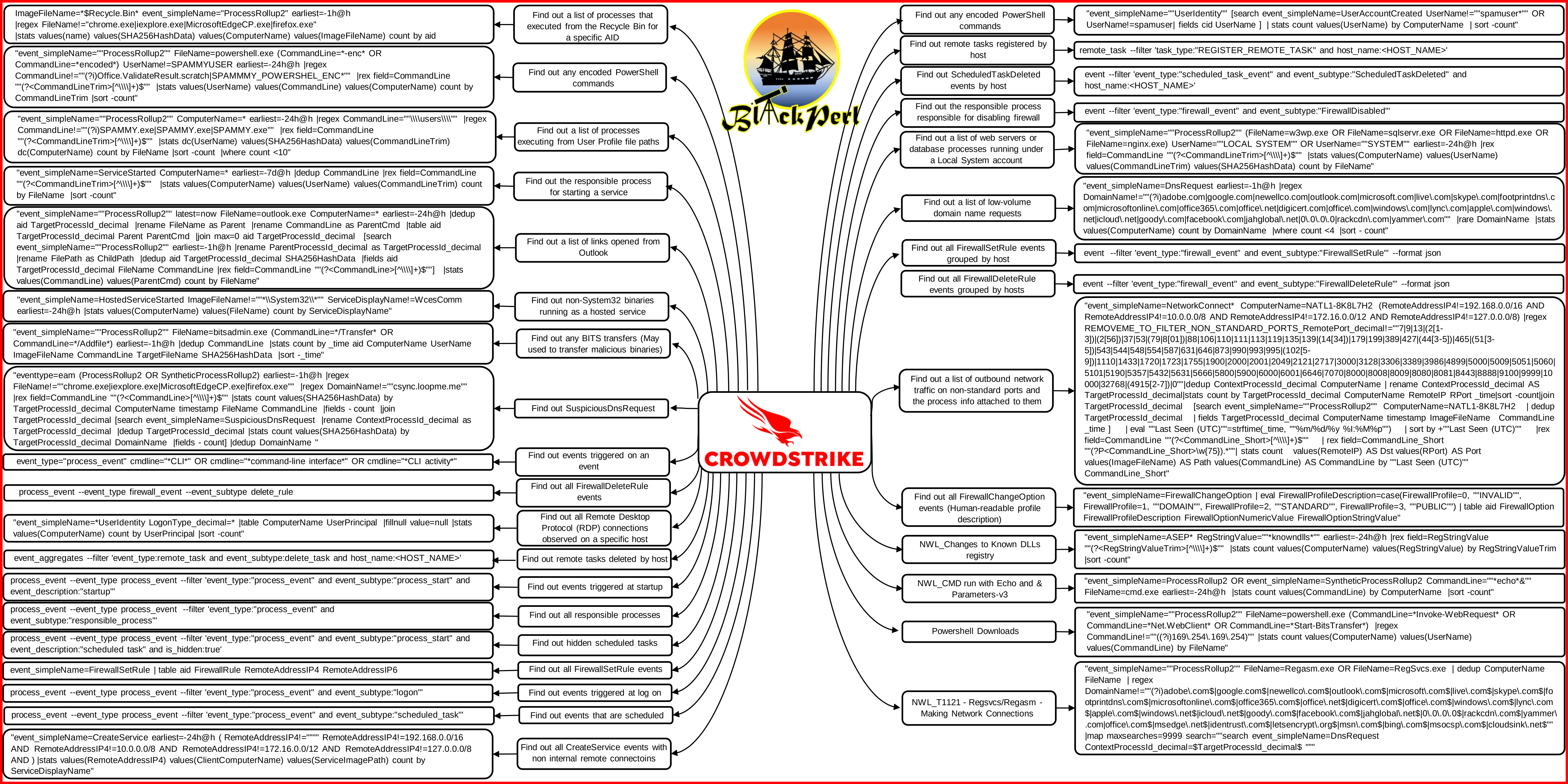


BlAckPerl

CROWDSTRIKE

Find out a list of processes that executed from the Recycle Bin for a specific AID

```
ImageFileName=*$Recycle.Bin* event_simpleName="ProcessRollup2" earliest=-1h@h
|regex FileName!="chrome.exe|iexplore.exe|MicrosoftEdgeCP.exe|firefox.exe"
|stats values(name) values(SHA256HashData) values(ComputerName) values(ImageFileName) count by aid
```

Find out any encoded PowerShell commands

```
"event_simpleName=""ProcessRollup2"" FileName=powershell.exe (CommandLine=*-enc* OR
CommandLine=*encoded*) UserName!=SPAMMYUSER earliest=-24h@h |regex
CommandLine!=""(?i)Office.ValidateResult.scratch|SPAMMMY_POWERSHEL_ENC*""  |rex field=CommandLine
""(?<CommandLineTrim>[^\\\\]+)$""  |stats values(UserName) values(CommandLine) values(ComputerName) count by
CommandLineTrim |sort -count"
```

Find out a list of processes executing from User Profile file paths

```
"event_simpleName=""ProcessRollup2"" ComputerName=* earliest=-24h@h |regex CommandLine=""\\\\users\\\\""  |regex
CommandLine!=""(?i)SPAMMY.exe|SPAMMY.exe|SPAMMY.exe""  |rex field=CommandLine
""(?<CommandLineTrim>[^\\\\]+)$""  |stats dc(UserName) values(SHA256HashData) values(CommandLineTrim)
dc(ComputerName) count by FileName |sort -count  |where count <10"
```

Find out the responsible process for starting a service

```
"event_simpleName=ServiceStarted ComputerName=* earliest=-7d@h |dedup CommandLine |rex field=CommandLine
""(?<CommandLineTrim>[^\\\\]+)$""   |stats values(ComputerName) values(UserName) values(CommandLineTrim) count
by FileName  |sort -count"
```

Find out a list of links opened from Outlook

```
"event_simpleName=""ProcessRollup2"" latest=now FileName=outlook.exe ComputerName=* earliest=-24h@h |dedup
aid TargetProcessId_decimal  |rename FileName as Parent  |rename CommandLine as ParentCmd  |table aid
TargetProcessId_decimal Parent ParentCmd  |join max=0 aid TargetProcessId_decimal   [search
event_simpleName=""ProcessRollup2"" earliest=-1h@h |rename ParentProcessId_decimal as TargetProcessId_decimal
|rename FilePath as ChildPath  |dedup aid TargetProcessId_decimal SHA256HashData  |fields aid
TargetProcessId_decimal FileName CommandLine |rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$""]   |stats
values(CommandLine) values(ParentCmd) count by FileName"
```

Find out non-System32 binaries running as a hosted service

```
"event_simpleName=HostedServiceStarted ImageFileName!=""*\\System32\\*"" ServiceDisplayName!=WcesComm
earliest=-24h@h |stats values(ComputerName) values(FileName) count by ServiceDisplayName"
```

Find out any BITS transfers (May used to transfer malicious binaries)

```
"event_simpleName=""ProcessRollup2"" FileName=bitsadmin.exe (CommandLine=*/Transfer* OR
CommandLine=*/Addfile*) earliest=-1h@h |dedup CommandLine  |stats count by _time aid ComputerName UserName
ImageFileName CommandLine TargetFileName SHA256HashData  |sort -_time"
```

Find out SuspiciousDnsRequest

```
"eventtype=eam (ProcessRollup2 OR SyntheticProcessRollup2) earliest=-1h@h |regex
FileName!=""chrome.exe|iexplore.exe|MicrosoftEdgeCP.exe|firefox.exe""  |regex DomainName!=""csync.loopme.me""
|rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$"" |stats count values(SHA256HashData) by
TargetProcessId_decimal ComputerName timestamp FileName CommandLine  |fields - count  |join
TargetProcessId_decimal [search event_simpleName=SuspiciousDnsRequest  |rename ContextProcessId_decimal as
TargetProcessId_decimal  |dedup TargetProcessId_decimal |stats count values(SHA256HashData) by
TargetProcessId_decimal DomainName  |fields - count] |dedup DomainName "
```

Find out events triggered on an event

```
event_type="process_event" cmdline="*CLI*" OR cmdline="*command-line interface*" OR cmdline="*CLI activity*"
```

Find out all FirewallDeleteRule events

```
process_event --event_type firewall_event --event_subtype delete_rule
```

Find out all Remote Desktop Protocol (RDP) connections observed on a specific host

```
"event_simpleName=*UserIdentity LogonType_decimal=* |table ComputerName UserPrincipal  |fillnull value=null |stats
values(ComputerName) count by UserPrincipal |sort -count"
```

Find out remote tasks deleted by host

```
event_aggregates --filter 'event_type:remote_task and event_subtype:delete_task and host_name:<HOST_NAME>'
```

Find out events triggered at startup

```
process_event --event_type process_event --filter 'event_type:"process_event" and event_subtype:"process_start" and
event_description:"startup"'
```

Find out all responsible processes

```
process_event --event_type process_event  --filter 'event_type:"process_event" and
event_subtype:"responsible_process"'
```

Find out hidden scheduled tasks

```
process_event --event_type process_event --filter 'event_type:"process_event" and event_subtype:"process_start" and
event_description:"scheduled task" and is_hidden:true'
```

Find out all FirewallSetRule events

```
event_simpleName=FirewallSetRule | table aid FirewallRule RemoteAddressIP4 RemoteAddressIP6
```

Find out events triggered at log on

```
process_event --event_type process_event --filter 'event_type:"process_event" and event_subtype:"logon"'
```

Find out events that are scheduled

```
process_event --event_type process_event --filter 'event_type:"process_event" and event_subtype:"scheduled_task"'
```

Find out all CreateService events with non internal remote connectoins

```
"event_simpleName=CreateService earliest=-24h@h ( RemoteAddressIP4!="""" RemoteAddressIP4!=192.168.0.0/16
AND  RemoteAddressIP4!=10.0.0.0/8 AND  RemoteAddressIP4!=172.16.0.0/12 AND  RemoteAddressIP4!=127.0.0.0/8
AND ) |stats values(RemoteAddressIP4) values(ClientComputerName) values(ServiceImagePath) count by
ServiceDisplayName"
```

Find out any encoded PowerShell commands

```
"event_simpleName=""UserIdentity"" [search event_simpleName=UserAccountCreated UserName!=""spamuser*"" OR
UserName!=spamuser|  fields cid UserName ]  | stats count values(UserName) by ComputerName  | sort -count"
```

Find out remote tasks registered by host

```
remote_task --filter 'task_type:"REGISTER_REMOTE_TASK" and host_name:<HOST_NAME>'
```

Find out ScheduledTaskDeleted events by host

```
event --filter 'event_type:"scheduled_task_event" and event_subtype:"ScheduledTaskDeleted" and
host_name:<HOST_NAME>'
```

Find out the responsible process responsible for disabling firewall

```
event --filter 'event_type:"firewall_event" and event_subtype:"FirewallDisabled"'
```

Find out a list of web servers or database processes running under a Local System account

```
"event_simpleName=""ProcessRollup2"" (FileName=w3wp.exe OR FileName=sqlservr.exe OR FileName=httpd.exe OR
FileName=nginx.exe) UserName=""LOCAL SYSTEM"" OR UserName=""SYSTEM"" earliest=-24h@h |rex
field=CommandLine ""(?<CommandLineTrim>[^\\\\]+)$""  |stats values(ComputerName) values(UserName)
values(CommandLineTrim) values(SHA256HashData) count by FileName"
```

Find out a list of low-volume domain name requests

```
"event_simpleName=DnsRequest earliest=-1h@h |regex
DomainName!=""(?i)adobe.com|google.com|newellco.com|outlook.com|microsoft.com|live\.com|skype\.com|footprintdns\.c
om|microsoftonline\.com|office365\.com|office\.net|digicert.com|office\.com|windows\.com|lync\.com|apple\.com|windows\.
net|icloud\.net|goody\.com|facebook\.com|jahglobal\.net|0\.0\.0\.0|rackcdn\.com|yammer\.com""  |rare DomainName  |stats
values(ComputerName) count by DomainName  |where count <4  |sort - count"
```

Find out all FirewallSetRule events grouped by host

```
event  --filter 'event_type:"firewall_event" and event_subtype:"FirewallSetRule"' --format json
```

Find out all FirewallDeleteRule events grouped by hosts

```
event --filter 'event_type:"firewall_event" and event_subtype:"FirewallDeleteRule"' --format json
```

Find out a list of outbound network traffic on non-standard ports and the process info attached to them

```
"event_simpleName=NetworkConnect*  ComputerName=NATL1-8K8L7H2  (RemoteAddressIP4!=192.168.0.0/16 AND
RemoteAddressIP4!=10.0.0.0/8 AND RemoteAddressIP4!=172.16.0.0/12 AND RemoteAddressIP4!=127.0.0.0/8) |regex
REMOVEME_TO_FILTER_NON_STANDARD_PORTS_RemotePort_decimal!=""7|9|13|(2[1-
3])|(2[56])|37|53|(79|8[01])|88|106|110|111|113|119|135|139|(14[34])|179|199|389|427|(44[3-5])|465|(51[3-
5])|543|544|548|554|587|631|646|873|990|993|995|(102[5-
9])|1110|1433|1720|1723|1755|1900|2000|2001|2049|2121|2717|3000|3128|3306|3389|3986|4899|5000|5009|5051|5060|
5101|5190|5357|5432|5631|5666|5800|5900|6000|6001|6646|7070|8000|8008|8009|8080|8081|8443|8888|9100|9999|10
000|32768|(4915[2-7])|0""|dedup ContextProcessId_decimal ComputerName | rename ContextProcessId_decimal AS
TargetProcessId_decimal|stats count by TargetProcessId_decimal ComputerName RemoteIP RPort _time|sort -count|join
TargetProcessId_decimal   [search event_simpleName=""ProcessRollup2""  ComputerName=NATL1-8K8L7H2   | dedup
TargetProcessId_decimal   | fields TargetProcessId_decimal ComputerName timestamp ImageFileName  CommandLine
_time ]    | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")    | sort by +""Last Seen (UTC)""    |rex
field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$""    | rex field=CommandLine_Short
""(?P<CommandLine_Short>\w{75}).*""| stats count   values(RemoteIP) AS Dst values(RPort) AS Port
values(ImageFileName) AS Path values(CommandLine) AS CommandLine by ""Last Seen (UTC)""
CommandLine_Short"
```

Find out all FirewallChangeOption events (Human-readable profile description)

```
"event_simpleName=FirewallChangeOption | eval FirewallProfileDescription=case(FirewallProfile=0, ""INVALID"",
FirewallProfile=1, ""DOMAIN"", FirewallProfile=2, ""STANDARD"", FirewallProfile=3, ""PUBLIC"") | table aid FirewallOption
FirewallProfileDescription  FirewallOptionNumericValue  FirewallOptionStringValue"
```

NWL_Changes to Known DLLs registry

```
"event_simpleName=ASEP* RegStringValue=""*knowndlls*"" earliest=-24h@h |rex field=RegStringValue
""(?<RegStringValueTrim>[^\\\\]+)$""  |stats count values(ComputerName) values(RegStringValue) by RegStringValueTrim
|sort -count"
```

NWL_CMD run with Echo and & Parameters-v3

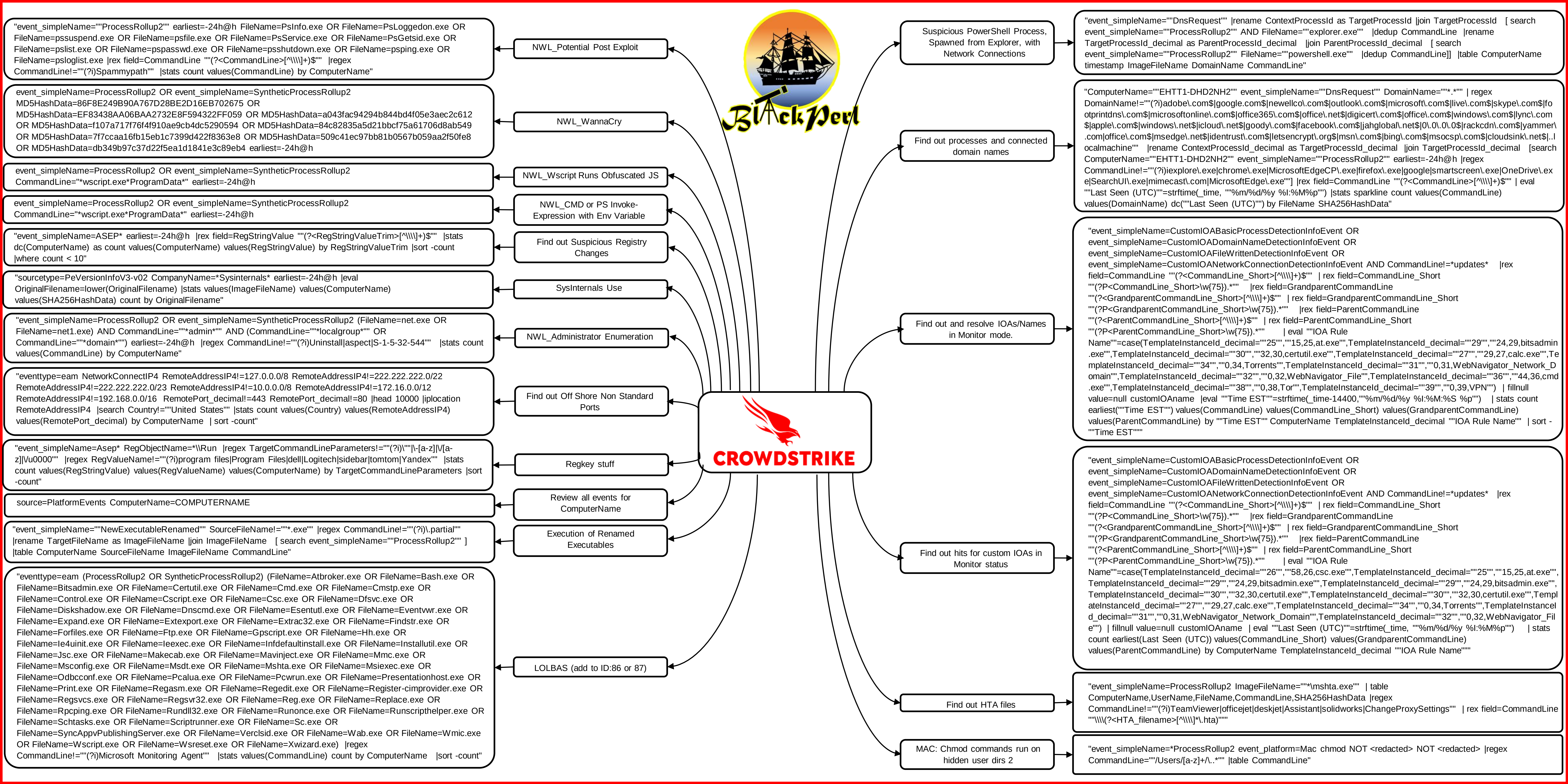
```
"event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2 CommandLine=""*echo*&""
FileName=cmd.exe earliest=-24h@h  |stats count values(CommandLine) by ComputerName  |sort -count"
```

Powershell Downloads

```
"event_simpleName=""ProcessRollup2"" FileName=powershell.exe (CommandLine=*Invoke-WebRequest* OR
CommandLine=*Net.WebClient* OR CommandLine=*Start-BitsTransfer*)  |regex
CommandLine!=""((?i)169\.254\.169\.254)"" |stats count values(ComputerName) values(UserName)
values(CommandLine) by FileName"
```

NWL_T1121 - Regsvcs/Regasm - Making Network Connections

```
"event_simpleName=""ProcessRollup2"" FileName=Regasm.exe OR FileName=RegSvcs.exe  | dedup ComputerName
FileName  | regex
DomainName!=""(?i)adobe\.com$|google.com$|newellco\.com$|outlook\.com$|microsoft\.com$|live\.com$|skype\.com$|fo
otprintdns\.com$|microsoftonline\.com$|office365\.com$|office\.net$|digicert\.com$|office\.com$|windows\.com$|lync\.com
$|apple\.com$|windows\.net$|icloud\.net$|goody\.com$|facebook\.com$|jahglobal\.net$|0\.0\.0\.0$|rackcdn\.com$|yammer\
.com|office\.com$|msedge\.net$|identrust\.com$|letsencrypt\.org$|msn\.com$|bing\.com$|msocsp\.com$|cloudsink\.net$""
|map maxsearches=9999 search=""search event_simpleName=DnsRequest
ContextProcessId_decimal=$TargetProcessId_decimal$ """
```



BlAckPerl

CROWDSTRIKE

NWL_Potential Post Exploit

```
"event_simpleName=""ProcessRollup2"" earliest=-24h@h FileName=PsInfo.exe OR FileName=PsLoggedon.exe OR
FileName=pssuspend.exe OR FileName=psfile.exe OR FileName=PsService.exe OR FileName=PsGetsid.exe OR
FileName=pslist.exe OR FileName=pspasswd.exe OR FileName=psshutdown.exe OR FileName=psping.exe OR
FileName=psloglist.exe |rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$""  |regex
CommandLine!=""(?i)Spammypath""  |stats count values(CommandLine) by ComputerName"
```

NWL_WannaCry

```
event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2
MD5HashData=86F8E249B90A767D28BE2D16EB702675 OR
MD5HashData=EF83438AA06BAA2732E8F594322FF059 OR MD5HashData=a043fac94294b844bd4f05e3aec2c612
OR MD5HashData=f107a717f76f4f910ae9cb4dc5290594 OR MD5HashData=84c82835a5d21bbcf75a61706d8ab549
OR MD5HashData=7f7ccaa16fb15eb1c7399d422f8363e8 OR MD5HashData=509c41ec97bb81b0567b059aa2f50fe8
OR MD5HashData=db349b97c37d22f5ea1d1841e3c89eb4 earliest=-24h@h
```

NWL_Wscript Runs Obfuscated JS

```
event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2
CommandLine="*wscript.exe*ProgramData*" earliest=-24h@h
```

NWL_CMD or PS Invoke-Expression with Env Variable

```
event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2
CommandLine="*wscript.exe*ProgramData*" earliest=-24h@h
```

Find out Suspicious Registry Changes

```
"event_simpleName=ASEP* earliest=-24h@h  |rex field=RegStringValue ""(?<RegStringValueTrim>[^\\\\]+)$""  |stats
dc(ComputerName) as count values(ComputerName) values(RegStringValue) by RegStringValueTrim |sort -count
|where count < 10"
```

SysInternals Use

```
"sourcetype=PeVersionInfoV3-v02 CompanyName=*Sysinternals* earliest=-24h@h |eval
OriginalFilename=lower(OriginalFilename) |stats values(ImageFileName) values(ComputerName)
values(SHA256HashData) count by OriginalFilename"
```

NWL_Administrator Enumeration

```
"event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2 (FileName=net.exe OR
FileName=net1.exe) AND CommandLine=""*admin*"" AND (CommandLine=""*localgroup*"" OR
CommandLine=""*domain*"") earliest=-24h@h  |regex CommandLine!=""(?i)Uninstall|aspect|S-1-5-32-544""   |stats count
values(CommandLine) by ComputerName"
```

Find out Off Shore Non Standard Ports

```
"eventtype=eam NetworkConnectIP4 RemoteAddressIP4!=127.0.0.0/8 RemoteAddressIP4!=222.222.222.0/22
RemoteAddressIP4!=222.222.222.0/23 RemoteAddressIP4!=10.0.0.0/8 RemoteAddressIP4!=172.16.0.0/12
RemoteAddressIP4!=192.168.0.0/16  RemotePort_decimal!=443 RemotePort_decimal!=80 |head 10000 |iplocation
RemoteAddressIP4  |search Country!=""United States"" |stats count values(Country) values(RemoteAddressIP4)
values(RemotePort_decimal) by ComputerName  | sort -count"
```

Regkey stuff

```
"event_simpleName=Asep* RegObjectName=*\\Run  |regex TargetCommandLineParameters!=""(?i)\""|\-[a-z]|\/[a-
z]|\/u0000""  |regex RegValueName!=""(?i)program files|Program Files|dell|Logitech|sidebar|tomtom|Yandex""  |stats
count values(RegStringValue) values(RegValueName) values(ComputerName) by TargetCommandLineParameters |sort
-count"
```

Review all events for ComputerName

```
source=PlatformEvents ComputerName=COMPUTERNAME
```

Execution of Renamed Executables

```
"event_simpleName=""NewExecutableRenamed"" SourceFileName!=""*.exe"" |regex CommandLine!=""(?i)\.partial""
|rename TargetFileName as ImageFileName |join ImageFileName   [ search event_simpleName=""ProcessRollup2"" ]
|table ComputerName SourceFileName ImageFileName CommandLine"
```

LOLBAS (add to ID:86 or 87)

```
"eventtype=eam (ProcessRollup2 OR SyntheticProcessRollup2) (FileName=Atbroker.exe OR FileName=Bash.exe OR
FileName=Bitsadmin.exe OR FileName=Certutil.exe OR FileName=Cmd.exe OR FileName=Cmstp.exe OR
FileName=Control.exe OR FileName=Cscript.exe OR FileName=Csc.exe OR FileName=Dfsvc.exe OR
FileName=Diskshadow.exe OR FileName=Dnscmd.exe OR FileName=Esentutl.exe OR FileName=Eventvwr.exe OR
FileName=Expand.exe OR FileName=Extexport.exe OR FileName=Extrac32.exe OR FileName=Findstr.exe OR
FileName=Forfiles.exe OR FileName=Ftp.exe OR FileName=Gpscript.exe OR FileName=Hh.exe OR
FileName=Ie4uinit.exe OR FileName=Ieexec.exe OR FileName=Infdefaultinstall.exe OR FileName=Installutil.exe OR
FileName=Jsc.exe OR FileName=Makecab.exe OR FileName=Mavinject.exe OR FileName=Mmc.exe OR
FileName=Msconfig.exe OR FileName=Msdt.exe OR FileName=Mshta.exe OR FileName=Msiexec.exe OR
FileName=Odbcconf.exe OR FileName=Pcalua.exe OR FileName=Pcwrun.exe OR FileName=Presentationhost.exe OR
FileName=Print.exe OR FileName=Regasm.exe OR FileName=Regedit.exe OR FileName=Register-cimprovider.exe OR
FileName=Regsvcs.exe OR FileName=Regsvr32.exe OR FileName=Reg.exe OR FileName=Replace.exe OR
FileName=Rpcping.exe OR FileName=Rundll32.exe OR FileName=Runonce.exe OR FileName=Runscripthelper.exe OR
FileName=Schtasks.exe OR FileName=Scriptrunner.exe OR FileName=Sc.exe OR
FileName=SyncAppvPublishingServer.exe OR FileName=Verclsid.exe OR FileName=Wab.exe OR FileName=Wmic.exe
OR FileName=Wscript.exe OR FileName=Wsreset.exe OR FileName=Xwizard.exe)  |regex
CommandLine!=""(?i)Microsoft Monitoring Agent""   |stats values(CommandLine) count by ComputerName   |sort -count"
```

Suspicious PowerShell Process, Spawned from Explorer, with Network Connections

```
"event_simpleName=""DnsRequest"" |rename ContextProcessId as TargetProcessId |join TargetProcessId   [ search
event_simpleName=""ProcessRollup2"" AND FileName=""explorer.exe""   |dedup CommandLine  |rename
TargetProcessId_decimal as ParentProcessId_decimal   |join ParentProcessId_decimal   [ search
event_simpleName=""ProcessRollup2"" FileName=""powershell.exe""   |dedup CommandLine]]  |table ComputerName
timestamp ImageFileName DomainName CommandLine"
```

Find out processes and connected domain names

```
"ComputerName=""EHTT1-DHD2NH2"" event_simpleName=""DnsRequest"" DomainName=""*.*"" | regex
DomainName!=""(?i)adobe\.com$|google.com$|newellco\.com$|outlook\.com$|microsoft\.com$|live\.com$|skype\.com$|fo
otprintdns\.com$|microsoftonline\.com$|office365\.com$|office\.net$|digicert\.com$|office\.com$|windows\.com$|lync\.com
$|apple\.com$|windows\.net$|icloud\.net$|goody\.com$|facebook\.com$|jahglobal\.net$|0\.0\.0\.0$|rackcdn\.com$|yammer\
.com|office\.com$|msedge\.net$|identrust\.com$|letsencrypt\.org$|msn\.com$|bing\.com$|msocsp\.com$|cloudsink\.net$|..l
ocalmachine""  |rename ContextProcessId_decimal as TargetProcessId_decimal  |join TargetProcessId_decimal  [search
ComputerName=""EHTT1-DHD2NH2"" event_simpleName=""ProcessRollup2"" earliest=-24h@h |regex
CommandLine!=""(?i)iexplore\.exe|chrome\.exe|MicrosoftEdgeCP\.exe|firefox\.exe|google|smartscreen\.exe|OneDrive\.ex
e|SearchUI\.exe|mimecast\.com|MicrosoftEdge\.exe""] |rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$"" | eval
""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") |stats sparkline count values(CommandLine)
values(DomainName) dc(""Last Seen (UTC)"") by FileName SHA256HashData"
```

Find out and resolve IOAs/Names in Monitor mode.

```
"event_simpleName=CustomIOABasicProcessDetectionInfoEvent OR
event_simpleName=CustomIOADomainNameDetectionInfoEvent OR
event_simpleName=CustomIOAFileWrittenDetectionInfoEvent OR
event_simpleName=CustomIOANetworkConnectionDetectionInfoEvent AND CommandLine!=*updates*    |rex
field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$""  | rex field=CommandLine_Short
""(?P<CommandLine_Short>\w{75}).*""    |rex field=GrandparentCommandLine
""(?<GrandparentCommandLine_Short>[^\\\\]+)$""  | rex field=GrandparentCommandLine_Short
""(?P<GrandparentCommandLine_Short>\w{75}).*""    |rex field=ParentCommandLine
""(?<ParentCommandLine_Short>[^\\\\]+)$""  | rex field=ParentCommandLine_Short
""(?P<ParentCommandLine_Short>\w{75}).*""       | eval ""IOA Rule
Name""=case(TemplateInstanceId_decimal=""25"",""15,25,at.exe"",TemplateInstanceId_decimal=""29"",""24,29,bitsadmin
.exe"",TemplateInstanceId_decimal=""30"",""32,30,certutil.exe"",TemplateInstanceId_decimal=""27"",""29,27,calc.exe"",Te
mplateInstanceId_decimal=""34"",""0,34,Torrents"",TemplateInstanceId_decimal=""31"",""0,31,WebNavigator_Network_D
omain"",TemplateInstanceId_decimal=""32"",""0,32,WebNavigator_File"",TemplateInstanceId_decimal=""36"",""44,36,cmd
.exe"",TemplateInstanceId_decimal=""38"",""0,38,Tor"",TemplateInstanceId_decimal=""39"",""0,39,VPN"")  | fillnull
value=null customIOAname  |eval ""Time EST""=strftime(_time-14400,""%m/%d/%y %I:%M:%S %p"")    | stats count
earliest(""Time EST"") values(CommandLine) values(CommandLine_Short) values(GrandparentCommandLine)
values(ParentCommandLine) by ""Time EST"" ComputerName TemplateInstanceId_decimal ""IOA Rule Name""  | sort -
""Time EST"""
```

Find out hits for custom IOAs in Monitor status

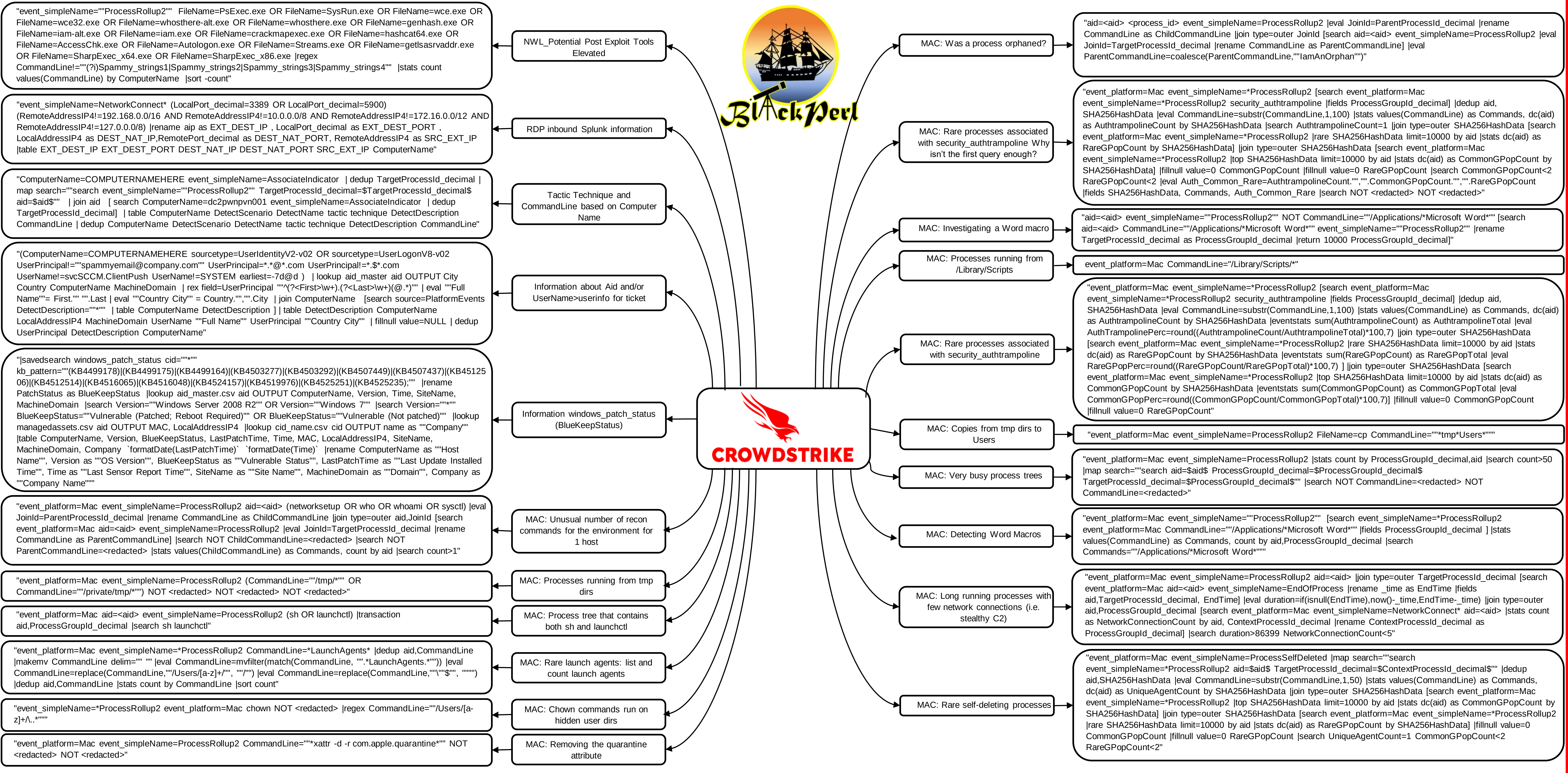
```
"event_simpleName=CustomIOABasicProcessDetectionInfoEvent OR
event_simpleName=CustomIOADomainNameDetectionInfoEvent OR
event_simpleName=CustomIOAFileWrittenDetectionInfoEvent OR
event_simpleName=CustomIOANetworkConnectionDetectionInfoEvent AND CommandLine!=*updates*   |rex
field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$""  | rex field=CommandLine_Short
""(?P<CommandLine_Short>\w{75}).*""    |rex field=GrandparentCommandLine
""(?<GrandparentCommandLine_Short>[^\\\\]+)$""  | rex field=GrandparentCommandLine_Short
""(?P<GrandparentCommandLine_Short>\w{75}).*""    |rex field=ParentCommandLine
""(?<ParentCommandLine_Short>[^\\\\]+)$""  | rex field=ParentCommandLine_Short
""(?P<ParentCommandLine_Short>\w{75}).*""       | eval ""IOA Rule
Name""=case(TemplateInstanceId_decimal=""26"",""58,26,csc.exe"",TemplateInstanceId_decimal=""25"",""15,25,at.exe"",
TemplateInstanceId_decimal=""29"",""24,29,bitsadmin.exe"",TemplateInstanceId_decimal=""29"",""24,29,bitsadmin.exe"",
TemplateInstanceId_decimal=""30"",""32,30,certutil.exe"",TemplateInstanceId_decimal=""30"",""32,30,certutil.exe"",Templ
ateInstanceId_decimal=""27"",""29,27,calc.exe"",TemplateInstanceId_decimal=""34"",""0,34,Torrents"",TemplateInstanceI
d_decimal=""31"",""0,31,WebNavigator_Network_Domain"",TemplateInstanceId_decimal=""32"",""0,32,WebNavigator_Fil
e"")  | fillnull value=null customIOAname  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")     | stats
count earliest(Last Seen (UTC)) values(CommandLine_Short) values(GrandparentCommandLine)
values(ParentCommandLine) by ComputerName TemplateInstanceId_decimal ""IOA Rule Name"""
```

Find out HTA files

```
"event_simpleName=ProcessRollup2 ImageFileName=""*\mshta.exe""  | table
ComputerName,UserName,FileName,CommandLine,SHA256HashData |regex
CommandLine!=""(?i)TeamViewer|officejet|deskjet|Assistant|solidworks|ChangeProxySettings""  | rex field=CommandLine
""\\\\(?<HTA_filename>[^\\\\]*\.hta)"""
```

MAC: Chmod commands run on hidden user dirs 2

```
"event_simpleName=*ProcessRollup2 event_platform=Mac chmod NOT <redacted> NOT <redacted> |regex
CommandLine=""/Users/[a-z]+/\..*"" |table CommandLine"
```



NWL_Potential Post Exploit Tools Elevated

```
"event_simpleName=""ProcessRollup2""  FileName=PsExec.exe OR FileName=SysRun.exe OR FileName=wce.exe OR
FileName=wce32.exe OR FileName=whosthere-alt.exe OR FileName=whosthere.exe OR FileName=genhash.exe OR
FileName=iam-alt.exe OR FileName=iam.exe OR FileName=crackmapexec.exe OR FileName=hashcat64.exe OR
FileName=AccessChk.exe OR FileName=Autologon.exe OR FileName=Streams.exe OR FileName=getlsasrvaddr.exe
OR FileName=SharpExec_x64.exe OR FileName=SharpExec_x86.exe |regex
CommandLine!=""(?i)Spammy_strings1|Spammy_strings2|Spammy_strings3|Spammy_strings4""  |stats count
values(CommandLine) by ComputerName  |sort -count"
```

RDP inbound Splunk information

```
"event_simpleName=NetworkConnect* (LocalPort_decimal=3389 OR LocalPort_decimal=5900)
(RemoteAddressIP4!=192.168.0.0/16 AND RemoteAddressIP4!=10.0.0.0/8 AND RemoteAddressIP4!=172.16.0.0/12 AND
RemoteAddressIP4!=127.0.0.0/8) |rename aip as EXT_DEST_IP , LocalPort_decimal as EXT_DEST_PORT ,
LocalAddressIP4 as DEST_NAT_IP,RemotePort_decimal as DEST_NAT_PORT, RemoteAddressIP4 as SRC_EXT_IP
|table EXT_DEST_IP EXT_DEST_PORT DEST_NAT_IP DEST_NAT_PORT SRC_EXT_IP ComputerName"
```

Tactic Technique and CommandLine based on Computer Name

```
"ComputerName=COMPUTERNAMEHERE event_simpleName=AssociateIndicator  | dedup TargetProcessId_decimal |
map search=""search event_simpleName=""ProcessRollup2""  TargetProcessId_decimal=$TargetProcessId_decimal$
aid=$aid$""   | join aid   [ search ComputerName=dc2pwnpvn001 event_simpleName=AssociateIndicator  | dedup
TargetProcessId_decimal]  | table ComputerName DetectScenario DetectName tactic technique DetectDescription
CommandLine | dedup ComputerName DetectScenario DetectName tactic technique DetectDescription CommandLine"
```

Information about Aid and/or UserName>userinfo for ticket

```
"(ComputerName=COMPUTERNAMEHERE sourcetype=UserIdentityV2-v02 OR sourcetype=UserLogonV8-v02
UserPrincipal!=""spammyemail@company.com"" UserPrincipal=*.*@*.com UserPrincipal!=*.$*.com
UserName!=svcSCCM.ClientPush UserName!=SYSTEM earliest=-7d@d )  | lookup aid_master aid OUTPUT City
Country ComputerName MachineDomain  | rex field=UserPrincipal ""^(?<First>\w+).(?<Last>\w+)(@.*)"" | eval ""Full
Name""= First."" "".Last | eval ""Country City"" = Country."","".City  | join ComputerName   [search source=PlatformEvents
DetectDescription=""*""  | table ComputerName DetectDescription ] | table DetectDescription ComputerName
LocalAddressIP4 MachineDomain UserName ""Full Name"" UserPrincipal ""Country City""  | fillnull value=NULL | dedup
UserPrincipal DetectDescription ComputerName"
```

Information windows_patch_status (BlueKeepStatus)

```
"|savedsearch windows_patch_status cid=""*""
kb_pattern=""(KB4499178)|(KB4499175)|(KB4499164)|(KB4503277)|(KB4503292)|(KB4507449)|(KB4507437)|(KB45125
06)|(KB4512514)|(KB4516065)|(KB4516048)|(KB4524157)|(KB4519976)|(KB4525251)|(KB4525235);""  |rename
PatchStatus as BlueKeepStatus  |lookup aid_master.csv aid OUTPUT ComputerName, Version, Time, SiteName,
MachineDomain  |search Version=""Windows Server 2008 R2"" OR Version=""Windows 7""  |search Version=""*""
BlueKeepStatus=""Vulnerable (Patched; Reboot Required)"" OR BlueKeepStatus=""Vulnerable (Not patched)""  |lookup
managedassets.csv aid OUTPUT MAC, LocalAddressIP4  |lookup cid_name.csv cid OUTPUT name as ""Company""
|table ComputerName, Version, BlueKeepStatus, LastPatchTime, Time, MAC, LocalAddressIP4, SiteName,
MachineDomain, Company  `formatDate(LastPatchTime)`  `formatDate(Time)`  |rename ComputerName as ""Host
Name"", Version as ""OS Version"", BlueKeepStatus as ""Vulnerable Status"", LastPatchTime as ""Last Update Installed
Time"", Time as ""Last Sensor Report Time"", SiteName as ""Site Name"", MachineDomain as ""Domain"", Company as
""Company Name"""
```

MAC: Unusual number of recon commands for the environment for 1 host

```
"event_platform=Mac event_simpleName=ProcessRollup2 aid=<aid> (networksetup OR who OR whoami OR sysctl) |eval
JoinId=ParentProcessId_decimal |rename CommandLine as ChildCommandLine |join type=outer aid,JoinId [search
event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 |eval JoinId=TargetProcessId_decimal |rename
CommandLine as ParentCommandLine] |search NOT ChildCommandLine=<redacted> |search NOT
ParentCommandLine=<redacted> |stats values(ChildCommandLine) as Commands, count by aid |search count>1"
```

MAC: Processes running from tmp dirs

```
"event_platform=Mac event_simpleName=ProcessRollup2 (CommandLine=""/tmp/*"" OR
CommandLine=""/private/tmp/*"") NOT <redacted> NOT <redacted> NOT <redacted>"
```

MAC: Process tree that contains both sh and launchctl

```
"event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 (sh OR launchctl) |transaction
aid,ProcessGroupId_decimal |search sh launchctl"
```

MAC: Rare launch agents: list and count launch agents

```
"event_platform=Mac event_simpleName=*ProcessRollup2 CommandLine=*LaunchAgents* |dedup aid,CommandLine
|makemv CommandLine delim="" "" |eval CommandLine=mvfilter(match(CommandLine, "".*LaunchAgents.*"")) |eval
CommandLine=replace(CommandLine,""/Users/[a-z]+/"", ""/"") |eval CommandLine=replace(CommandLine,""\""$"", """")
|dedup aid,CommandLine |stats count by CommandLine |sort count"
```

MAC: Chown commands run on hidden user dirs

```
"event_simpleName=*ProcessRollup2 event_platform=Mac chown NOT <redacted> |regex CommandLine=""/Users/[a-
z]+/\..*"""
```

MAC: Removing the quarantine attribute

```
"event_platform=Mac event_simpleName=ProcessRollup2 CommandLine=""*xattr -d -r com.apple.quarantine*"" NOT
<redacted> NOT <redacted>"
```

MAC: Was a process orphaned?

```
"aid=<aid> <process_id> event_simpleName=ProcessRollup2 |eval JoinId=ParentProcessId_decimal |rename
CommandLine as ChildCommandLine |join type=outer JoinId [search aid=<aid> event_simpleName=ProcessRollup2 |eval
JoinId=TargetProcessId_decimal |rename CommandLine as ParentCommandLine] |eval
ParentCommandLine=coalesce(ParentCommandLine,""IamAnOrphan"")"
```

MAC: Rare processes associated with security_authtrampoline Why isn't the first query enough?

```
"event_platform=Mac event_simpleName=*ProcessRollup2 [search event_platform=Mac
event_simpleName=*ProcessRollup2 security_authtrampoline |fields ProcessGroupId_decimal] |dedup aid,
SHA256HashData |eval CommandLine=substr(CommandLine,1,100) |stats values(CommandLine) as Commands, dc(aid)
as AuthtrampolineCount by SHA256HashData |search AuthtrampolineCount=1 |join type=outer SHA256HashData [search
event_platform=Mac event_simpleName=*ProcessRollup2 |rare SHA256HashData limit=10000 by aid |stats dc(aid) as
RareGPopCount by SHA256HashData] |join type=outer SHA256HashData [search event_platform=Mac
event_simpleName=*ProcessRollup2 |top SHA256HashData limit=10000 by aid |stats dc(aid) as CommonGPopCount by
SHA256HashData] |fillnull value=0 CommonGPopCount |fillnull value=0 RareGPopCount |search CommonGPopCount<2
RareGPopCount<2 |eval Auth_Common_Rare=AuthtrampolineCount."","".CommonGPopCount."","".RareGPopCount
|fields SHA256HashData, Commands, Auth_Common_Rare |search NOT <redacted> NOT <redacted>"
```

MAC: Investigating a Word macro

```
"aid=<aid> event_simpleName=""ProcessRollup2"" NOT CommandLine=""/Applications/*Microsoft Word*"" [search
aid=<aid> CommandLine=""/Applications/*Microsoft Word*"" event_simpleName=""ProcessRollup2"" |rename
TargetProcessId_decimal as ProcessGroupId_decimal |return 10000 ProcessGroupId_decimal]"
```

MAC: Processes running from /Library/Scripts

```
event_platform=Mac CommandLine="/Library/Scripts/*"
```

MAC: Rare processes associated with security_authtrampoline

```
"event_platform=Mac event_simpleName=*ProcessRollup2 [search event_platform=Mac
event_simpleName=*ProcessRollup2 security_authtrampoline |fields ProcessGroupId_decimal] |dedup aid,
SHA256HashData |eval CommandLine=substr(CommandLine,1,100) |stats values(CommandLine) as Commands, dc(aid)
as AuthtrampolineCount by SHA256HashData |eventstats sum(AuthtrampolineCount) as AuthtrampolineTotal |eval
AuthTrampolinePerc=round((AuthtrampolineCount/AuthtrampolineTotal)*100,7) |join type=outer SHA256HashData
[search event_platform=Mac event_simpleName=*ProcessRollup2 |rare SHA256HashData limit=10000 by aid |stats
dc(aid) as RareGPopCount by SHA256HashData |eventstats sum(RareGPopCount) as RareGPopTotal |eval
RareGPopPerc=round((RareGPopCount/RareGPopTotal)*100,7) ] |join type=outer SHA256HashData [search
event_platform=Mac event_simpleName=*ProcessRollup2 |top SHA256HashData limit=10000 by aid |stats dc(aid) as
CommonGPopCount by SHA256HashData |eventstats sum(CommonGPopCount) as CommonGPopTotal |eval
CommonGPopPerc=round((CommonGPopCount/CommonGPopTotal)*100,7)] |fillnull value=0 CommonGPopCount
|fillnull value=0 RareGPopCount"
```

MAC: Copies from tmp dirs to Users

```
"event_platform=Mac event_simpleName=ProcessRollup2 FileName=cp CommandLine=""*tmp*Users*"""
```

MAC: Very busy process trees

```
"event_platform=Mac event_simpleName=ProcessRollup2 |stats count by ProcessGroupId_decimal,aid |search count>50
|map search=""search aid=$aid$ ProcessGroupId_decimal=$ProcessGroupId_decimal$
TargetProcessId_decimal=$ProcessGroupId_decimal$"" |search NOT CommandLine=<redacted> NOT
CommandLine=<redacted>"
```

MAC: Detecting Word Macros

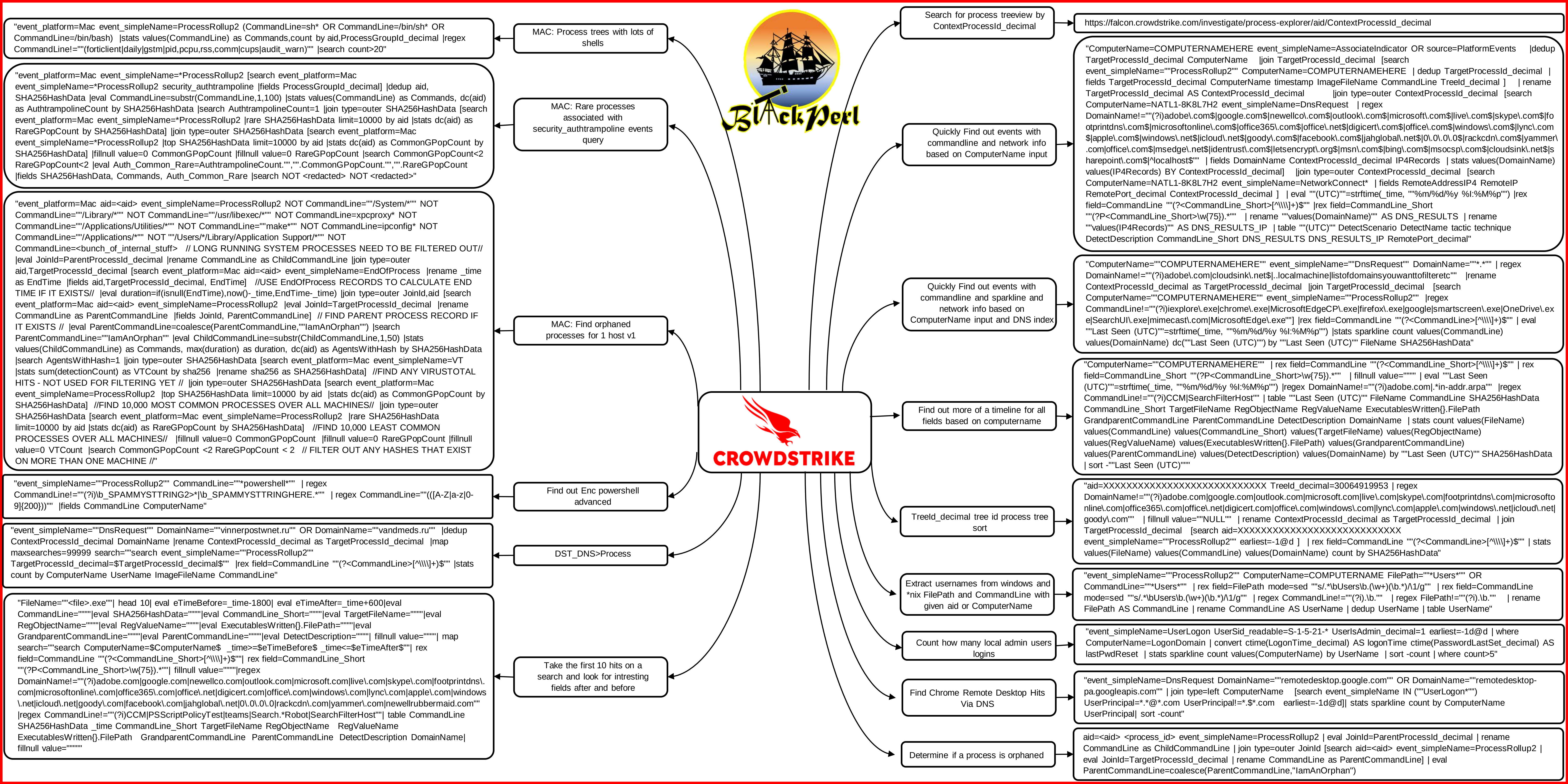
```
"event_platform=Mac event_simpleName=""ProcessRollup2""  [search event_simpleName=*ProcessRollup2
event_platform=Mac CommandLine=""/Applications/*Microsoft Word*"" |fields ProcessGroupId_decimal ] |stats
values(CommandLine) as Commands, count by aid,ProcessGroupId_decimal |search
Commands=""/Applications/*Microsoft Word*"""
```

MAC: Long running processes with few network connections (i.e. stealthy C2)

```
"event_platform=Mac event_simpleName=ProcessRollup2 aid=<aid> |join type=outer TargetProcessId_decimal [search
event_platform=Mac aid=<aid> event_simpleName=EndOfProcess |rename _time as EndTime |fields
aid,TargetProcessId_decimal, EndTime] |eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) |join type=outer
aid,ProcessGroupId_decimal [search event_platform=Mac event_simpleName=NetworkConnect* aid=<aid> |stats count
as NetworkConnectionCount by aid, ContextProcessId_decimal |rename ContextProcessId_decimal as
ProcessGroupId_decimal] |search duration>86399 NetworkConnectionCount<5"
```

MAC: Rare self-deleting processes

```
"event_platform=Mac event_simpleName=ProcessSelfDeleted |map search=""search
event_simpleName=*ProcessRollup2 aid=$aid$ TargetProcessId_decimal=$ContextProcessId_decimal$"" |dedup
aid,SHA256HashData |eval CommandLine=substr(CommandLine,1,50) |stats values(CommandLine) as Commands,
dc(aid) as UniqueAgentCount by SHA256HashData |join type=outer SHA256HashData [search event_platform=Mac
event_simpleName=*ProcessRollup2 |top SHA256HashData limit=10000 by aid |stats dc(aid) as CommonGPopCount by
SHA256HashData] |join type=outer SHA256HashData [search event_platform=Mac event_simpleName=*ProcessRollup2
|rare SHA256HashData limit=10000 by aid |stats dc(aid) as RareGPopCount by SHA256HashData] |fillnull value=0
CommonGPopCount |fillnull value=0 RareGPopCount |search UniqueAgentCount=1 CommonGPopCount<2
RareGPopCount<2"
```



CROWDSTRIKE

MAC: Process trees with lots of shells

```
"event_platform=Mac event_simpleName=ProcessRollup2 (CommandLine=sh* OR CommandLine=/bin/sh* OR
CommandLine=/bin/bash)  |stats values(CommandLine) as Commands,count by aid,ProcessGroupId_decimal |regex
CommandLine!=""(forticlient|daily|gstm|pid,pcpu,rss,comm|cups|audit_warn)"" |search count>20"
```

MAC: Rare processes associated with security_authtrampoline events query

```
"event_platform=Mac event_simpleName=*ProcessRollup2 [search event_platform=Mac
event_simpleName=*ProcessRollup2 security_authtrampoline |fields ProcessGroupId_decimal] |dedup aid,
SHA256HashData |eval CommandLine=substr(CommandLine,1,100) |stats values(CommandLine) as Commands, dc(aid)
as AuthtrampolineCount by SHA256HashData |search AuthtrampolineCount=1 |join type=outer SHA256HashData [search
event_platform=Mac event_simpleName=*ProcessRollup2 |rare SHA256HashData limit=10000 by aid |stats dc(aid) as
RareGPopCount by SHA256HashData] |join type=outer SHA256HashData [search event_platform=Mac
event_simpleName=*ProcessRollup2 |top SHA256HashData limit=10000 by aid |stats dc(aid) as CommonGPopCount by
SHA256HashData] |fillnull value=0 CommonGPopCount |fillnull value=0 RareGPopCount |search CommonGPopCount<2
RareGPopCount<2 |eval Auth_Common_Rare=AuthtrampolineCount."","".CommonGPopCount."","".RareGPopCount
|fields SHA256HashData, Commands, Auth_Common_Rare |search NOT <redacted> NOT <redacted>"
```

MAC: Find orphaned processes for 1 host v1

```
"event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 NOT CommandLine=""/System/*"" NOT
CommandLine=""/Library/*"" NOT CommandLine=""/usr/libexec/*"" NOT CommandLine=xpcproxy* NOT
CommandLine=""/Applications/Utilities/*"" NOT CommandLine=""make*"" NOT CommandLine=ipconfig* NOT
CommandLine=""/Applications/*"" NOT ""/Users/*/Library/Application Support/*"" NOT
CommandLine=<bunch_of_internal_stuff>   // LONG RUNNING SYSTEM PROCESSES NEED TO BE FILTERED OUT//
|eval JoinId=ParentProcessId_decimal |rename CommandLine as ChildCommandLine |join type=outer
aid,TargetProcessId_decimal [search event_platform=Mac aid=<aid> event_simpleName=EndOfProcess  |rename _time
as EndTime  |fields aid,TargetProcessId_decimal, EndTime]   //USE EndOfProcess RECORDS TO CALCULATE END
TIME IF IT EXISTS//  |eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) |join type=outer JoinId,aid [search
event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2  |eval JoinId=TargetProcessId_decimal  |rename
CommandLine as ParentCommandLine  |fields JoinId, ParentCommandLine]  // FIND PARENT PROCESS RECORD IF
IT EXISTS //  |eval ParentCommandLine=coalesce(ParentCommandLine,""IamAnOrphan"") |search
ParentCommandLine=""IamAnOrphan"" |eval ChildCommandLine=substr(ChildCommandLine,1,50) |stats
values(ChildCommandLine) as Commands, max(duration) as duration, dc(aid) as AgentsWithHash by SHA256HashData
|search AgentsWithHash=1 |join type=outer SHA256HashData [search event_platform=Mac event_simpleName=VT
|stats sum(detectionCount) as VTCount by sha256  |rename sha256 as SHA256HashData]  //FIND ANY VIRUSTOTAL
HITS - NOT USED FOR FILTERING YET //  |join type=outer SHA256HashData [search event_platform=Mac
event_simpleName=ProcessRollup2  |top SHA256HashData limit=10000 by aid  |stats dc(aid) as CommonGPopCount by
SHA256HashData]  //FIND 10,000 MOST COMMON PROCESSES OVER ALL MACHINES//  |join type=outer
SHA256HashData [search event_platform=Mac event_simpleName=ProcessRollup2  |rare SHA256HashData
limit=10000 by aid |stats dc(aid) as RareGPopCount by SHA256HashData]   //FIND 10,000 LEAST COMMON
PROCESSES OVER ALL MACHINES//   |fillnull value=0 CommonGPopCount  |fillnull value=0 RareGPopCount |fillnull
value=0 VTCount  |search CommonGPopCount <2 RareGPopCount < 2   // FILTER OUT ANY HASHES THAT EXIST
ON MORE THAN ONE MACHINE //"
```

Find out Enc powershell advanced

```
"event_simpleName=""ProcessRollup2"" CommandLine=""*powershell*""  | regex
CommandLine!=""(?i)\b_SPAMMYSTTRING2>*|\b_SPAMMYSTTRINGHERE.*""  | regex CommandLine=""(([A-Z|a-z|0-
9]{200}))""  |fields CommandLine ComputerName"
```

DST_DNS>Process

```
"event_simpleName=""DnsRequest"" DomainName=""vinnerpostwnet.ru"" OR DomainName=""vandmeds.ru""  |dedup
ContextProcessId_decimal DomainName |rename ContextProcessId_decimal as TargetProcessId_decimal  |map
maxsearches=99999 search=""search event_simpleName=""ProcessRollup2""
TargetProcessId_decimal=$TargetProcessId_decimal$""  |rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$"" |stats
count by ComputerName UserName ImageFileName CommandLine"
```

Take the first 10 hits on a search and look for intresting fields after and before

```
"FileName=""<file>.exe""| head 10| eval eTimeBefore=_time-1800| eval eTimeAfter=_time+600|eval
CommandLine=""""|eval SHA256HashData=""""|eval CommandLine_Short=""""|eval TargetFileName=""""|eval
RegObjectName=""""|eval RegValueName=""""|eval ExecutablesWritten{}.FilePath=""""|eval
GrandparentCommandLine=""""|eval ParentCommandLine=""""|eval DetectDescription=""""| fillnull value=""""| map
search=""search ComputerName=$ComputerName$  _time>=$eTimeBefore$ _time<=$eTimeAfter$""| rex
field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$""| rex field=CommandLine_Short
""(?P<CommandLine_Short>\w{75}).*""| fillnull value=""""|regex
DomainName!=""(?i)adobe.com|google.com|newellco.com|outlook.com|microsoft.com|live\.com|skype\.com|footprintdns\.
com|microsoftonline\.com|office365\.com|office\.net|digicert.com|office\.com|windows\.com|lync\.com|apple\.com|windows
\.net|icloud\.net|goody\.com|facebook\.com|jahglobal\.net|0\.0\.0\.0|rackcdn\.com|yammer\.com|newellrubbermaid.com""
|regex CommandLine!=""(?i)CCM|PSScriptPolicyTest|teams|Search.*Robot|SearchFilterHost""| table CommandLine
SHA256HashData _time CommandLine_Short TargetFileName RegObjectName   RegValueName
ExecutablesWritten{}.FilePath   GrandparentCommandLine  ParentCommandLine  DetectDescription DomainName|
fillnull value="""""
```

Search for process treeview by ContextProcessId_decimal

```
https://falcon.crowdstrike.com/investigate/process-explorer/aid/ContextProcessId_decimal
```

Quickly Find out events with commandline and network info based on ComputerName input

```
"ComputerName=COMPUTERNAMEHERE event_simpleName=AssociateIndicator OR source=PlatformEvents    |dedup
TargetProcessId_decimal ComputerName    |join TargetProcessId_decimal  [search
event_simpleName=""ProcessRollup2"" ComputerName=COMPUTERNAMEHERE  | dedup TargetProcessId_decimal  |
fields TargetProcessId_decimal ComputerName timestamp ImageFileName CommandLine TreeId_decimal ]    | rename
TargetProcessId_decimal AS ContextProcessId_decimal         |join type=outer ContextProcessId_decimal  [search
ComputerName=NATL1-8K8L7H2 event_simpleName=DnsRequest   | regex
DomainName!=""(?i)adobe\.com$|google.com$|newellco\.com$|outlook\.com$|microsoft\.com$|live\.com$|skype\.com$|fo
otprintdns\.com$|microsoftonline\.com$|office365\.com$|office\.net$|digicert\.com$|office\.com$|windows\.com$|lync\.com
$|apple\.com$|windows\.net$|icloud\.net$|goody\.com$|facebook\.com$|jahglobal\.net$|0\.0\.0\.0$|rackcdn\.com$|yammer\
.com|office\.com$|msedge\.net$|identrust\.com$|letsencrypt\.org$|msn\.com$|bing\.com$|msocsp\.com$|cloudsink\.net$|s
harepoint\.com$|^localhost$""  | fields DomainName ContextProcessId_decimal IP4Records  | stats values(DomainName)
values(IP4Records) BY ContextProcessId_decimal]    |join type=outer ContextProcessId_decimal  [search
ComputerName=NATL1-8K8L7H2 event_simpleName=NetworkConnect*  | fields RemoteAddressIP4 RemoteIP
RemotePort_decimal ContextProcessId_decimal ]  | eval ""(UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") |rex
field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$"" |rex field=CommandLine_Short
""(?P<CommandLine_Short>\w{75}).*""   | rename ""values(DomainName)"" AS DNS_RESULTS  | rename
""values(IP4Records)"" AS DNS_RESULTS_IP  | table ""(UTC)"" DetectScenario DetectName tactic technique
DetectDescription CommandLine_Short DNS_RESULTS DNS_RESULTS_IP RemotePort_decimal"
```

Quickly Find out events with commandline and sparkline and network info based on ComputerName input and DNS index

```
"ComputerName=""COMPUTERNAMEHERE"" event_simpleName=""DnsRequest"" DomainName=""*.*"" | regex
DomainName!=""(?i)adobe\.com|cloudsink\.net$|..localmachine|listofdomainsyouwanttofilteretc""   |rename
ContextProcessId_decimal as TargetProcessId_decimal  |join TargetProcessId_decimal   [search
ComputerName=""COMPUTERNAMEHERE"" event_simpleName=""ProcessRollup2""  |regex
CommandLine!=""(?i)iexplore\.exe|chrome\.exe|MicrosoftEdgeCP\.exe|firefox\.exe|google|smartscreen\.exe|OneDrive\.ex
e|SearchUI\.exe|mimecast\.com|MicrosoftEdge\.exe""] |rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$"" | eval
""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") |stats sparkline count values(CommandLine)
values(DomainName) dc(""Last Seen (UTC)"") by ""Last Seen (UTC)"" FileName SHA256HashData"
```

Find out more of a timeline for all fields based on computername

```
"ComputerName=""COMPUTERNAMEHERE""  | rex field=CommandLine ""(?<CommandLine_Short>[^\\\\]+)$"" | rex
field=CommandLine_Short ""(?P<CommandLine_Short>\w{75}).*""   | fillnull value="""" | eval ""Last Seen
(UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") |regex DomainName!=""(?i)adobe.com|.*in-addr.arpa""  |regex
CommandLine!=""(?i)CCM|SearchFilterHost"" | table ""Last Seen (UTC)"" FileName CommandLine SHA256HashData
CommandLine_Short TargetFileName RegObjectName RegValueName ExecutablesWritten{}.FilePath
GrandparentCommandLine ParentCommandLine DetectDescription DomainName  | stats count values(FileName)
values(CommandLine) values(CommandLine_Short) values(TargetFileName) values(RegObjectName)
values(RegValueName) values(ExecutablesWritten{}.FilePath) values(GrandparentCommandLine)
values(ParentCommandLine) values(DetectDescription) values(DomainName) by ""Last Seen (UTC)"" SHA256HashData
| sort -""Last Seen (UTC)"""
```

TreeId_decimal tree id process tree sort

```
"aid=XXXXXXXXXXXXXXXXXXXXXXXXXXXX TreeId_decimal=30064919953 | regex
DomainName!=""(?i)adobe.com|google.com|outlook.com|microsoft.com|live\.com|skype\.com|footprintdns\.com|microsofto
nline\.com|office365\.com|office\.net|digicert.com|office\.com|windows\.com|lync\.com|apple\.com|windows\.net|icloud\.net|
goody\.com""   | fillnull value=""NULL""  | rename ContextProcessId_decimal as TargetProcessId_decimal  | join
TargetProcessId_decimal   [search aid=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
event_simpleName=""ProcessRollup2"" earliest=-1@d ]   | rex field=CommandLine ""(?<CommandLine>[^\\\\]+)$"" | stats
values(FileName) values(CommandLine) values(DomainName) count by SHA256HashData"
```

Extract usernames from windows and *nix FilePath and CommandLine with given aid or ComputerName

```
"event_simpleName=""ProcessRollup2"" ComputerName=COMPUTERNAME FilePath=""*Users*"" OR
CommandLine=""*Users*""  | rex field=FilePath mode=sed ""s/.*\bUsers\b.(\w+)(\b.*)/\1/g""  | rex field=CommandLine
mode=sed ""s/.*\bUsers\b.(\w+)(\b.*)/\1/g""  | regex CommandLine!=""(?i).\b.""   | regex FilePath!=""(?i).\b.""    | rename
FilePath AS CommandLine  | rename CommandLine AS UserName | dedup UserName | table UserName"
```

Count how many local admin users logins

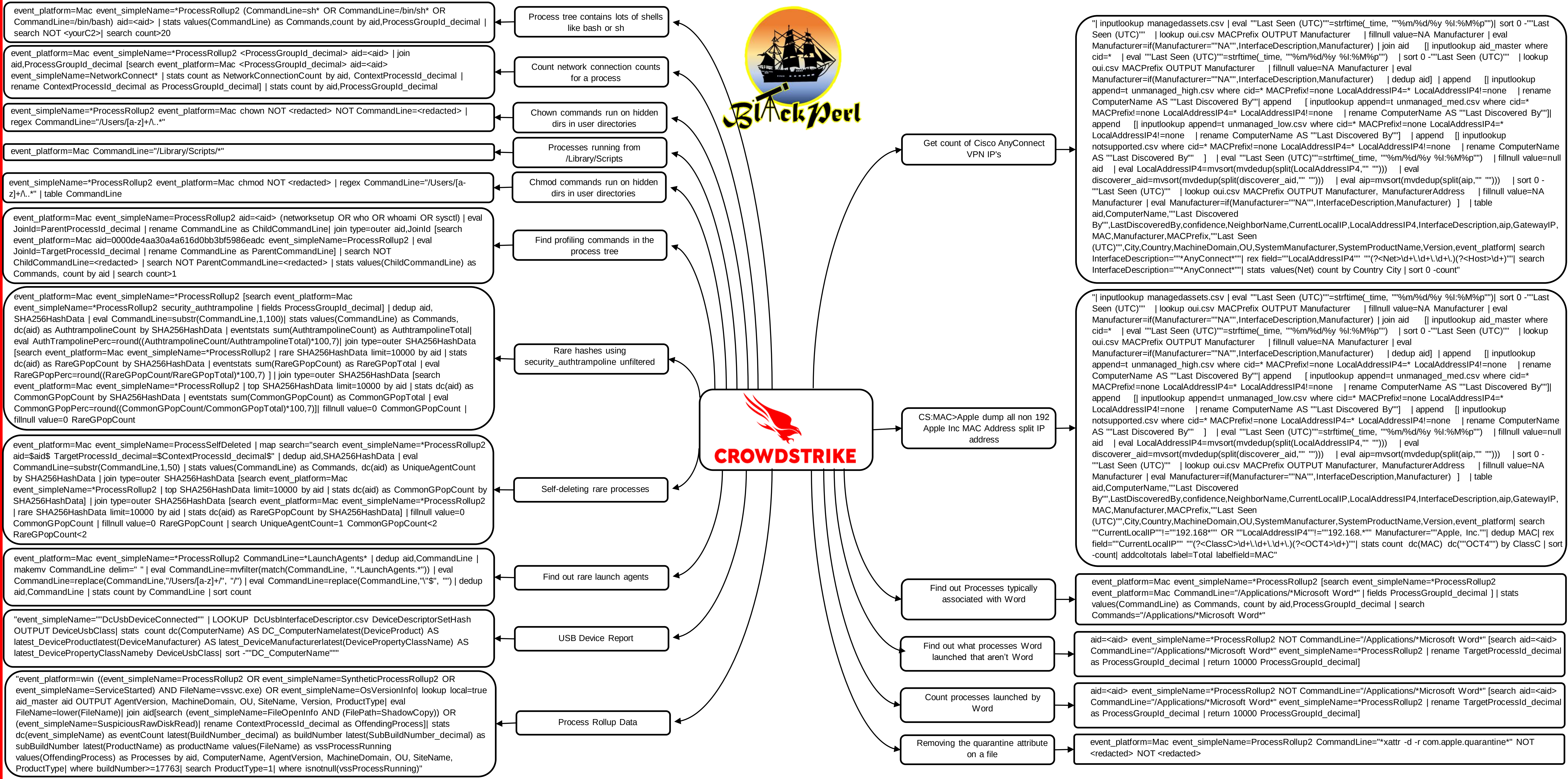
```
"event_simpleName=UserLogon UserSid_readable=S-1-5-21-* UserIsAdmin_decimal=1 earliest=-1d@d | where
ComputerName=LogonDomain | convert ctime(LogonTime_decimal) AS logonTime ctime(PasswordLastSet_decimal) AS
lastPwdReset  | stats sparkline count values(ComputerName) by UserName  | sort -count | where count>5"
```

Find Chrome Remote Desktop Hits Via DNS

```
"event_simpleName=DnsRequest DomainName=""remotedesktop.google.com"" OR DomainName=""remotedesktop-
pa.googleapis.com"" | join type=left ComputerName    [search event_simpleName IN (""UserLogon*"")
UserPrincipal=*.*@*.com UserPrincipal!=*.$*.com   earliest=-1d@d]| stats sparkline count by ComputerName
UserPrincipal| sort -count"
```

Determine if a process is orphaned

```
aid=<aid> <process_id> event_simpleName=ProcessRollup2 | eval JoinId=ParentProcessId_decimal | rename
CommandLine as ChildCommandLine | join type=outer JoinId [search aid=<aid> event_simpleName=ProcessRollup2 |
eval JoinId=TargetProcessId_decimal | rename CommandLine as ParentCommandLine] | eval
ParentCommandLine=coalesce(ParentCommandLine,"IamAnOrphan")
```



Process tree contains lots of shells like bash or sh

```
event_platform=Mac event_simpleName=*ProcessRollup2 (CommandLine=sh* OR CommandLine=/bin/sh* OR
CommandLine=/bin/bash) aid=<aid> | stats values(CommandLine) as Commands,count by aid,ProcessGroupId_decimal |
search NOT <yourC2>| search count>20
```

Count network connection counts for a process

```
event_platform=Mac event_simpleName=*ProcessRollup2 <ProcessGroupId_decimal> aid=<aid> | join
aid,ProcessGroupId_decimal [search event_platform=Mac <ProcessGroupId_decimal> aid=<aid>
event_simpleName=NetworkConnect* | stats count as NetworkConnectionCount by aid, ContextProcessId_decimal |
rename ContextProcessId_decimal as ProcessGroupId_decimal] | stats count by aid,ProcessGroupId_decimal
```

Chown commands run on hidden dirs in user directories

```
event_simpleName=*ProcessRollup2 event_platform=Mac chown NOT <redacted> NOT CommandLine=<redacted> |
regex CommandLine="/Users/[a-z]+/\..*"
```

Processes running from /Library/Scripts

```
event_platform=Mac CommandLine="/Library/Scripts/*"
```

Chmod commands run on hidden dirs in user directories

```
event_simpleName=*ProcessRollup2 event_platform=Mac chmod NOT <redacted> | regex CommandLine="/Users/[a-
z]+/\..*" | table CommandLine
```

Find profiling commands in the process tree

```
event_platform=Mac event_simpleName=ProcessRollup2 aid=<aid> (networksetup OR who OR whoami OR sysctl) | eval
JoinId=ParentProcessId_decimal | rename CommandLine as ChildCommandLine| join type=outer aid,JoinId [search
event_platform=Mac aid=0000de4aa30a4a616d0bb3bf5986eadc event_simpleName=ProcessRollup2 | eval
JoinId=TargetProcessId_decimal | rename CommandLine as ParentCommandLine] | search NOT
ChildCommandLine=<redacted> | search NOT ParentCommandLine=<redacted> | stats values(ChildCommandLine) as
Commands, count by aid | search count>1
```

Rare hashes using security_authtrampoline unfiltered

```
event_platform=Mac event_simpleName=*ProcessRollup2 [search event_platform=Mac
event_simpleName=*ProcessRollup2 security_authtrampoline | fields ProcessGroupId_decimal] | dedup aid,
SHA256HashData | eval CommandLine=substr(CommandLine,1,100)| stats values(CommandLine) as Commands,
dc(aid) as AuthtrampolineCount by SHA256HashData | eventstats sum(AuthtrampolineCount) as AuthtrampolineTotal|
eval AuthTrampolinePerc=round((AuthtrampolineCount/AuthtrampolineTotal)*100,7)| join type=outer SHA256HashData
[search event_platform=Mac event_simpleName=*ProcessRollup2 | rare SHA256HashData limit=10000 by aid | stats
dc(aid) as RareGPopCount by SHA256HashData | eventstats sum(RareGPopCount) as RareGPopTotal | eval
RareGPopPerc=round((RareGPopCount/RareGPopTotal)*100,7) ] | join type=outer SHA256HashData [search
event_platform=Mac event_simpleName=*ProcessRollup2 | top SHA256HashData limit=10000 by aid | stats dc(aid) as
CommonGPopCount by SHA256HashData | eventstats sum(CommonGPopCount) as CommonGPopTotal | eval
CommonGPopPerc=round((CommonGPopCount/CommonGPopTotal)*100,7)]| fillnull value=0 CommonGPopCount |
fillnull value=0 RareGPopCount
```

Self-deleting rare processes

```
event_platform=Mac event_simpleName=ProcessSelfDeleted | map search="search event_simpleName=*ProcessRollup2
aid=$aid$ TargetProcessId_decimal=$ContextProcessId_decimal$" | dedup aid,SHA256HashData | eval
CommandLine=substr(CommandLine,1,50) | stats values(CommandLine) as Commands, dc(aid) as UniqueAgentCount
by SHA256HashData | join type=outer SHA256HashData [search event_platform=Mac
event_simpleName=*ProcessRollup2 | top SHA256HashData limit=10000 by aid | stats dc(aid) as CommonGPopCount by
SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=*ProcessRollup2
| rare SHA256HashData limit=10000 by aid | stats dc(aid) as RareGPopCount by SHA256HashData] | fillnull value=0
CommonGPopCount | fillnull value=0 RareGPopCount | search UniqueAgentCount=1 CommonGPopCount<2
RareGPopCount<2
```

Find out rare launch agents

```
event_platform=Mac event_simpleName=*ProcessRollup2 CommandLine=*LaunchAgents* | dedup aid,CommandLine |
makemv CommandLine delim=" " | eval CommandLine=mvfilter(match(CommandLine, ".*LaunchAgents.*")) | eval
CommandLine=replace(CommandLine,"/Users/[a-z]+/", "/") | eval CommandLine=replace(CommandLine,"\"$", "") | dedup
aid,CommandLine | stats count by CommandLine | sort count
```

USB Device Report

```
"event_simpleName=""DcUsbDeviceConnected"" | LOOKUP  DcUsbInterfaceDescriptor.csv DeviceDescriptorSetHash
OUTPUT DeviceUsbClass| stats  count dc(ComputerName) AS DC_ComputerNamelatest(DeviceProduct) AS
latest_DeviceProductlatest(DeviceManufacturer) AS latest_DeviceManufacturerlatest(DevicePropertyClassName) AS
latest_DevicePropertyClassNameby DeviceUsbClass| sort -""DC_ComputerName"""
```

Process Rollup Data

```
"event_platform=win ((event_simpleName=ProcessRollup2 OR event_simpleName=SyntheticProcessRollup2 OR
event_simpleName=ServiceStarted) AND FileName=vssvc.exe) OR event_simpleName=OsVersionInfo| lookup local=true
aid_master aid OUTPUT AgentVersion, MachineDomain, OU, SiteName, Version, ProductType| eval
FileName=lower(FileName)| join aid[search (event_simpleName=FileOpenInfo AND (FilePath=ShadowCopy)) OR
(event_simpleName=SuspiciousRawDiskRead)| rename ContextProcessId_decimal as OffendingProcess]| stats
dc(event_simpleName) as eventCount latest(BuildNumber_decimal) as buildNumber latest(SubBuildNumber_decimal) as
subBuildNumber latest(ProductName) as productName values(FileName) as vssProcessRunning
values(OffendingProcess) as Processes by aid, ComputerName, AgentVersion, MachineDomain, OU, SiteName,
ProductType| where buildNumber>=17763| search ProductType=1| where isnotnull(vssProcessRunning)"
```

Get count of Cisco AnyConnect VPN IP's

```
"| inputlookup managedassets.csv | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")| sort 0 -""Last
Seen (UTC)""    | lookup oui.csv MACPrefix OUTPUT Manufacturer     | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer) | join aid     [| inputlookup aid_master where
cid=*    | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")    | sort 0 -""Last Seen (UTC)""    | lookup
oui.csv MACPrefix OUTPUT Manufacturer     | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)    | dedup aid]  | append     [| inputlookup
append=t unmanaged_high.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename
ComputerName AS ""Last Discovered By""| append     [ inputlookup append=t unmanaged_med.csv where cid=*
MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename ComputerName AS ""Last Discovered By""]|
append     [| inputlookup append=t unmanaged_low.csv where cid=* MACPrefix!=none LocalAddressIP4=*
LocalAddressIP4!=none    | rename ComputerName AS ""Last Discovered By""]    | append    [| inputlookup
notsupported.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename ComputerName
AS ""Last Discovered By""    ]    | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")    | fillnull value=null
aid    | eval LocalAddressIP4=mvsort(mvdedup(split(LocalAddressIP4,"" "")))    | eval
discoverer_aid=mvsort(mvdedup(split(discoverer_aid,"" "")))    | eval aip=mvsort(mvdedup(split(aip,"" "")))    | sort 0 -
""Last Seen (UTC)""    | lookup oui.csv MACPrefix OUTPUT Manufacturer, ManufacturerAddress     | fillnull value=NA
Manufacturer | eval Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)  ]    | table
aid,ComputerName,""Last Discovered
By"",LastDiscoveredBy,confidence,NeighborName,CurrentLocalIP,LocalAddressIP4,InterfaceDescription,aip,GatewayIP,
MAC,Manufacturer,MACPrefix,""Last Seen
(UTC)"",City,Country,MachineDomain,OU,SystemManufacturer,SystemProductName,Version,event_platform| search
InterfaceDescription=""*AnyConnect*""| rex field=""LocalAddressIP4"" ""(?<Net>\d+\.\d+\.\d+\.)(?<Host>\d+)""| search
InterfaceDescription=""*AnyConnect*""| stats  values(Net) count by Country City | sort 0 -count"
```

CS:MAC>Apple dump all non 192 Apple Inc MAC Address split IP address

```
"| inputlookup managedassets.csv | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")| sort 0 -""Last
Seen (UTC)""    | lookup oui.csv MACPrefix OUTPUT Manufacturer     | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer) | join aid     [| inputlookup aid_master where
cid=*    | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")    | sort 0 -""Last Seen (UTC)""    | lookup
oui.csv MACPrefix OUTPUT Manufacturer     | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)    | dedup aid]  | append     [| inputlookup
append=t unmanaged_high.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename
ComputerName AS ""Last Discovered By""| append     [ inputlookup append=t unmanaged_med.csv where cid=*
MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename ComputerName AS ""Last Discovered By""]|
append     [| inputlookup append=t unmanaged_low.csv where cid=* MACPrefix!=none LocalAddressIP4=*
LocalAddressIP4!=none    | rename ComputerName AS ""Last Discovered By""]    | append    [| inputlookup
notsupported.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none    | rename ComputerName
AS ""Last Discovered By""    ]    | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")    | fillnull value=null
aid    | eval LocalAddressIP4=mvsort(mvdedup(split(LocalAddressIP4,"" "")))    | eval
discoverer_aid=mvsort(mvdedup(split(discoverer_aid,"" "")))    | eval aip=mvsort(mvdedup(split(aip,"" "")))    | sort 0 -
""Last Seen (UTC)""    | lookup oui.csv MACPrefix OUTPUT Manufacturer, ManufacturerAddress     | fillnull value=NA
Manufacturer | eval Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)  ]    | table
aid,ComputerName,""Last Discovered
By"",LastDiscoveredBy,confidence,NeighborName,CurrentLocalIP,LocalAddressIP4,InterfaceDescription,aip,GatewayIP,
MAC,Manufacturer,MACPrefix,""Last Seen
(UTC)"",City,Country,MachineDomain,OU,SystemManufacturer,SystemProductName,Version,event_platform| search
""CurrentLocalIP""!=""192.168*"" OR ""LocalAddressIP4""!=""192.168.*"" Manufacturer=""Apple, Inc.""| dedup MAC| rex
field=""CurrentLocalIP"" ""(?<ClassC>\d+\.\d+\.\d+\.)(?<OCT4>\d+)""| stats count  dc(MAC)  dc(""OCT4"") by ClassC | sort
-count| addcoltotals label=Total labelfield=MAC"
```

Find out Processes typically associated with Word

```
event_platform=Mac event_simpleName=*ProcessRollup2 [search event_simpleName=*ProcessRollup2
event_platform=Mac CommandLine="/Applications/*Microsoft Word*" | fields ProcessGroupId_decimal ] | stats
values(CommandLine) as Commands, count by aid,ProcessGroupId_decimal | search
Commands="/Applications/*Microsoft Word*"
```

Find out what processes Word launched that aren’t Word

```
aid=<aid> event_simpleName=*ProcessRollup2 NOT CommandLine="/Applications/*Microsoft Word*" [search aid=<aid>
CommandLine="/Applications/*Microsoft Word*" event_simpleName=*ProcessRollup2 | rename TargetProcessId_decimal
as ProcessGroupId_decimal | return 10000 ProcessGroupId_decimal]
```

Count processes launched by Word

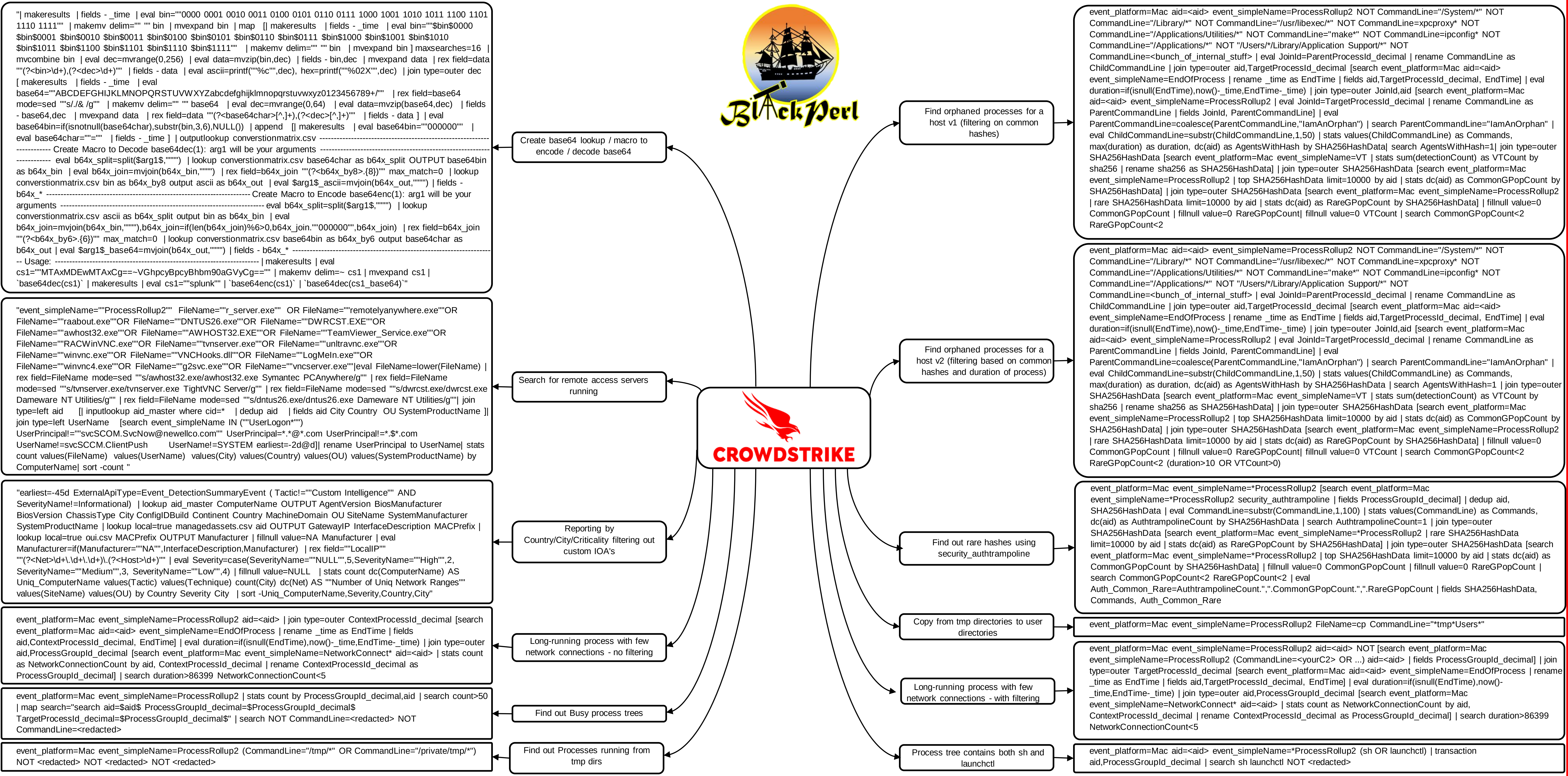
```
aid=<aid> event_simpleName=*ProcessRollup2 NOT CommandLine="/Applications/*Microsoft Word*" [search aid=<aid>
CommandLine="/Applications/*Microsoft Word*" event_simpleName=*ProcessRollup2 | rename TargetProcessId_decimal
as ProcessGroupId_decimal | return 10000 ProcessGroupId_decimal]
```

Removing the quarantine attribute on a file

```
event_platform=Mac event_simpleName=ProcessRollup2 CommandLine="*xattr -d -r com.apple.quarantine*" NOT
<redacted> NOT <redacted>
```

BlackPerl

CROWDSTRIKE



Create base64 lookup / macro to encode / decode base64

```
"| makeresults  | fields - _time  | eval bin=""0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111""  | makemv delim="" "" bin  | mvexpand bin  | map   [| makeresults   | fields - _time   | eval bin=""$bin$0000 $bin$0001 $bin$0010 $bin$0011 $bin$0100 $bin$0101 $bin$0110 $bin$0111 $bin$1000 $bin$1001 $bin$1010 $bin$1011 $bin$1100 $bin$1101 $bin$1110 $bin$1111""   | makemv delim="" "" bin   | mvexpand bin ] maxsearches=16  | mvcombine bin  | eval dec=mvrange(0,256)  | eval data=mvzip(bin,dec)  | fields - bin,dec  | mvexpand data  | rex field=data ""(?<bin>\d+),(?<dec>\d+)""  | fields - data  | eval ascii=printf(""%c"",dec), hex=printf(""%02X"",dec)  | join type=outer dec [ makeresults   | fields - _time   | eval base64=""ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/""   | rex field=base64 mode=sed ""s/./& /g""   | makemv delim="" "" base64   | eval dec=mvrange(0,64)   | eval data=mvzip(base64,dec)   | fields - base64,dec   | mvexpand data   | rex field=data ""(?<base64char>[^,]+),(?<dec>[^,]+)""   | fields - data ]  | eval base64bin=if(isnotnull(base64char),substr(bin,3,6),NULL())  | append   [| makeresults   | eval base64bin=""000000""   | eval base64char=""=""   | fields - _time ]  | outputlookup converstionmatrix.csv ------------------------------------------------------------------------ Create Macro to Decode base64dec(1): arg1 will be your arguments ------------------------------------------------------------------------ eval b64x_split=split($arg1$,"""")  | lookup converstionmatrix.csv base64char as b64x_split OUTPUT base64bin as b64x_bin  | eval b64x_join=mvjoin(b64x_bin,"""")  | rex field=b64x_join ""(?<b64x_by8>.{8})"" max_match=0  | lookup converstionmatrix.csv bin as b64x_by8 output ascii as b64x_out  | eval $arg1$_ascii=mvjoin(b64x_out,"""") | fields - b64x_* ------------------------------------------------------------------------ Create Macro to Encode base64enc(1): arg1 will be your arguments ------------------------------------------------------------------------ eval b64x_split=split($arg1$,"""")  | lookup converstionmatrix.csv ascii as b64x_split output bin as b64x_bin  | eval b64x_join=mvjoin(b64x_bin,""""),b64x_join=if(len(b64x_join)%6>0,b64x_join.""000000"",b64x_join)  | rex field=b64x_join ""(?<b64x_by6>.{6})"" max_match=0  | lookup converstionmatrix.csv base64bin as b64x_by6 output base64char as b64x_out | eval $arg1$_base64=mvjoin(b64x_out,"""") | fields - b64x_* -------------------------------------------------------------------------- Usage: ------------------------------------------------------------------------ | makeresults | eval cs1=""MTAxMDEwMTAxCg==~VGhpcyBpcyBhbm90aGVyCg=="" | makemv delim=~ cs1 | mvexpand cs1 | `base64dec(cs1)` | makeresults | eval cs1=""splunk"" | `base64enc(cs1)` | `base64dec(cs1_base64)`"
```

Search for remote access servers running

```
"event_simpleName=""ProcessRollup2""   FileName=""r_server.exe""   OR FileName=""remotelyanywhere.exe""OR FileName=""raabout.exe""OR FileName=""DNTUS26.exe""OR FileName=""DWRCST.EXE""OR FileName=""awhost32.exe""OR FileName=""AWHOST32.EXE""OR FileName=""TeamViewer_Service.exe""OR FileName=""RACWinVNC.exe""OR FileName=""tvnserver.exe""OR FileName=""unltravnc.exe""OR FileName=""winvnc.exe""OR FileName=""VNCHooks.dll""OR FileName=""LogMeIn.exe""OR FileName=""winvnc4.exe""OR FileName=""g2svc.exe""OR FileName=""vncserver.exe""|eval FileName=lower(FileName) | rex field=FileName mode=sed ""s/awhost32.exe/awhost32.exe Symantec PCAnywhere/g"" | rex field=FileName mode=sed ""s/tvnserver.exe/tvnserver.exe TightVNC Server/g"" | rex field=FileName mode=sed ""s/dwrcst.exe/dwrcst.exe Dameware NT Utilities/g"" | rex field=FileName mode=sed ""s/dntus26.exe/dntus26.exe Dameware NT Utilities/g""| join type=left aid     [| inputlookup aid_master where cid=*    | dedup aid    | fields aid City Country  OU SystemProductName ]| join type=left UserName    [search event_simpleName IN (""UserLogon*"") UserPrincipal!=""svcSCOM.SvcNow@newellco.com"" UserPrincipal=*.*@*.com UserPrincipal!=*.$*.com UserName!=svcSCCM.ClientPush       UserName!=SYSTEM earliest=-2d@d]| rename UserPrincipal to UserName| stats count values(FileName)  values(UserName)  values(City) values(Country) values(OU) values(SystemProductName) by ComputerName| sort -count "
```

Reporting by Country/City/Criticality filtering out custom IOA's

```
"earliest=-45d ExternalApiType=Event_DetectionSummaryEvent ( Tactic!=""Custom Intelligence"" AND SeverityName!=Informational)  | lookup aid_master ComputerName OUTPUT AgentVersion BiosManufacturer BiosVersion ChassisType City ConfigIDBuild Continent Country MachineDomain OU SiteName SystemManufacturer SystemProductName | lookup local=true managedassets.csv aid OUTPUT GatewayIP InterfaceDescription MACPrefix | lookup local=true oui.csv MACPrefix OUTPUT Manufacturer | fillnull value=NA Manufacturer | eval Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)  | rex field=""LocalIP"" ""(?<Net>\d+\.\d+\.\d+)\.(?<Host>\d+)"" | eval Severity=case(SeverityName=""NULL"",5,SeverityName=""High"",2, SeverityName=""Medium"",3, SeverityName=""Low"",4) | fillnull value=NULL   | stats count dc(ComputerName) AS Uniq_ComputerName values(Tactic) values(Technique) count(City) dc(Net) AS ""Number of Uniq Network Ranges"" values(SiteName) values(OU) by Country Severity City   | sort -Uniq_ComputerName,Severity,Country,City"
```

Long-running process with few network connections - no filtering

```
event_platform=Mac event_simpleName=ProcessRollup2 aid=<aid> | join type=outer ContextProcessId_decimal [search event_platform=Mac aid=<aid> event_simpleName=EndOfProcess | rename _time as EndTime | fields aid,ContextProcessId_decimal, EndTime] | eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) | join type=outer aid,ProcessGroupId_decimal [search event_platform=Mac event_simpleName=NetworkConnect* aid=<aid> | stats count as NetworkConnectionCount by aid, ContextProcessId_decimal | rename ContextProcessId_decimal as ProcessGroupId_decimal] | search duration>86399 NetworkConnectionCount<5
```

Find out Busy process trees

```
event_platform=Mac event_simpleName=ProcessRollup2 | stats count by ProcessGroupId_decimal,aid  | search count>50 | map search="search aid=$aid$ ProcessGroupId_decimal=$ProcessGroupId_decimal$ TargetProcessId_decimal=$ProcessGroupId_decimal$" | search NOT CommandLine=<redacted> NOT CommandLine=<redacted>
```

Find out Processes running from tmp dirs

```
event_platform=Mac event_simpleName=ProcessRollup2 (CommandLine="/tmp/*" OR CommandLine="/private/tmp/*") NOT <redacted> NOT <redacted> NOT <redacted>
```

Find orphaned processes for a host v1 (filtering on common hashes)

```
event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 NOT CommandLine="/System/*" NOT CommandLine="/Library/*" NOT CommandLine="/usr/libexec/*" NOT CommandLine=xpcproxy* NOT CommandLine="/Applications/Utilities/*" NOT CommandLine="make*" NOT CommandLine=ipconfig* NOT CommandLine="/Applications/*" NOT "/Users/*/Library/Application Support/*" NOT CommandLine=<bunch_of_internal_stuff> | eval JoinId=ParentProcessId_decimal | rename CommandLine as ChildCommandLine | join type=outer aid,TargetProcessId_decimal [search event_platform=Mac aid=<aid> event_simpleName=EndOfProcess | rename _time as EndTime | fields aid,TargetProcessId_decimal, EndTime] | eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) | join type=outer JoinId,aid [search event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 | eval JoinId=TargetProcessId_decimal | rename CommandLine as ParentCommandLine | fields JoinId, ParentCommandLine] | eval ParentCommandLine=coalesce(ParentCommandLine,"IamAnOrphan") | search ParentCommandLine="IamAnOrphan" | eval ChildCommandLine=substr(ChildCommandLine,1,50) | stats values(ChildCommandLine) as Commands, max(duration) as duration, dc(aid) as AgentsWithHash by SHA256HashData| search AgentsWithHash=1| join type=outer SHA256HashData [search event_platform=Mac event_simpleName=VT | stats sum(detectionCount) as VTCount by sha256 | rename sha256 as SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=ProcessRollup2 | top SHA256HashData limit=10000 by aid | stats dc(aid) as CommonGPopCount by SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=ProcessRollup2 | rare SHA256HashData limit=10000 by aid | stats dc(aid) as RareGPopCount by SHA256HashData] | fillnull value=0 CommonGPopCount | fillnull value=0 RareGPopCount| fillnull value=0 VTCount | search CommonGPopCount<2 RareGPopCount<2
```

Find orphaned processes for a host v2 (filtering based on common hashes and duration of process)

```
event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 NOT CommandLine="/System/*" NOT CommandLine="/Library/*" NOT CommandLine="/usr/libexec/*" NOT CommandLine=xpcproxy* NOT CommandLine="/Applications/Utilities/*" NOT CommandLine="make*" NOT CommandLine=ipconfig* NOT CommandLine="/Applications/*" NOT "/Users/*/Library/Application Support/*" NOT CommandLine=<bunch_of_internal_stuff> | eval JoinId=ParentProcessId_decimal | rename CommandLine as ChildCommandLine | join type=outer aid,TargetProcessId_decimal [search event_platform=Mac aid=<aid> event_simpleName=EndOfProcess | rename _time as EndTime | fields aid,TargetProcessId_decimal, EndTime] | eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) | join type=outer JoinId,aid [search event_platform=Mac aid=<aid> event_simpleName=ProcessRollup2 | eval JoinId=TargetProcessId_decimal | rename CommandLine as ParentCommandLine | fields JoinId, ParentCommandLine] | eval ParentCommandLine=coalesce(ParentCommandLine,"IamAnOrphan") | search ParentCommandLine="IamAnOrphan" | eval ChildCommandLine=substr(ChildCommandLine,1,50) | stats values(ChildCommandLine) as Commands, max(duration) as duration, dc(aid) as AgentsWithHash by SHA256HashData | search AgentsWithHash=1 | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=VT | stats sum(detectionCount) as VTCount by sha256 | rename sha256 as SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=ProcessRollup2 | top SHA256HashData limit=10000 by aid | stats dc(aid) as CommonGPopCount by SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=ProcessRollup2 | rare SHA256HashData limit=10000 by aid | stats dc(aid) as RareGPopCount by SHA256HashData] | fillnull value=0 CommonGPopCount | fillnull value=0 RareGPopCount| fillnull value=0 VTCount | search CommonGPopCount<2 RareGPopCount<2 (duration>10 OR VTCount>0)
```

Find out rare hashes using security_authtrampoline

```
event_platform=Mac event_simpleName=*ProcessRollup2 [search event_platform=Mac event_simpleName=*ProcessRollup2 security_authtrampoline | fields ProcessGroupId_decimal] | dedup aid, SHA256HashData | eval CommandLine=substr(CommandLine,1,100) | stats values(CommandLine) as Commands, dc(aid) as AuthtrampolineCount by SHA256HashData | search AuthtrampolineCount=1 | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=*ProcessRollup2 | rare SHA256HashData limit=10000 by aid | stats dc(aid) as RareGPopCount by SHA256HashData] | join type=outer SHA256HashData [search event_platform=Mac event_simpleName=*ProcessRollup2 | top SHA256HashData limit=10000 by aid | stats dc(aid) as CommonGPopCount by SHA256HashData] | fillnull value=0 CommonGPopCount | fillnull value=0 RareGPopCount | search CommonGPopCount<2 RareGPopCount<2 | eval Auth_Common_Rare=AuthtrampolineCount.",".CommonGPopCount.",".RareGPopCount | fields SHA256HashData, Commands, Auth_Common_Rare
```

Copy from tmp directories to user directories

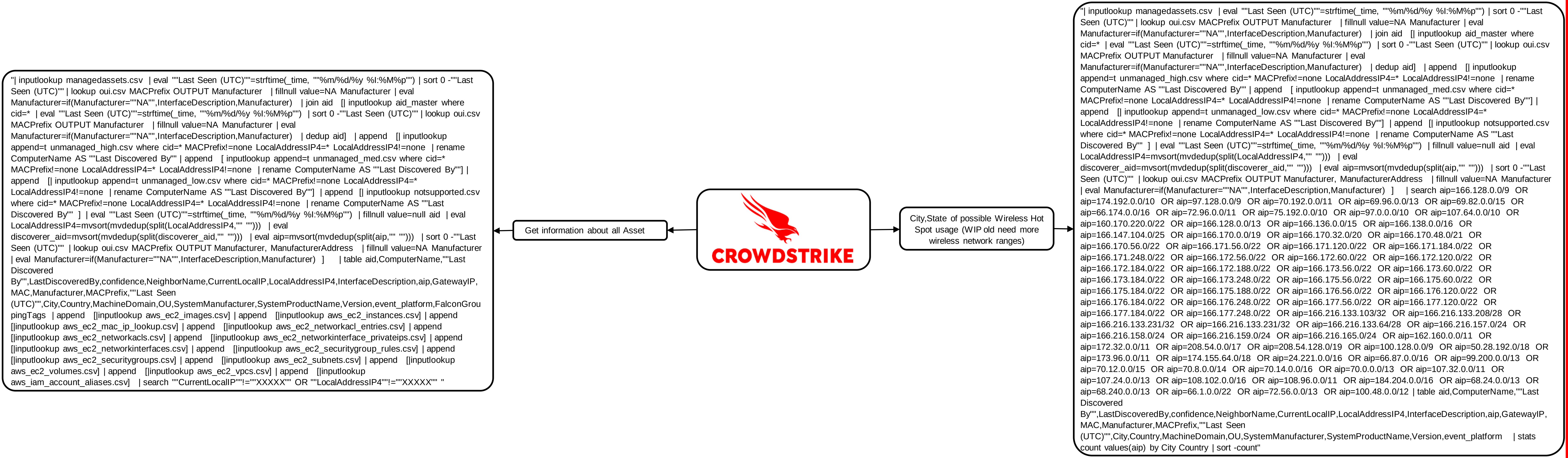
```
event_platform=Mac event_simpleName=ProcessRollup2 FileName=cp CommandLine="*tmp*Users*"
```

Long-running process with few network connections - with filtering

```
event_platform=Mac event_simpleName=ProcessRollup2 aid=<aid> NOT [search event_platform=Mac event_simpleName=ProcessRollup2 (CommandLine=<yourC2> OR ...) aid=<aid> | fields ProcessGroupId_decimal] | join type=outer TargetProcessId_decimal [search event_platform=Mac aid=<aid> event_simpleName=EndOfProcess | rename _time as EndTime | fields aid,TargetProcessId_decimal, EndTime] | eval duration=if(isnull(EndTime),now()-_time,EndTime-_time) | join type=outer aid,ProcessGroupId_decimal [search event_platform=Mac event_simpleName=NetworkConnect* aid=<aid> | stats count as NetworkConnectionCount by aid, ContextProcessId_decimal | rename ContextProcessId_decimal as ProcessGroupId_decimal] | search duration>86399 NetworkConnectionCount<5
```

Process tree contains both sh and launchctl

```
event_platform=Mac aid=<aid> event_simpleName=*ProcessRollup2 (sh OR launchctl) | transaction aid,ProcessGroupId_decimal  | search sh launchctl NOT <redacted>
```

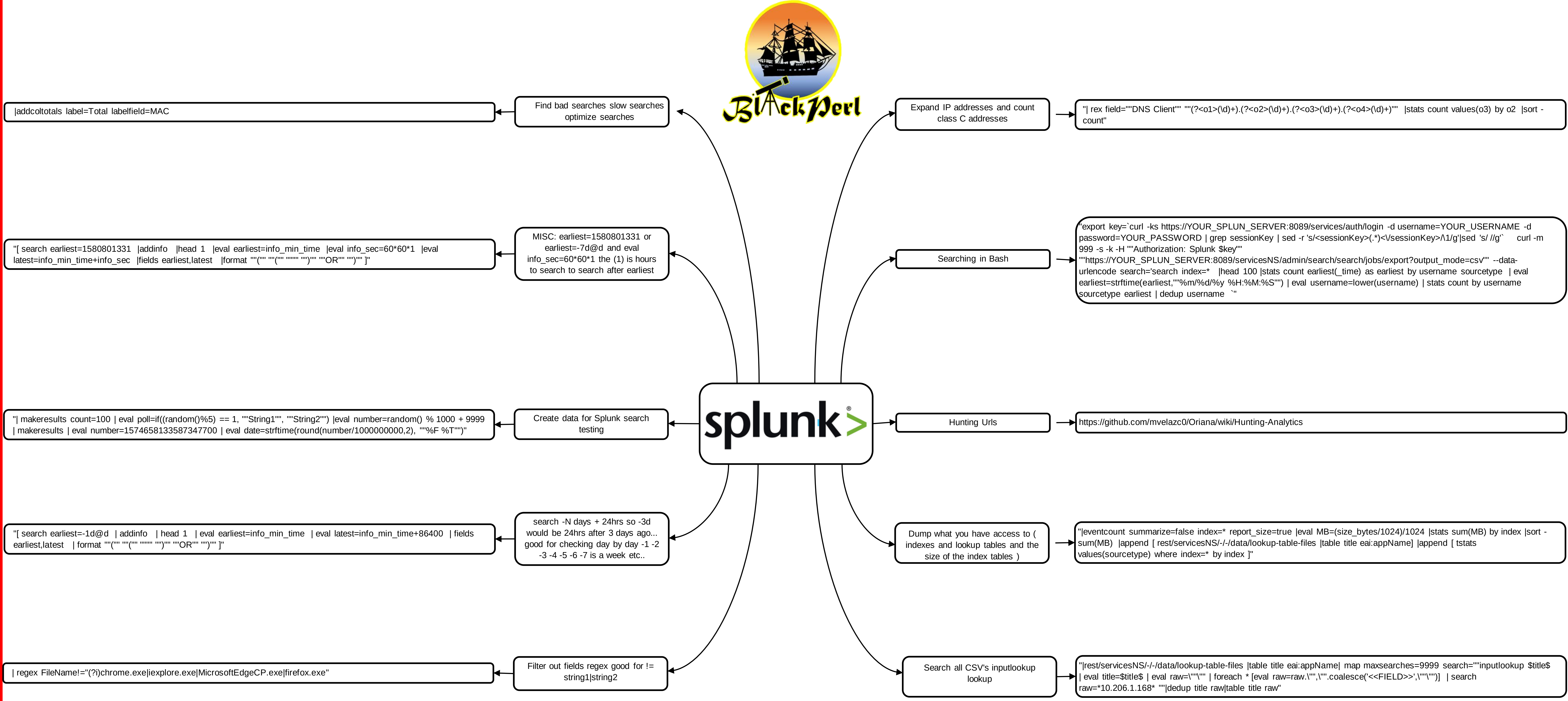


Get information about all Asset

```
"| inputlookup managedassets.csv  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") | sort 0 -""Last
Seen (UTC)"" | lookup oui.csv MACPrefix OUTPUT Manufacturer   | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)   | join aid   [| inputlookup aid_master where
cid=*  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")  | sort 0 -""Last Seen (UTC)"" | lookup oui.csv
MACPrefix OUTPUT Manufacturer   | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)   | dedup aid]   | append   [| inputlookup
append=t unmanaged_high.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename
ComputerName AS ""Last Discovered By"" | append   [ inputlookup append=t unmanaged_med.csv where cid=*
MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename ComputerName AS ""Last Discovered By""] |
append   [| inputlookup append=t unmanaged_low.csv where cid=* MACPrefix!=none LocalAddressIP4=*
LocalAddressIP4!=none  | rename ComputerName AS ""Last Discovered By""]  | append  [| inputlookup notsupported.csv
where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename ComputerName AS ""Last
Discovered By""  ]  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")  | fillnull value=null aid  | eval
LocalAddressIP4=mvsort(mvdedup(split(LocalAddressIP4,"" "")))   | eval
discoverer_aid=mvsort(mvdedup(split(discoverer_aid,"" "")))   | eval aip=mvsort(mvdedup(split(aip,"" "")))   | sort 0 -""Last
Seen (UTC)""  | lookup oui.csv MACPrefix OUTPUT Manufacturer, ManufacturerAddress   | fillnull value=NA Manufacturer
| eval Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)  ]     | table aid,ComputerName,""Last
Discovered
By"",LastDiscoveredBy,confidence,NeighborName,CurrentLocalIP,LocalAddressIP4,InterfaceDescription,aip,GatewayIP,
MAC,Manufacturer,MACPrefix,""Last Seen
(UTC)"",City,Country,MachineDomain,OU,SystemManufacturer,SystemProductName,Version,event_platform,FalconGrou
pingTags  | append   [|inputlookup aws_ec2_images.csv] | append   [|inputlookup aws_ec2_instances.csv] | append
[|inputlookup aws_ec2_mac_ip_lookup.csv] | append   [|inputlookup aws_ec2_networkacl_entries.csv] | append
[|inputlookup aws_ec2_networkacls.csv] | append   [|inputlookup aws_ec2_networkinterface_privateips.csv] | append
[|inputlookup aws_ec2_networkinterfaces.csv] | append   [|inputlookup aws_ec2_securitygroup_rules.csv] | append
[|inputlookup aws_ec2_securitygroups.csv] | append   [|inputlookup aws_ec2_subnets.csv] | append   [|inputlookup
aws_ec2_volumes.csv] | append   [|inputlookup aws_ec2_vpcs.csv] | append   [|inputlookup
aws_iam_account_aliases.csv]   | search ""CurrentLocalIP""!=""XXXXX""  OR ""LocalAddressIP4""!=""XXXXX""  "
```

City,State of possible Wireless Hot Spot usage (WIP old need more wireless network ranges)

```
"| inputlookup managedassets.csv  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"") | sort 0 -""Last
Seen (UTC)"" | lookup oui.csv MACPrefix OUTPUT Manufacturer   | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)   | join aid   [| inputlookup aid_master where
cid=*  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")  | sort 0 -""Last Seen (UTC)"" | lookup oui.csv
MACPrefix OUTPUT Manufacturer   | fillnull value=NA Manufacturer | eval
Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)   | dedup aid]   | append   [| inputlookup
append=t unmanaged_high.csv where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename
ComputerName AS ""Last Discovered By"" | append   [ inputlookup append=t unmanaged_med.csv where cid=*
MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename ComputerName AS ""Last Discovered By""] |
append   [| inputlookup append=t unmanaged_low.csv where cid=* MACPrefix!=none LocalAddressIP4=*
LocalAddressIP4!=none  | rename ComputerName AS ""Last Discovered By""]  | append  [| inputlookup notsupported.csv
where cid=* MACPrefix!=none LocalAddressIP4=* LocalAddressIP4!=none  | rename ComputerName AS ""Last
Discovered By""  ]  | eval ""Last Seen (UTC)""=strftime(_time, ""%m/%d/%y %I:%M%p"")  | fillnull value=null aid  | eval
LocalAddressIP4=mvsort(mvdedup(split(LocalAddressIP4,"" "")))   | eval
discoverer_aid=mvsort(mvdedup(split(discoverer_aid,"" "")))   | eval aip=mvsort(mvdedup(split(aip,"" "")))   | sort 0 -""Last
Seen (UTC)""  | lookup oui.csv MACPrefix OUTPUT Manufacturer, ManufacturerAddress   | fillnull value=NA Manufacturer
| eval Manufacturer=if(Manufacturer=""NA"",InterfaceDescription,Manufacturer)  ]    | search aip=166.128.0.0/9  OR
aip=174.192.0.0/10  OR aip=97.128.0.0/9  OR aip=70.192.0.0/11  OR aip=69.96.0.0/13  OR aip=69.82.0.0/15  OR
aip=66.174.0.0/16  OR aip=72.96.0.0/11  OR aip=75.192.0.0/10  OR aip=97.0.0.0/10  OR aip=107.64.0.0/10  OR
aip=160.170.220.0/22  OR aip=166.128.0.0/13  OR aip=166.136.0.0/15  OR aip=166.138.0.0/16  OR
aip=166.147.104.0/25  OR aip=166.170.0.0/19  OR aip=166.170.32.0/20  OR aip=166.170.48.0/21  OR
aip=166.170.56.0/22  OR aip=166.171.56.0/22  OR aip=166.171.120.0/22  OR aip=166.171.184.0/22  OR
aip=166.171.248.0/22  OR aip=166.172.56.0/22  OR aip=166.172.60.0/22  OR aip=166.172.120.0/22  OR
aip=166.172.184.0/22  OR aip=166.172.188.0/22  OR aip=166.173.56.0/22  OR aip=166.173.60.0/22  OR
aip=166.173.184.0/22  OR aip=166.173.248.0/22  OR aip=166.175.56.0/22  OR aip=166.175.60.0/22  OR
aip=166.175.184.0/22  OR aip=166.175.188.0/22  OR aip=166.176.56.0/22  OR aip=166.176.120.0/22  OR
aip=166.176.184.0/22  OR aip=166.176.248.0/22  OR aip=166.177.56.0/22  OR aip=166.177.120.0/22  OR
aip=166.177.184.0/22  OR aip=166.177.248.0/22  OR aip=166.216.133.103/32  OR aip=166.216.133.208/28  OR
aip=166.216.133.231/32  OR aip=166.216.133.231/32  OR aip=166.216.133.64/28  OR aip=166.216.157.0/24  OR
aip=166.216.158.0/24  OR aip=166.216.159.0/24  OR aip=166.216.165.0/24  OR aip=162.160.0.0/11  OR
aip=172.32.0.0/11  OR aip=208.54.0.0/17  OR aip=208.54.128.0/19  OR aip=100.128.0.0/9  OR aip=50.28.192.0/18  OR
aip=173.96.0.0/11  OR aip=174.155.64.0/18  OR aip=24.221.0.0/16  OR aip=66.87.0.0/16  OR aip=99.200.0.0/13  OR
aip=70.12.0.0/15  OR aip=70.8.0.0/14  OR aip=70.14.0.0/16  OR aip=70.0.0.0/13  OR aip=107.32.0.0/11  OR
aip=107.24.0.0/13  OR aip=108.102.0.0/16  OR aip=108.96.0.0/11  OR aip=184.204.0.0/16  OR aip=68.24.0.0/13  OR
aip=68.240.0.0/13  OR aip=66.1.0.0/22  OR aip=72.56.0.0/13  OR aip=100.48.0.0/12 | table aid,ComputerName,""Last
Discovered
By"",LastDiscoveredBy,confidence,NeighborName,CurrentLocalIP,LocalAddressIP4,InterfaceDescription,aip,GatewayIP,
MAC,Manufacturer,MACPrefix,""Last Seen
(UTC)"",City,Country,MachineDomain,OU,SystemManufacturer,SystemProductName,Version,event_platform    | stats
count values(aip) by City Country | sort -count"
```



Find bad searches slow searches optimize searches

```
|addcoltotals label=Total labelfield=MAC
```

MISC: earliest=1580801331 or earliest=-7d@d and eval info_sec=60*60*1 the (1) is hours to search to search after earliest

```
"[ search earliest=1580801331  |addinfo   |head 1   |eval earliest=info_min_time  |eval info_sec=60*60*1  |eval latest=info_min_time+info_sec  |fields earliest,latest   |format ""("" ""("" """" "")"" ""OR"" "")"" ]"
```

Create data for Splunk search testing

```
"| makeresults count=100 | eval poll=if((random()%5) == 1, ""String1"", ""String2"") |eval number=random() % 1000 + 9999 | makeresults  | eval number=1574658133587347700 | eval date=strftime(round(number/1000000000,2), ""%F %T"")"
```

search -N days + 24hrs so -3d would be 24hrs after 3 days ago... good for checking day by day -1 -2 -3 -4 -5 -6 -7 is a week etc..

```
"[ search earliest=-1d@d  | addinfo   | head 1   | eval earliest=info_min_time  | eval latest=info_min_time+86400  | fields earliest,latest   | format ""("" ""("" """" "")"" ""OR"" "")"" ]"
```

Filter out fields regex good for != string1|string2

```
| regex FileName!="(?i)chrome.exe|iexplore.exe|MicrosoftEdgeCP.exe|firefox.exe"
```

Expand IP addresses and count class C addresses

```
"| rex field=""DNS Client"" ""(?<o1>(\d)+).(?<o2>(\d)+).(?<o3>(\d)+).(?<o4>(\d)+)""  |stats count values(o3) by o2  |sort -count"
```

Searching in Bash

```
"export key=`curl -ks https://YOUR_SPLUN_SERVER:8089/services/auth/login -d username=YOUR_USERNAME -d password=YOUR_PASSWORD | grep sessionKey | sed -r 's/<sessionKey>(.*)<\/sessionKey>/\1/g'|sed 's/ //g'`     curl -m 999 -s -k -H ""Authorization: Splunk $key"" ""https://YOUR_SPLUN_SERVER:8089/servicesNS/admin/search/search/jobs/export?output_mode=csv"" --data-urlencode search='search index=*   |head 100 |stats count earliest(_time) as earliest by username sourcetype  | eval earliest=strftime(earliest,""%m/%d/%y %H:%M:%S"") | eval username=lower(username) | stats count by username sourcetype earliest | dedup username  `"
```

Hunting Urls

https://github.com/mvelazc0/Oriana/wiki/Hunting-Analytics

Dump what you have access to ( indexes and lookup tables and the size of the index tables )

```
"|eventcount summarize=false index=* report_size=true |eval MB=(size_bytes/1024)/1024 |stats sum(MB) by index |sort -sum(MB)  |append [ rest/servicesNS/-/-/data/lookup-table-files |table title eai:appName] |append [ tstats values(sourcetype) where index=* by index ]"
```

Search all CSV's inputlookup lookup

```
"|rest/servicesNS/-/-/data/lookup-table-files |table title eai:appName| map maxsearches=9999 search=""inputlookup $title$ | eval title=$title$ | eval raw=\""\"" | foreach * [eval raw=raw.\"",\"".coalesce('<<FIELD>>',\""\"")]  | search raw=*10.206.1.168* ""|dedup title raw|table title raw"
```