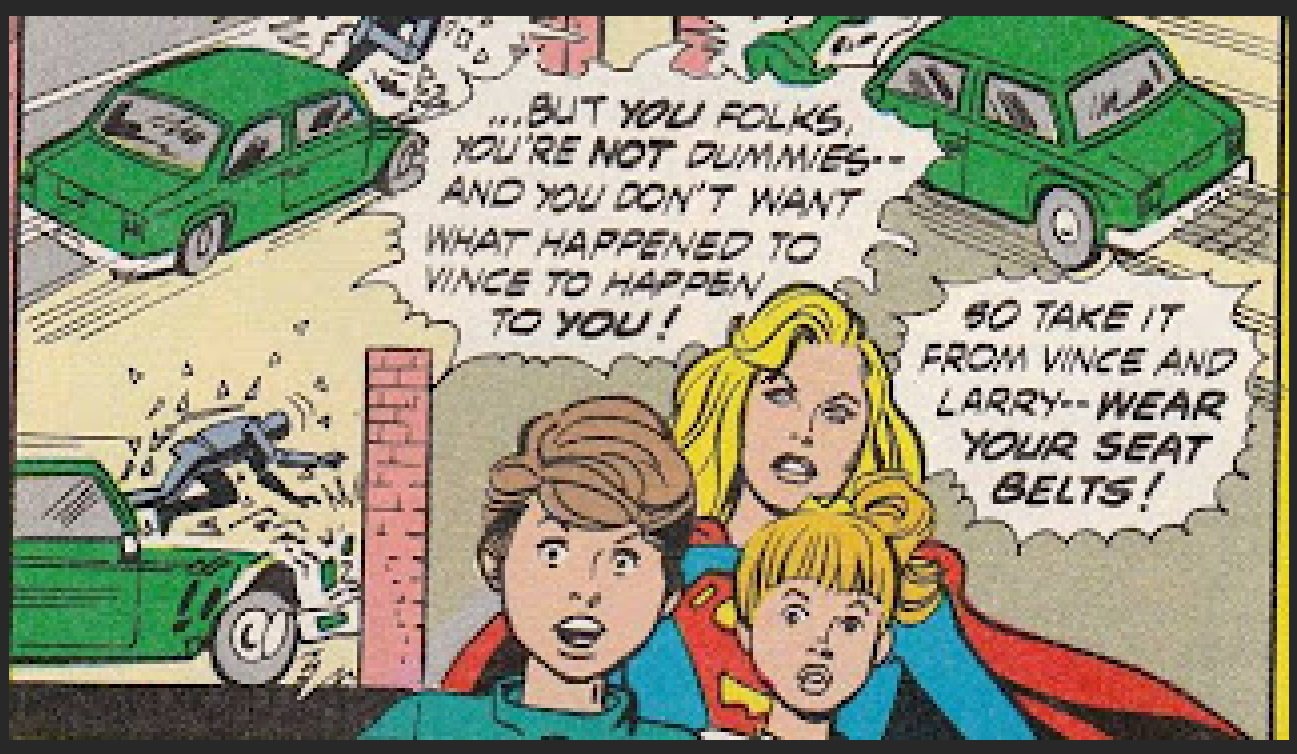
blue tangle

blue team dreams, splunk related detections and security insights. I poke around red team and threat actor tools and try to shed some light for cybersecurity wins.

Fastening the Seatbelt on.. Threat Hunting for Seatbelt

- August 26, 2022



Quick blog entry on detections for the Ghostpack discovery/reconnaissance tool Seatbelt.

This entry will focus on looking at command line parameters that can be caught even if the executable itself is renamed, if I have time we can delve into other event log artefacts another time.

From the Seatbelt github repo:

> *Seatbelt is a C# project that performs a number of security oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.*

So essentially what the tool does is retrieve local system information that might have security or safety implications.

In terms of commands that can be tacked on to Seatbelt there are a literal ton of options.

```
+ AMSIProviders          - Providers registered for AMSI
+ AntiVirus              - Registered antivirus (via WMI)
+ AppLocker              - AppLocker settings, if installed
  ARPTable               - Lists the current ARP table and adapter information (equivalent to
  AuditPolicies          - Enumerates classic and advanced audit policy settings
+ AuditPolicyRegistry    - Audit settings via the registry
+ AutoRuns               - Auto run executables/scripts/programs
  Certificates           - Finds user and machine personal certificate files
  CertificateThumbprints - Finds thumbprints for all certificate store certs on the systen
+ ChromiumBookmarks      - Parses any found Chrome/Edge/Brave/Opera bookmark files
+ ChromiumHistory        - Parses any found Chrome/Edge/Brave/Opera history files
+ ChromiumPresence       - Checks if interesting Chrome/Edge/Brave/Opera files exist
+ CloudCredentials       - AWS/Google/Azure/Bluemix cloud credential files
+ CloudSyncProviders     - All configured Office 365 endpoints (tenants and teamsites) which a
  CredEnum               - Enumerates the current user's saved credentials using CredEnumerate
+ CredGuard              - CredentialGuard configuration
  dir                    - Lists files/folders. By default, lists users' downloads, documents,
+ DNSCache               - DNS cache entries (via WMI)
+ DotNet                 - DotNet versions
+ DpapiMasterKeys        - List DPAPI master keys
  Dsregcmd               - Return Tenant information - Replacement for Dsregcmd /status
  EnvironmentPath        - Current environment %PATH% folders and SDDL information
+ EnvironmentVariables   - Current environment variables
+ ExplicitLogonEvents    - Explicit Logon events (Event ID 4648) from the security event log.
  ExplorerMRUs           - Explorer most recently used files (last 7 days, argument == last X
+ ExplorerRunCommands    - Recent Explorer "run" commands
  FileInfo               - Information about a file (version information, timestamps, basic PE
+ FileZilla              - FileZilla configuration files
+ FirefoxHistory         - Parses any found FireFox history files
+ FirefoxPresence        - Checks if interesting Firefox files exist
+ Hotfixes               - Installed hotfixes (via WMI)
  IdleTime               - Returns the number of seconds since the current user's last input.
+ IEFavorites            - Internet Explorer favorites
  IETabs                 - Open Internet Explorer tabs
+ IEUrls                 - Internet Explorer typed URLs (last 7 days, argument == last X days)
+ InstalledProducts      - Installed products via the registry
  InterestingFiles       - "Interesting" files matching various patterns in the user's folder.
```

But what we are going to focus on here are the command groups, which break the many, many available commands down into categories, so we have: `All, User, System, Slack, Chromium, Remote, Misc.`

The groups above are invoked like this, if you wanted to run all checks:

Seatbelt.exe -group=all

And so on and so forth. As well as specifying a group of checks to run an adversary is also going to want to specify an output file, as otherwise they will be left scrolling back through screenfulls of dense text in their Windows terminal.

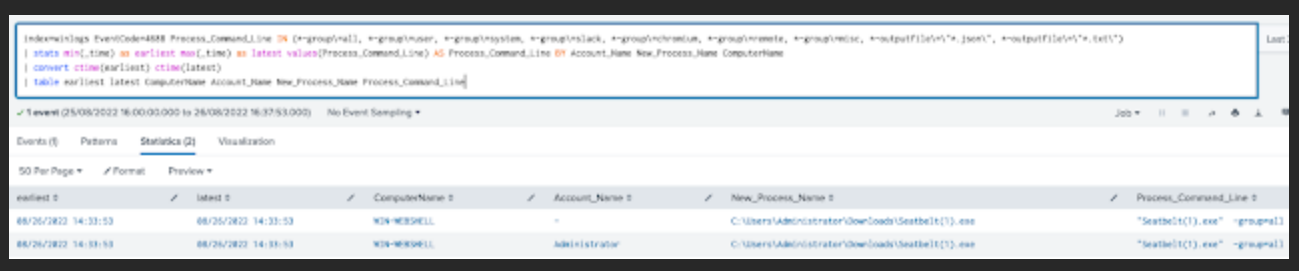
Let's combine both of these into an executable name agnostic detection for Splunk:

```
index=<winlogs-index> EventCode=4688 Process_Command_Line IN (*-group\=all, *-group\=user,
*-group\=system, *-group\=slack, *-group\=chromium, *-group\=remote, *-group\=misc, *-
outputfile\=\"*.json\", *-outputfile\=\"*.txt\")

| stats min(_time) as earliest max(_time) as latest values(Process_Command_Line) AS
Process_Command_Line BY Account_Name New_Process_Name ComputerName

| convert ctime(earliest) ctime(latest)

| table earliest latest ComputerName Account_Name New_Process_Name Process_Command_Line
```

Please note the escaped "=" in the SPL and the liberal sprinkling of necessary asterisks.

You may get some false positives depending on how many programs you run that use a similar command line syntax to what I've outlined above, testing will be required in your environment.

If it works you wind up with something like this:

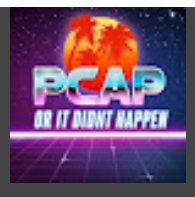


Happy hunting everyone.

ATT&CK command commandline cybersecurity Discovery elastic ghostpack Hacktool Invoke-Seatbelt lateral movement lockbit Lockbit Black ransomware seatbelt security splunk T1082 updates windows

---

Popular posts from this blog



the system") then you can take the barebones splunk SPL from below and make it work for you. So how ⋯

READ MORE »

Webshells automating reconnaissance gives us an easy detection win

- *July 22, 2020*

For those following along with ATT&CK this entry is about Server Software Component: Web Shell which is now a sub-technique of T1505, specifically it is T1505.003. If I can avoid combing through web access logs to find stuff like webshells I'll happily dodge it, having looked at the log artefacts left by a number of ⋯

READ MORE »

Powered by Blogger