Your META(DATA) is QUEEN for Cyber-Security Vulnerability Management

Shell

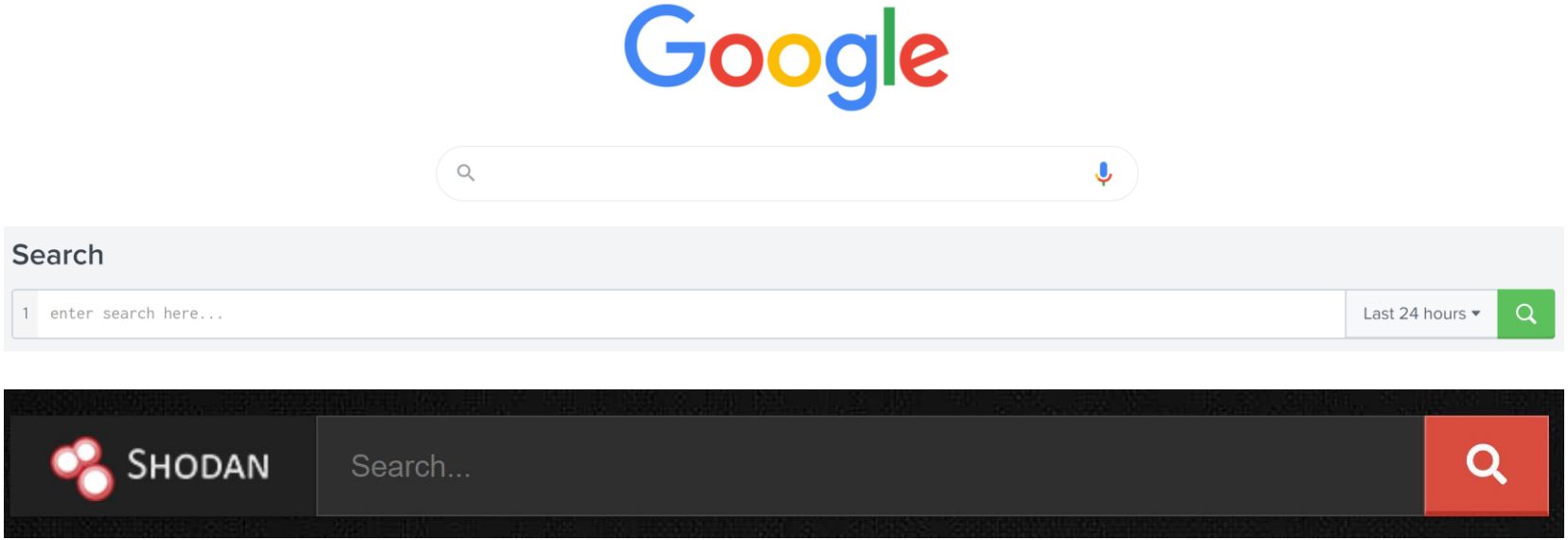
splunk> turn data into doing™

How can you define Splunk?

splunk> turn data into doing™

What I see about Splunk being…

A time-based, Search engine



```
`asset` conf.splunk.com
| table asset_owner asset_business_unit asset_exposure asset_exposed_ports
```

```
`asset` (AWS OR Azure) virtual machine internet facing
| table asset_name asset_ad_domain asset_ad_memberof asset_installed_agent
```

```
`asset` tcp (80 OR 443 OR 8443) internet facing (BigIP OR Big-IP OR TMUI)
| table a*name a*type a*owner a*country* a*ports a*last_seen a*service
```

```
`asset` "solarwinds agent"
| table a*name a*type a*owner a*platform a*last_logged_on
```

Daniel Ferreira

Vulnerability Management Lead
Global Cyber-Security Team
Shell IT International B.V.

This keynote is based on

.conf21's

SEC1075A - Effective and affordable Cyber-Security Vulnerability Management

New content: more real examples of metadata sources and search use cases useful for cyber-security vulnerability management

It is all about knowing what you have

Assets

Software

splunk> turn data into doing™

Asset inventory sources

Machine-generated only sources, they all produce a valid **Hostname**

Always avoid ‘human generated’ asset inventory data (aka spreadsheets).

Scanners

Traditional appliance-based scanners

nmap

nmap ‘discovery-only’ scripts for isolated networks or in case you don’t have scanners

Cloud APIs

Enumeration of VMs, storage, load balancers, etc.

Agents

EDR agent, AV agent, scanning agent, all other agents you can read from

Active Directory

Enumeration of AD domain trust, later proceed with enumeration of Computers per domain

However

Merge problem

A lot of **|inputlookup**, **|append** or **|join** commands are required. This solution is slow and does not scale on asset inventories with millions of rows and daily refreshes.

The PREFIX() directive game-changer

PREFIX() = Ability to **retrieve** terms from .tsidx!

- Before Splunk 8, we were only capable of **querying** terms with the **TERM()** directive in **|tstats**, but not having them back to the pipeline (retrieve them).
  - In 'OR' conditions, you never knew which term you hit.
  - E.g. TERM(process=powershell.exe) OR TERM(process=cmd.exe)

Data Models are not needed.

Note there is no 'FROM' clause!

```
| tstats
    latest(PREFIX(scheduled_time=)) AS latest_schedule_time
    WHERE
    index=_internal
    source=/opt/splunk/var/log/splunk/scheduler.log
    earliest=@d
    TERM(daniel.ferreira@secret.com)
    TERM(important-scheduled-report-name)
    BY
    PREFIX(status=)
    PREFIX(success=)
| rename *= AS *
```

✓ **656 events** (7/23/21 12:00:00.000 AM to 1/19/38 4:14:07.000 AM) No Event Sampling

Events (656) Patterns **Statistics (1)** Visualization

100 Per Page ▾ Format Preview ▾

status	success
delegated_remote_completion	1

Introducing the ‘dotnotation’ property

Making all JSON data sources **PREFIX()**-ready

```
{
    "dnshostname": "WSAMSW89102.addomain.local",
    "cn": "WSAMSW89102",
    "lastlogontimestamp": "2021-07-15T09:50:07.6943274+02:00",
    "operatingsystemversion": "10.0 (18901)",
    "operatingsystem": "Windows 10 Enterprise",
    "dotnotation": [
        "cn=WSAMSW89102",
        "dnshostname=WSAMSW891025.addomain.local"
    ]
}
```

Guaranteeing presence of key fields in **PREFIX()**-ready format.

- Asset name, hostname, FQDN or any other DNS information is always preferable.
- Asset ID is not mandatory as it will vary per source system.
- IP address only if no DNS datapoint is seen.

- Not meant for ‘all properties’, **just ‘key fields’**.
- Trade offs:
  - Some bytes more per event on license.
  - Some bytes more per event on storage.
  - Processing compute power (cheap alternative: server-less).
  - ⚠️ Be careful with prefixes clashes.

Get **ConvertTo-DotNotation** PowerShell cmdlet from Github.

```
| tstats
    fillnull_value=null
    latest(_time) AS asset_last_seen_epoch

    ``` CMDB: ```
    latest(PREFIX(sys_id=)) AS cmdb__sys_id

    ``` Scanner: ```
    latest(PREFIX(asset_id=)) AS scanner__asset_id

    WHERE
    (index=cmdb OR index=scanner OR index=ad)
    sourcetype=asset:inventory
    earliest=-2mon

    BY
    PREFIX(name=)          ``` name:       DNS name coming from CMDB ```
    PREFIX(cn=)            ``` cn:         DNS name coming from Active Directory ```
    PREFIX(host_name=)     ``` host_name:  DNS name coming from Scanner ```
                           ``` Why not 'host' field? Because it can have different meaning or unexpected value: e.g. UF hostname. ```
| rename *= AS *

``` Convert "null" to real null() to enable coalesce to work: ```
| foreach name,cn,host_name [ | eval <<FIELD>>=if(match(<<FIELD>>,"^null$"), null(), <<FIELD>>) ]

``` Coalesce fields across data sources: ```
| eval asset_id=coalesce(cmdb__sys_id, scanner__asset_id)
| eval asset_name=coalesce(name, cn, host_name)

``` Enforce NetBIOS-equivalent DNS name: ```
| rex field=asset_name "(?<asset_name>[^.]*)"

``` Merge: ```
| stats max(asset_last_seen_epoch) AS asset_last_seen_epoch, values(asset_id) AS asset_id BY asset_name
```

Alive assets: your asset inventory contains assets seen ‘alive’ over the last **2 months.**

Because you only import ‘alive assets as of yesterday’, per source, per day; you’re naturally embedding ‘asset decay’ for assets not showing up over the period.

Last 24 hours

✓ 3 events (7/22/21 8:37:11.000 PM to 1/19/38 4:14:07.000 AM) No Event Sampling

Job Fast Mode

Events Patterns Statistics (1) Visualization

100 Per Page Format Preview

asset_name	asset_last_seen_epoch	asset_id
wsamsw89102	1627063915	2de56707148c2e303fbebb645267a284613135
```

Asset Enrichment Feeds

1) CMDB
   - Business, Sub-business, Service provider, Owner, Deployment, Class, Status, etc.
2) Cloud tenants: Azure, AWS, Google
   - Platform, Owner, Owner by action, Last seen, ID, Local users, Installed software, etc.
3) Agents (EDR, AV, sysmon, etc):
   - OS, Installed software, Installed KBs, Missing KBs, Last reboot, Logged on, etc.
4) Active directory:
   - Last seen, Memberof, Forest, Domain, OUs, custom properties, etc.
5) Custom pulls (local scripts, UF scripts, etc):
   - Installed KBs, Last reboot, Installed software, Local users, etc.
6) Custom scans:
   - HTTP Headers, HTML Title, Redirect URLs, etc.
7) Any other reliable asset information...

However

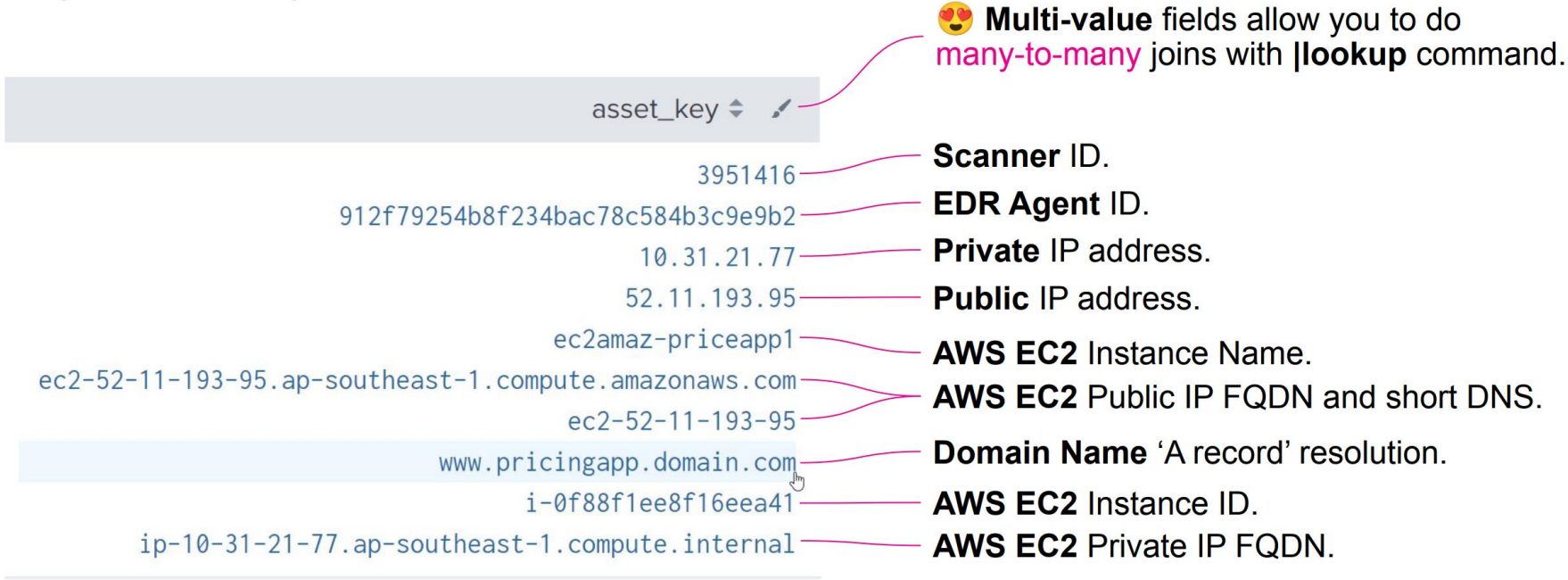
Merge problem again 😟

How do we merge all of these data repositories? What to use as ‘**unique key**’ field?

Answer: we do not need a single ‘unique key’, instead we can have ‘**many possible unique keys**’ 😉

Introducing multi-value: asset_key

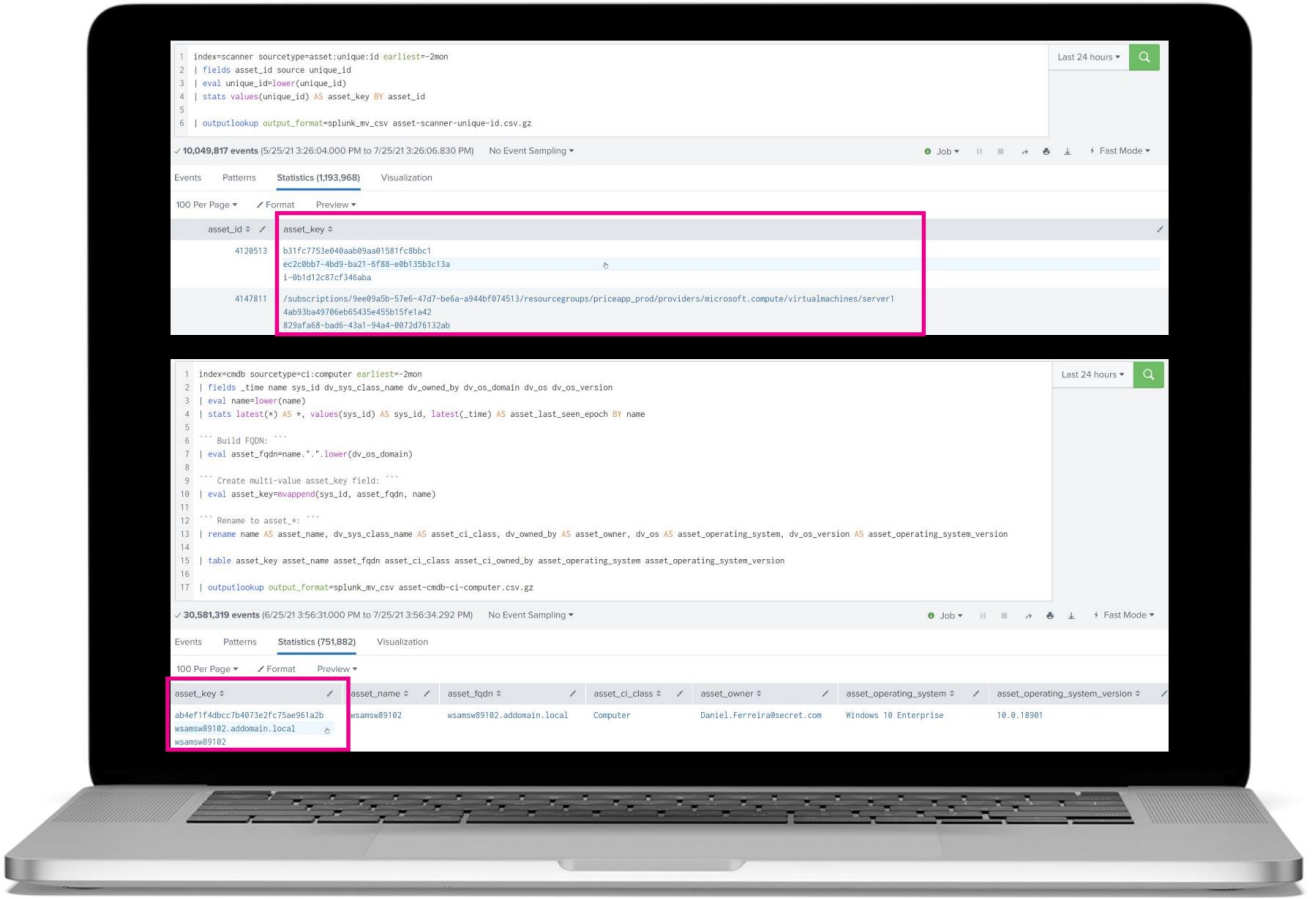
All possible ‘unique identifiers’ for an IT asset



Integration lookups

Each enrichment source will have its own lookup

Prepare all asset enrichment data in **independent lookup files**, all with its very unique ‘**asset_key**’ multi-value field.



```
25
26  ``` Coalesce fields across data sources: ```
27  | eval asset_id=coalesce(cmdb__sys_id, scanner__asset_id)
28  | eval asset_name=coalesce(name, cn, host_name)
29
30  ``` Enforce NetBIOS-equivalent DNS name: ```
31  | rex field=asset_name "(?<asset_name>[^.]*)"
32
33  ``` Merge: ```
34  | stats max(asset_last_seen_epoch) AS asset_last_seen_epoch, values(asset_id) AS asset_id BY asset_name
35
36  ``` asset_key definition: ```
37  | eval asset_key=mvappend(asset_id, asset_name)
38
39  ``` Bring other possible unique-identifiers for the asset when they have agent installed. Coming from scanner, assets with agents installed: ```
40  | lookup asset-scanner-unique-id.csv.gz asset_id AS asset_key OUTPUT asset_key AS temp_scanner_unique_id__asset_key
41  | eval asset_key=mvdedup(mvappend(asset_key, temp_scanner_unique_id__asset_key))
42
43  ``` Bring CMDB enrichment fields: ```
44  | lookup asset-cmdb-ci-computer.csv.gz asset_key OUTPUT asset_key AS temp_cmdb__asset_key, asset_ci_class asset_owner asset_operating_system asset_operating_system_version
45  | eval asset_key=mvdedup(mvappend(asset_key, temp_cmdb__asset_key))
46
47  | table asset_key asset_name a*
```

asset_key: initial definition using all unique identifiers coming from the ‘line-item generation command’ and DNS name.

Many-to-many match: search in each lookup by all possible values seen in ‘asset_key’ field. Then, bring all enrichment fields.

Add more values to ‘asset_key’: each enrichment source may have detected other unique identifiers for the asset, so bring them all into ‘asset_key’ too.

✓ **3 events** (7/22/21 5:11:14.000 PM to 1/19/38 4:14:07.000 AM) No Event Sampling ▾ Job ▾ Fast Mode ▾

Events Patterns **Statistics (1)** Visualization

100 Per Page ▾ Format Preview ▾

asset_key	asset_name	asset_ci_class	asset_id	asset_last_seen_epoch	asset_operating_system	asset_operating_system_version	asset_owner
2de56707148c2e303fbebb645267a284613135 wsamsw89102 wsamsw89102.addomain.local	wsamsw89102	Computer	2de56707148c2e303fbebb645267a284613135	1627063915	Windows 10 Enterprise	10.0.18901	Daniel.Ferreira@secret.com

```
### ACTIVE DIRECTORY:
```
| lookup asset-active-directory.csv.gz asset_key OUTPUT asset_domain_cn, asset_ad_domain, asset_ad_distinguished_name, asset_ad_last_logon_timestamp, asset_operating_system AS temp_ad__asset_os, asset_operating_system_version AS temp_ad__asset_os_version, asset_ad_memberof
| eval asset_operating_system=coalesce(temp_ad__asset_os, asset_operating_system)
, asset_operating_system_version=coalesce(temp_ad__asset_os_version, asset_operating_system_version)
, asset_operating_system_source=if(isnull(asset_domain_cn), asset_operating_system_source, "AD:Computers")
, asset_operating_system_certainty=if(isnull(asset_domain_cn), asset_operating_system_certainty, "1")
, asset_is_unlocked=if(isnull(asset_domain_cn), asset_is_unlocked, if(match(lower(asset_ad_distinguished_name),"ou=admin rights") AND match(lower(asset_ad_distinguished_name),"ou=workstations"), "true", asset_is_unlocked))
, asset_is_domain_joined=if(isnull(asset_domain_cn),"false","true")
```

**AD Groups this asset belongs to:** remember this for later...

```
👀: define first field 'asset_operating_system_source'.
👀: define first field 'asset_operating_system_certainty'.
👀 : detect admin-rights workstations. Define the field first.
```

```
### EDR:
```
| lookup asset-edr.csv.gz asset_key OUTPUT asset_key AS temp_edr__asset_key, asset_os_name AS temp_edr__asset_os_name, asset_os_version AS temp_edr__asset_os_version
| eval asset_operating_system=coalesce(temp_edr__asset_os_name, asset_operating_system)
, asset_operating_system_version=coalesce(temp_edr__asset_os_version, asset_operating_system_version)
, asset_operating_system_source=if(isnull(temp_edr__asset_key), asset_operating_system_source, "EDR")
, asset_operating_system_certainty=if(isnull(temp_edr__asset_key), asset_operating_system_certainty, "1")
, asset_installed_agent=if(isnull(temp_edr__asset_key), asset_installed_agent, mvappend(asset_installed_agent,"EDR-vendor-X"))
, asset_key=if(isnull(temp_edr__asset_key), asset_key, mvdedup(mvappend(asset_key,temp_edr__asset_key)))

```
### EDR: Last logged on:
```
| lookup asset-edr-userlogon.csv.gz asset_key OUTPUT asset_last_logged_on_username AS temp_edr__asset_last_logged_on_username
| eval asset_last_logged_on=mvappend(asset_last_logged_on, temp_edr__asset_last_logged_on_username)

```
### EDR: Host applied vulnerability kb:
```
| lookup asset-edr-installed-kb.csv.gz asset_key OUTPUT asset_installed_vulnerability_kb AS temp_edr__installed_kb
| eval asset_installed_vulnerability_kb=mvappend(asset_installed_vulnerability_kb, temp_edr__installed_kb)

```
### EDR: Host last reboot::
```
| lookup asset-edr-last-reboot.csv.gz asset_key OUTPUT asset_last_reboot_epoch AS temp_edr__asset_last_reboot_epoch
| eval asset_last_reboot_epoch=if(isnull(temp_edr__asset_last_reboot_epoch), asset_last_reboot_epoch, temp_edr__asset_last_reboot_epoch)
```

Installed KBs per machine: thereby the opposite can also be calculated (assets missing certain KB), in conjunction with ‘asset_operating_system*’ fields.

```
88  ```
89  ### Azure Virtual Machines:
90  ```
91  | lookup asset-cloud-azure-vms.csv.gz asset_key OUTPUT asset_key AS temp_az__asset_key, asset_platform AS temp_az__asset_platform, asset_platform_description AS temp_az__asset_platform_description, asset_owner_by_action AS
      temp_az__asset_owner_by_action, asset_owner AS temp_az__asset_owner, asset_installed_agent AS temp_az__asset_installed_agent
92  | eval asset_platform=if(isnull(temp_az__asset_key), asset_platform, temp_az__asset_platform)
93  , asset_platform_description=if(isnull(temp_az__asset_key), asset_platform_description, temp_az__asset_platform_description)
94  , asset_owner=mvdedup(mvappend(asset_owner,temp_az__asset_owner))
95  , asset_owner_by_action=if(isnull(temp_az__asset_owner_by_action), asset_owner_by_action, mvdedup(mvappend(asset_owner_by_action,temp_az__asset_owner_by_action)))
96  , asset_key=if(isnull(temp_az__asset_key), asset_key, mvdedup(mvappend(asset_key,temp_az__asset_key)))
97  , asset_installed_agent=mvdedup(mvappend(asset_installed_agent, temp_az__asset_installed_agent))
98
99  ```
100 ### AWS Virtual Machines:
101 ```
102 | lookup asset-cloud-aws-vms.csv.gz asset_key OUTPUT asset_key AS temp_aws__asset_key, asset_platform AS temp_aws__asset_platform, asset_platform_description AS temp_aws__asset_platform_description, asset_owner_by_action AS
      temp_aws__asset_owner_by_action, asset_owner AS temp_aws__asset_owner
103 | eval asset_platform=if(isnull(temp_aws__asset_key), asset_platform, temp_aws__asset_platform)
104 , asset_platform_description=if(isnull(temp_aws__asset_key), asset_platform_description, temp_aws__asset_platform_description)
105 , asset_owner=mvdedup(mvappend(asset_owner,temp_aws__asset_owner))
106 , asset_owner_by_action=if(isnull(temp_aws__asset_owner_by_action), asset_owner_by_action, mvdedup(mvappend(asset_owner_by_action,temp_aws__asset_owner_by_action)))
107 , asset_key=if(isnull(temp_aws__asset_key), asset_key, mvdedup(mvappend(asset_key,temp_aws__asset_key)))
108
```

asset_key	asset_name	asset_platform	asset_platform_description
3951416 912f79254b8f234bac78c584b3c9e9b2 10.31.21.77 52.11.193.95 ec2amaz-priceapp1 ec2-52-11-193-95.ap-southeast-1.compute.amazonaws.com ec2-52-11-193-95 www.pricingapp.domain.com i-0f88f1ee8f16eea41 ip-10-31-21-77.ap-southeast-1.compute.internal	ec2amaz-priceapp1	AWS	AWS Account: SD9100039 (Retail Price App - Prod) AWS Instance name: ec2amaz-priceapp1 AWS EC2 virtual machine instance id: i-0f88f1ee8f16eea41 AWS Public IP: 52.11.193.95

An official website of the United States government Here's how you know

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

NVD MENU

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

NVD

PRODUCTS

Official Common Platform Enumeration (CPE) Dictionary

CPE is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name.

CPE Dictionary

1. Official CPE Dictionary v2.3, gz format - 14.92 MB, Updated: 10/08/2022; 12:39:58 AM -0400

```
1 | inputlookup NVD-CPE.csv.gz
2 | search cpe_key="cpe:/a:microsoft:exchange_server:2019*"
```

Last 24 hours

✓ 26 results (10/8/22 5:50:26.000 PM to 10/9/22 5:50:26.000 PM) No Event Sampling

Job Fast Mode

Events Patterns Statistics (26) Visualization

100 Per Page Format Preview

cpe_key	max_version	software_product	software_vendor
cpe:/a:microsoft:exchange_server:2019:cumulative_update_12	2019.0	exchange_server	microsoft

splunk> turn data into doing

```
index=shodan source=search:* earliest=@w

| eval asset_shodan_tags=coalesce('tags{}', tags)
| eval asset_seen_cpe=mvappend('cpe23{}', cpe23, 'cpe{}', cpe)
| eval asset_host_name=lower(coalesce('hostnames{}', hostnames, 'ntlm.fqdn'))
| eval asset_seen_product=serviceport." product = ".product
| eval asset_service_banner=null(),
    , asset_service_banner=mvappend(asset_service_banner, coalesce(serviceport, "unknown")."  http.banner.server=".coalesce('http.server', "unknown"))
    , asset_service_banner=mvappend(asset_service_banner, coalesce(serviceport, "unknown")."  http.title = ".coalesce('http.title', "unknown"))
    , asset_service_banner=mvappend(asset_service_banner, asset_seen_product)
| eval asset_network_asn=org
| eval asset_extranet_available_ports=serviceport
| eval asset_certificate_expires=serviceport."=".'ssl.cert.expires'
| eval asset_certificate_cn=serviceport."=".ssl.cert.subject.CN
| eval asset_ip_address=ip_str
| eval asset_is_seen_in_shodan="true"
| eval asset_exposure="extranet"
| eval asset_is_extranet_exposed="true"
| eval asset_fqdn=mvdedup(lower('ntlm.fqdn'))
| eval asset_netbios=mvdedup(lower('netbios_computer_name'))

| fields asset*
| stats values(*) AS *, latest(asset_host_name) AS asset_host_name BY asset_ip_address

| rex field=asset_host_name "(?<asset_short_name>[^\.]*)"
| rex field=asset_fqdn "(?<asset_fqdn_short_name>[^\.]*)"
| eval asset_short_name=if(isnull(asset_short_name) AND !isnull(asset_ip_address), asset_ip_address, asset_short_name)

| eval asset_service_banner=mvfilter(!match(asset_service_banner,"\snull$"))
| eval asset_service_banner=mvfilter(!match(asset_service_banner,"\sunknown$"))
| eval asset_service_banner=mvfilter(!match(asset_service_banner,"\sNot\sFound$"))
| eval asset_service_banner=mvfilter(!match(asset_service_banner,"\sError$"))

| eval asset_key=mvdedup(mvappend(asset_ip_address, asset_host_name, asset_short_name, asset_netbios, asset_fqdn, asset_fqdn_short_name))

| table asset_ip_address asset_short_name asset_key asset*

| outputlookup output_format=splunk_mv_csv asset-enrichment-shodan.csv.gz
```

Owner By Action

Important fields - example

Keeping difference between owners: officially registered owners should keep their own fields, so ‘discovered’ ones can be identified.

Owner by action:

- Anybody that ‘touches’ a resource within a Cloud Subscription/Account becomes automatically an owner.
  - Need to onboard to Splunk the Audit Logs for all Azure Subscriptions / AWS Accounts.
    (you should onboard this regardless)

asset_key	asset_platform	asset_platform_description	asset_owner	asset_owner_key	asset_comment
11ab9f3e57a7470db9d729200b538a514156479 conq-eh-test 10.201.21.40 conq-eh-test.addomain.local a1277570-cd85-486f-a7ef-c077c00ac4b1	Azure	Azure tenant: aztenant.onmicrosoft.com Azure subscription id: 8180ab5b-d0c3-41a7-8e91-220ed8d721bb Azure subscription name: Conq-App-TEST Azure resource group: ConQ-EH-TEST-521341 Azure virtual machine: ConQ-EH-TEST	unknown	daniel.ferreira@secret.com	Owner-by-action found: owner(s) 'daniel.ferreira@secret.com' found performing administrative actions on Azure Subscription 'Conq-App-TEST' and Resource Group 'ConQ-EH-TEST-521341'.

Freeze asset inventory data daily

Usage of **|collect** command

```
2148
2149      ```
2150      ### Set _time as of "today at the top of the day" and increase seconds each 100 rows to avoid "sub-second" error:
2151      ```
2152  | streamstats count as row
2153  | eval _time=relative_time(now(),"@d")+(row/100)
2154
2155      ```
2156      ### Convert all Multi-value fields into non-mv by using the " | " delimiter.
2157      ```
2158  | foreach asset* [ eval <<FIELD>>=if(mvcount(<<FIELD>>)>1,mvjoin(<<FIELD>>, " | "),<<FIELD>>) ]
2159
2160      ```
2161      ### Table desired fields:
2162      ```
2163  | table asset_key asset*
2164
2165      ```
2166      ### COLLECT:
2167      ```
2168  | collect index=summary source=asset
```

😱 … a few months later …

`asset` macro definition:

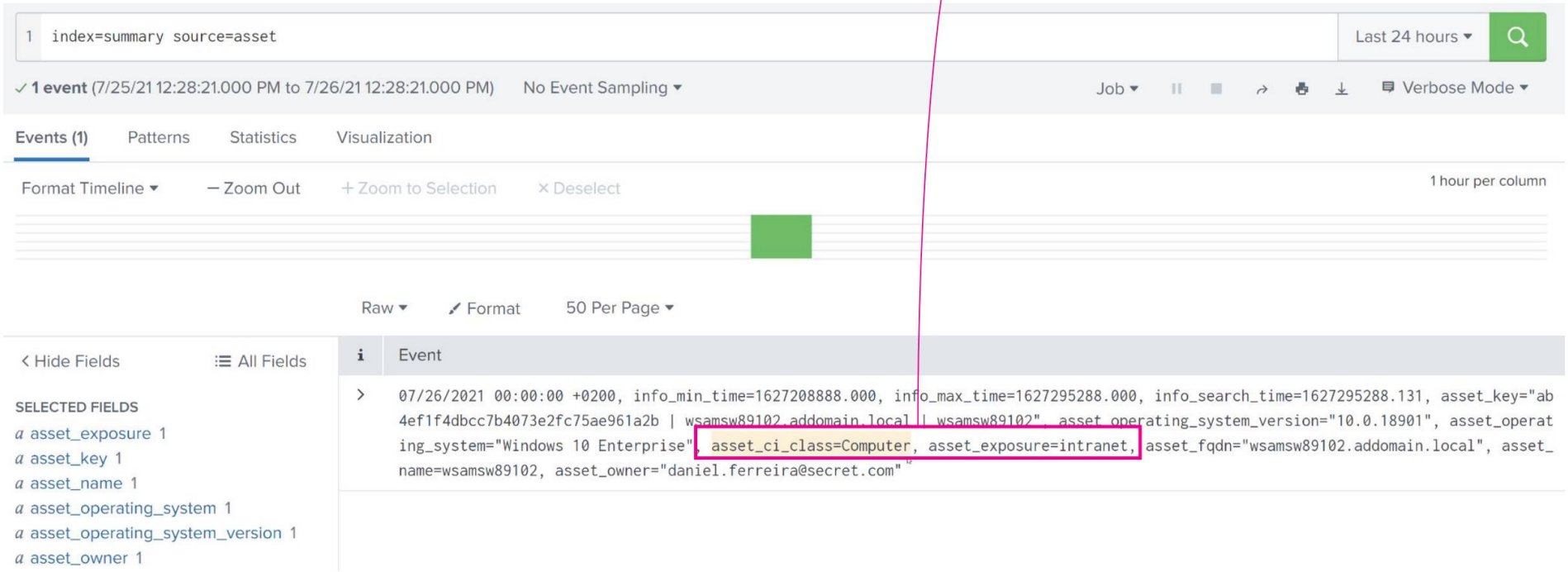
index=summary source=asset earliest=-01d@d latest=@d

Why the 'earliest' and 'latest'? Because the idea is that Asset Inventory Summary Index gets updated once a day, and all automations required for it will take a while to execute (e.g. AD integration), potentially hours. This timing problem will make ASIA-PAC users unable to search at their morning. So, Asset Inventory in **`asset`** macro is "updated as of yesterday" by default (unless you resolve it manually and move it to earliest=@d). You should also create a KV lookup containing updated data as of today, which then can feed Splunk ES —if you have it— and allow users to **|lookup** it in their other searches.

Summary index results

Stash format, automatic field extraction

💛💙❤️ **TERM() & PREFIX() ready fields:**
Summary Index data is in ‘stash format’, which will make your asset inventory data **|tstats ready**.



Are you still following? 🙃

Let’s see real use-cases!

splunk> turn data into doing

Asset inventory search examples

Use-case: breaching report on an internet facing application

```
1 `asset` pricingapp.domain.com
```

```
1 `asset` 52.11.193.95
```

```
1 `asset` retail app
```

```
1 `asset` aws virtual machine public ip prod*
```

```
1 `asset` i-0f88f1ee8f16eea41 ec2
```

```
1 `asset` 921f79254b8f234bac78c584b3c9e9b2
```

```
1 `asset` aws retail machine
```

```
1 `asset` 10.31.21.77
```

```
1 `asset` sd91*
```

asset_key	asset_name	asset_platform	asset_platform_description
3951416 912f79254b8f234bac78c584b3c9e9b2 10.31.21.77 52.11.193.95 ec2amaz-priceapp1 ec2-52-11-193-95.ap-southeast-1.compute.amazonaws.com ec2-52-11-193-95 www.pricingapp.domain.com i-0f88f1ee8f16eea41 ip-10-31-21-77.ap-southeast-1.compute.internal	ec2amaz-priceapp1	AWS	AWS Account: SD9100039 (Retail Price App - Prod) AWS Instance name: ec2amaz-priceapp1 AWS EC2 virtual machine instance id: i-0f88f1ee8f16eea41 AWS Public IP: 52.11.193.95

```
1 `asset` fortiOS internet-facing ``` searching for Fortinet CVE-2022-40684. ```
2 | table asset_name asset_key a*extranet_cpe a*extranet*ports
```

Last 24 hours

✓ **1 event** (10/9/22 12:00:00.000 AM to 10/10/22 12:00:00.000 AM) No Event Sampling Job Fast Mode

Events Patterns **Statistics (1)** Visualization

100 Per Page Format Preview

asset_name	asset_key	asset_seen_extranet_cpe	asset_extranet_available_ports
www.lubsdemand.de	4331916 | 19.163.16.63 | www.lubsdemand.de	cpe:/o:fortinet:fortios | cpe:2.3:o:fortinet:fortios	tcp/10443 | tcp/443

```
1 `asset` enabled-only it-managed ("exchange server" OR "CN=Exchange Servers" OR microsoft:exchange_server) ``` searching for Exchange CVE-2022-40684 ```
2 | table asset_name asset_exposure a*ad_distinguished_name a*cpe a*agent a*_is_edr_seen a*is_vulnerability_agent_seen
```

✓ **21 events** (10/11/22 12:00:00.000 AM to 10/12/22 12:00:00.000 AM) No Event Sampling

Events (21) Patterns **Statistics (21)** Visualization

100 Per Page Format Preview

asset_name	asset_exposure	asset_ad_distinguished_name	asset_seen_cpe	asset_seen_extranet_cpe
dopss103-a	extranet	unknown	cpe:/o:microsoft:windows_server_2012:r2:-:~~~standard~~~ | cpe:/a:microsoft:exchange_server | cpe:/o:microsoft:windows | cpe:2.3:a:microsoft:exchange_server | cpe:2.3:o:microsoft:windows | cpe:/a:microsoft:.net_framework:3.5:sp1 | cpe:/a:microsoft:.net_framework:3.0:sp2 | cpe:/a:microsoft:.net_framework:2.0:sp2	cpe:/a:microsoft:exchange_server | cpe:/o:microsoft:windows | cpe:2.3:a:microsoft:exchange_server | cpe:2.3:o:microsoft:windows

```
`asset` internet-facing windows server tcp/53 ``` searching for Windows DNS SIGRed CVE-2020-1350 ```
| table asset_name asset_key a*exposure a*ports a*business_unit a*owner_key
```

```
`asset` servers-only (internet-facing OR business-critical) "apache log4j" ``` searching for Log4j CVE-2021-44228. ```
| table asset_name a*exposure a*landscape asset_owner_key
```

```
`asset` windows server "ou=domain controllers" ``` searching for NetLogon CVE-2020-1472. ```
| table asset_name asset_ad_domain asset_ad_last_logon_timestamp asset_ad_distinguished_name asset_ad_memberof asset_is_aad_joined asset_last_seen
```

```
`asset` internet-facing (rdp-exposed OR smb-exposed) qualified-server it-managed
| table asset_name asset_key asset_exposure asset_extranet_available_ports asset_owner_key
```

```
`asset` qualified-workstation admin-rights credential-harvesting-available
| table asset_name asset_key asset_last_logged_on asset_last_logged_on_reusable_lsa
```

```
`asset` domain-admin-seen NOT(domain-controllers)
| table asset_name asset_key a*exposure a*is_business_critical asset_last_logged_on asset_last_logged_on_reusable_lsa
```

Asset inventory search examples

Use-case: hunting for ‘solarwinds’ software

Search by software: assets become ‘searchable’ by their installed software.

```
1  `asset` solarwinds OR (solar winds)
2  | table asset_key asset_installed_software asset_owner_key
```

Real search: this is a real search we ran when Solarwinds supply-chain attack came up. 🧐

asset_key	asset_installed_software	asset_owner_key
5106419 | 2a2b4fb9be744cb5b23c3e8e35e0a12a | 10.25.14.72 | ats-mor.addomain.local	solarwinds network automation manager | solarwinds database performance analyzer integration module | solarwinds orion quality of experience | solarwinds orion log viewer | solarwinds log analyzer | solarwinds user device tracker | solarwinds virtual infrastructure monitor | solarwinds voip and network quality manager | solarwinds server & application monitor | solarwinds orion network configuration manager | solarwinds orion netflow traffic analyzer | solarwinds netpath | solarwinds administration service | solarwinds active diagnostics | solarwinds ip address manager | solarwinds orion core services | solarwinds information service | solarwinds agent | solarwinds orion network performance monitor | solarwinds rabbitmq server | nmap npcap | solarwinds orion syslog/traps | solarwinds scp server | oracle open java development kit | microsoft monitoring agent | microsoft windows server | microsoft office access database engine | solarwinds orion network atlas | microsoft azure active directory authentication library for sql server | winpcap team winpcap | solarwinds tftp server | microsoft internet explorer | microsoft sql server management objects | microsoft sql server system clr types | microsoft windows media player | solarwinds job engine | microsoft sql server native client	alice@secret.com | bob@secret.com

Asset inventory search examples

Use-case: compromised account, where **else** is this account seen? (reverse search)
(e.g. pass-the-hash, ransomware/lateral movement attempt)

```
1 `asset` daniel.ferreira
2 | table asset_key asset_owner_key asset_last_logged_on
3 | `recover-mv`
```

asset_key	asset_owner_key	asset_last_logged_on
010b1dcc3877404a8bccf37e948d4aba 5840891 azrsrv1415 10.21.11.18 azrsrv1415.addomain.local	daniel.ferreira@secret.com	bloodhound prv.daniel.ferreira

```
1  `asset` earliest=-30d@d internet-facing qualified-servers-only missing-edr enabled-only
2  | timechart count
```

Endpoint control - Workstations (updated as 2022-10-09)

Total count

Without EDR

Unlocked (admin rights)

Useful fields

Worth adding fields to Asset Inventory

- a asset_ad_distinguished_name 1
- a asset_ad_domain 1
- a asset_ad_last_logon_timestamp 1
- a asset_ad_local_ou 1
- a asset_ad_memberof 1
- a asset_available_ports 37
- a asset_environment 1
- a asset_exposure 1
- a asset_external_dns 58
- a asset_extranet_available_ports 25
- a asset_highest_vulnerability_adjusted_risk_rating 6
- # asset_id 100+
- a asset_install_status 6
- a asset_installed_agent 47
- a asset_installed_software 24
- a asset_installed_vulnerability_kb 1
- a asset_installed_vulnerability_kb_last_update_time 1

```
| eval asset_search_term=if(match(asset_is_edr_in_scope,"true") AND match(asset_is_edr_seen,"false"), mvappend(asset_search_term, "missing-edr"), asset_search_term)
, asset_search_term=if(match(asset_ad_distinguished_name,"ou=domain controllers"), mvappend(asset_search_term, "domain-controllers", "DCs", "DCs-only"), asset_search_term)
``` ... ```
```

- a asset_is_business_critical 2
- a asset_is_business_criticality_inherited 2
- a asset_is_ci_retired_and_online 2
- a asset_is_domain_joined 2
- a asset_is_double_facing 1
- a asset_landscape 5
- a asset_last_logged_on 100+
- a asset_last_logged_on_reusable_lsa 100+
- a asset_last_reboot 100+
- a asset_is_edr_in_scope 2
- a asset_is_edr_seen 2
- a asset_is_edr_stale 2
- a asset_is_enabled 2
- a asset_is_extranet_exposed 2
- a asset_is_in_cmdb 2

- a asset_operating_system_scanner 100+
- # asset_operating_system_scanner_certainty 100+
- a asset_operating_system_source 15
- a asset_operating_system_version 100+
- a asset_owner 100+
- a asset_owner_by_action 100+
- a asset_owner_key 100+
- a asset_owner_key_ad_displayname 100+
- a asset_owner_key_description 100+
- a asset_platform 12
- a asset_platform_description 100+
- a asset_search_term 100+
- a asset_seen_certificate_cn 100+
- a asset_seen_certificate_dn 100+
- a asset_seen_cpe 100+
- a asset_seen_extranet_cpe 100+
- a asset_seen_file 100+
- a asset_service_banner 100+
- a asset_service_configuration 100+
- a asset_shared_smb 100+
- a asset_site_business_unit 6
- a asset_site_city 100+
- a asset_site_code 100+
- a asset_site_contact 100+
- a asset_site_country 100+

**CREATE YOUR OWN SHODAN BY ENFORCING PRESENCE OF EASY SEARCH TERMS ON EACH ASSET!**

**ALL FILES AND FOLDERS ARE NOW SEARCHABLE!**

splunk> turn data into doing
```

CREATE YOUR OWN `asset` AND BECOME THE NEXT

Thank You

splunk> turn data into doing™