl\r\n        ${Forwarding},\r\n\r\n\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [Microsoft.PowerShell.Cmdletization.GeneratedTypes.NetIPInterface.ClampMss[]]\r\n        ${ClampMss},\r\n        [ValidateNotNull()]\r\n        [Microsoft.PowerShell.Cmdletization.GeneratedTypes.NetIPInterface.Advert\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [uint32[]]\r\n        ${NlMtuB\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [uint32[]]\r\n        ${Interf\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [ValidateNotNull()]\r\n        [Microsoft.PowerShell.Cmdletization.GeneratedTypes.NetIPInterface.NeighborUnreachabilityDetection[\r\n        [Parameter(ParameterSetName='ByName')]\r\n        [Alias('BaseReachableTime')]\r\n        [ValidateNotNull(
```

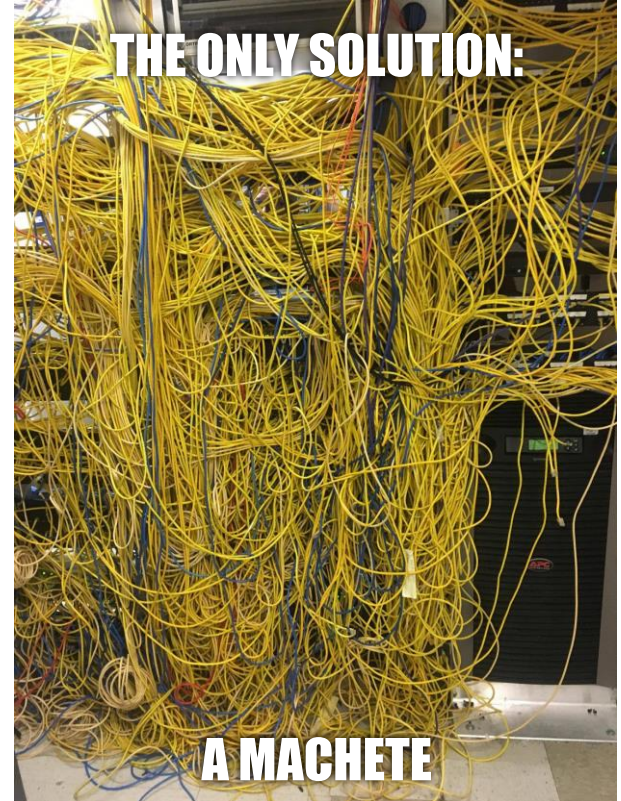
# THE PROBLEM - DEVELOPERS

- “This web app needs to run as root”
- “I need to finish this by Friday”
- “I copied my code from Stack Overflow”
- “Why would you use the app like that? I didn’t design it that way”
- “My prod and dev passwords are the same”
- “My prod and dev DBs are the same”
- “I base64 encrypt the passwords”



# THE PROBLEM - SYSADMINS

- “I start every session with ‘sudo su -’”
- “I forgot about that server”
  - Or: “There’s no patch schedule for that”
- “I forgot I left TCP/22 open to the Internet”
- “I copied this script from Stack Overflow”
- “I use the same password for my standard and admin accounts”
- “I let Bob use my password just this once”
- “I turned logging down to the minimum to save disk space”



# **FILTER OUT PARENT PROCESSES**

## **Base64-encoded Powershell:**

((process\_name:powershell.exe OR process\_name:powershell\_ise.exe OR  
process\_name:pwsh.exe OR process\_name:sqlps.exe)  
process\_cmdline:\*FromBase64String\* NOT parent\_name:agent.worker.exe)

## **Net User Command:**

(process\_name:cmd.exe AND process\_cmdline:"net user") NOT  
parent\_name:gpscript.exe



# IT WORKS!!

INVESTIGATE

✓ (process\_name:powershell.exe OR process\_name:powershell\_ise.exe OR process\_name:powershell.exe)

**FILTERS** Clear ⋮ <<

— **Process (1)**

Search

...shell\v1.0\powershell.exe 100.0%

+ **Effective Reputation (1)**

+ **Process Hash (2)**

— **Device (6)**

Search

1vstsbuid02	51.2%
1vstsbuid03	30.3%
1vstsbuid04	6.0%
3deployer01	5.3%

**271 results**

**PROCESS** ▾

powershell.exe Watchlist Hit  
c:\windows\system32\wind  
dowspowershell\v1.0\po  
wershell.exe

powershell.exe Watchlist Hit  
c:\windows\system32\win  
dowspowershell\v1.0\po  
wershell.exe


powershell.exe Watchlist Hit  
c:\windows\system32\win  
dowspowershell\v1.0\po  
wershell.exe

powershell.exe Watchlist Hit

BEFORE

process\_name:sqlps.exe) process\_cmdline:\*FromBase64String\* NO

**DEVICE TIME** ▾ **PID** **USERNAME** **REGMODS** ▾ **FILEMO**



**No results found with the selected criteria.  
Try broadening your search.**

AFTER

# FILTER OUT COMMAND LINES

## Dir Command:

(process\_cmdline:dir) NOT (process\_name:erl.exe OR parent\_name:ccstudio.exe  
OR parent\_name:erl.exe OR parent\_name:g2mstart.exe OR  
process\_cmdline:quinta\_jenkins OR process\_cmdline:quinta\\winro OR  
process\_cmdline:winros\_9)

# FILTER OUT PATH IN COMMAND LINE

**Powershell w/ mshta parent:**

```
((process_name:powershell.exe OR process_name:powershell_ise.exe OR  
process_name:pwsh.exe) AND parent_name:mshta.exe NOT  
process_cmdline:Desktop\\Check4Update.ps1)
```

# FILTER OUT DELL SUPPORT AGENT

## Powershell Execution Policy Change:

```
((regmod_name:software\\Microsoft\\PowerShell\\1\\ShellIds\\Microsoft.PowerShell\\ExecutionPolicy NOT process_name:powershell* NOT  
process_name:del\\supportassistagent\\bin\\*))
```

# FILTER OUT SCANNERS

**SAM or System registry hives being saved:**

((process\_cmdline:reg OR process\_name:reg.exe) process\_cmdline:save  
(process\_cmdline:sam OR process\_cmdline:system) NOT  
process\_cmdline:rapid7)

**Filters out this command line:**

reg save HKLM\SYSTEM "C:\Program Files\Rapid7\Insight  
Agent\components\insight\_agent\common\ir\_agent\_tmp\agent.jobs.tem\_realtime\HKEY\_LO  
CAL\_MACHINE\_SYSTEM.hiv"

# FUN WITH FILELESS SCRIPTLOADS

## Powershell Download Cradle:

(fileless\_scriptload\_cmdline:.downloadfile OR  
fileless\_scriptload\_cmdline:.downloaddata OR  
fileless\_scriptload\_cmdline:.downloadstring) NOT  
(fileless\_scriptload\_cmdline:chocolatey.org\* OR  
fileless\_scriptload\_cmdline:"http://download.virtualbox.org" OR  
fileless\_scriptload\_cmdline:"https://github.com/JuliaLang/Pkg.jl" OR  
netconn\_ipv4:10.0.0.0/8)



# DAMMIT JULIA

Prolyly OK

May contain malicious content

```
[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]::Tls12; $webclient = (New-Object System.Net.WebClient); $webclient.UseDefaultCredentials = $true; $webclient.Proxy.Credentials = $webclient.Credentials; $webclient.Headers.Add("user-agent", "Pkg.jl (https://github.com/JuliaLang/Pkg.jl)"); $webclient.Headers.Add("Julia-Pkg-Protocol", "1.0"); $webclient.Headers.Add("Julia-Version", "1.5.2"); $webclient.Headers.Add("Julia-System", "x86_64-w64-mingw32-libgfortran5-cxx11"); $webclient.Headers.Add("Julia-CI-Variables", "APPVEYOR=n;CI=n;CI_SERVER=n;CIRCLECI=n;CONTINUOUS_INTEGRATION=n;GITHUB_ACTIONS=n;GITLAB_CI=n;JULIA_CI=n;JULIA_PKGEVAL=n;JULIA_REGISTRYCI_AUTOMERGE=n;TF_BUILD=n;TRAVIS=n"); $webclient.Headers.Add("Julia-Interactive", "true"); $webclient.DownloadFile("https://pkg.julialang.org/registry/23338594-aafe-5451-b93e-139f81909106/060949c67cecd4819167e83183344befaa4a0866", "C:\Users\egrr002\AppData\Local\Temp\jl_vquaFtcokM-download.gz")
```

## Powershell Download Cradle

### Script Insights

#### Key Indicators ⓘ

##### Other

ServicePointManager,  
SecurityProtocol,  
SecurityProtocolType, Tls12, New-Object, WebClient,  
UseDefaultCredentials, Credentials, Add, DownloadFile

#### Formatted PowerShell Script

```
1 [System.net.servicepointmanager]::securityprotocol = [System.net.securityprotocoltype]::tls12
2 $webclient = (New-Object System.net.webclient)
3 $webclient.usedefaultcredentials = $true
4 $webclient.proxy.credentials = $webclient.credentials
5 $webclient.headers.add("user-agent", "Pkg.jl (https://github.com/JuliaLang/Pkg.jl)")
6 $webclient.headers.add("julia-pkg-protocol", "1.0")
7 $webclient.headers.add("julia-version", "1.5.2")
8 $webclient.headers.add("julia-system", "x86_64-w64-mingw32-libgfortran5-cxx11")
9 $webclient.headers.add("julia-ci-variables",
    "APPVEYOR=n;CI=n;CI_SERVER=n;CIRCLECI=n;CONTINUOUS_INTEGRATION=n;GITHUB_ACTIONS=n;GITLAB_CI=n;JULIA_CI=n;JULIA_PKGEVAL=n;JULIA_REGISTRYCI_AUTOMERGE=n;TF_BUILD=n;TRAVIS=n")
10 $webclient.headers.add("julia-interactive", "true")
11 $webclient.downloadfile("https://pkg.julialang.org/registry/23338594-aafe-5451-b93e-139f81909106/060949c67cecd4819167e83183344befaa4a0866",
    "C:\Users\egrr002\AppData\Local\Temp\jl_vquaFtcokM-download.gz")
```

# MORE FUN WITH FILELESS SCRIPTLOADS

## Portable Executables in Memory:

(fileless\_scriptload\_cmdline:\*TVoA\* OR fileless\_scriptload\_cmdline:\*TVpB\* OR  
fileless\_scriptload\_cmdline:\*TVpQ\* OR fileless\_scriptload\_cmdline:\*TVqA\* OR  
fileless\_scriptload\_cmdline:\*TVqQ\* OR fileless\_scriptload\_cmdline:\*TVro\* NOT  
(fileless\_scriptload\_cmdline:\*tv\*\\*.txt))

Hint: think Base64

May contain malicious content

```
[System.IO.File]::WriteAllBytes("$env:windir\DtlDownloads\V  
('TVqQAAMAAAEAAAA//8AALgAAAAAAAAAAQAAAAAAAAAA  
GUuDQ0KJAAAAAAAAABQRQAATAEDAOLftlAAAAAAAAAAO,  
AAAAAAEAAAEAAAAAAAAABAAAAAAAAAAAAAAAAAMTxAQBPAA  
AAAAAAIAAACAAAAAAAAAAAAAAAAACCAAEEgAAAAAAAAAA  
AABALnJlbG9jAAAMAAAAACACAAACAAAA3gEAAAAAAAAAA
```

# MORE MORE FUN WITH FILELESS SCRIPTLOADS

**Bad:**

```
Set-NetTCPSetting -InitialCongestionWindow 4 -SettingName InternetCustom |  
out-file -encoding ASCII -filepath  
C:\WINDOWS\system32\config\systemprofile\AppData\Roaming\pow\psTVpQ0.txt
```

**Filter by adding:**

```
NOT (fileless_scriptload_cmdline:*tv*\*.txt))
```

# OR GIVE UP



- Gotta know when to fold 'em

# TOXIC FALSE POSITIVITY

- Test, test test
- Check for FP/FN
- Revisit your watchlists periodically
- Get red team to help you
- Modify/add/delete as necessary



# THANK YOU!

- Twitterz:
  - @cybershredder
- Insta:
  - @blackenedchicago
  - (Blackened - a tribute to Metallica)
- #saveourstages

