# SecKit IDM Windows Documentation

Ryan Faircloth/Splunk Inc.

Jan 14, 2020

# Contents

Version:

Release:

Success Enablement Content “SecKit” apps for Splunk are designed to accelerate the tedious or difficult tasks. This application IDM Windows is an add on for Splunk Enterprise Security designed to identify and enrich asset and identity information based on Microsoft Active Directory. Assets and Identities based on Active Directory can give critical insights into machine data

- Is this device properly domain joined?
- Who is responsible for this device?
- Does this user have a privileged account?

# CHAPTER 1

## Before you get started

- Implement v3.0.0 or newer of SecKit IDM Windows Common.
- Complete Splunk Enterprise Security Administration training
- Review the current Assets and Identities section of the Administration Manual
- **Review the use of lookup data in Splunk**
  - Lookup Command
  - CIDR and Matching Rules

# CHAPTER 2

## Support

- Reporting issues or requesting enhancements Issue Tracker
- Source

CHAPTER 3

# Documentation

## 3.1 Splunk System Requirements

### 3.1.1 Mandatory

- Splunk Enterprise >7.1.0
- Splunk Enterprise Security >5.1.0
- SecKit IDM Common >=3.0.0
- Splunk TA Windows >=5.0.1

## 3.2 Installation

Installation of the apps is intended to be minimally impactful to a Splunk ES environment. If existing assets and identities have been configured care should be take to ensure an asset or identity is only defined once.

### 3.2.1 Migration from legacy assets and identities

Enterprise Security does not “merge” records from multiple sources having multiple conflicting definitions can impact systems active users.

- Identify and remove the identity file definition from Splunk Enterprise Security
- Identify and remove the lookup definition from Splunk Enterprise
- Identify and remove the lookup file from disk. *Important to ensure large bundles do not impact search replication*

### 3.2.2 Installation

This add on is installed on the Splunk Enterprise Security Search head.

Splunk Enterprise:

- First Install SecKit IDM Common https://splunkbase.splunk.com/app/3055/
- Download the latest published release of SecKit Windows Assets Add-on for Splunk Enterprise Security.
- Download the latest master build of SecKit SA IDM Windows from bitbucket
- See installing apps This add on only requires installation on the search head in a distributed deployment.

Splunk Cloud:

- Using a service request ask for the app installation SecKit_SA_idm_windows id "3059" specify version 3.0 or later

### 3.2.3 Verify Installation

*Verification*

- As an es_admin navigate to Splunk Enterprise Security
- Click the Search menu
- Click Search again
- Execute the search:

```
| inputlookup seckit_idm_windows_os_lookup
```

*verify results are returned.*

- Execute the search:

```
| inputlookup seckit_idm_windows_assets_lookup
```

*verify no error is report results may yield zero records.*

- Execute the search:

```
| inputlookup seckit_idm_windows_assets_identities_lookup
```

*verify no error is report results may yield zero records.*

- Execute the search:

```
| inputlookup seckit_idm_windows_identities_lookup
```

*verify no error is report results may yield zero records.*

### 3.2.4 Initialize Lookups and Collections

- Navigate to a Splunk Search window

Run the following searches in order note the count of records may be zero continue following the installation documentation to complete the process.

#### Windows Identities

**Build:**

```
| from savedsearch:seckit_idm_windows_identities_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_identities_lookup
```

#### Windows Assets

**Build:**

```
| from savedsearch:seckit_idm_windows_assets_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_assets_lookup
```

#### Windows Computer Identities

**Build:**

```
| from savedsearch:seckit_idm_windows_assets_identities_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_assets_identities_lookup
```

#### Force Enterprise Asset and Identity Merge

**Run the search:**

```
| from savedsearch:"Identity - Asset String Matches - Lookup Gen"
```

**Run the search:**

```
| from savedsearch:"Identity - Asset CIDR Matches - Lookup Gen"
```

**Run the search:**

```
| from savedsearch:"Identity - Identity Matches - Lookup Gen"
```

Continue to *Collect Active Directory Data*

## 3.3 Upgrading from prior versions

### 3.3.1 Upgrade from version <2.0

Changes from version 1.0 are drastic, recommendation is to remove the apps, and review the contents of `local/*.conf` and lookups and port the config as if a new installation.

### 3.3.2 Upgrade from version 2.x

The macro get_manager has been renamed to seckit_idm_windows_get_identity_manager if this has been customized move the customizations to the new macro

## 3.4 Collect Active Directory Data

Collection of Active Directory data from all domains in all forests is required for the correct use of this app. In many instances the collection is already occurring but the records may not be retained as needed.

### 3.4.1 Determine if events have been collected

Execute the following search to determine if any previous collect has occurred:

```
index=* sourcetype=ActiveDirectory dNSHostName=*
| rex field=dNSHostName "\.(?<domain>.*)"
| stats values(index) dc(dNSHostName) as number_computers earliest(_time) as first_
↪event latest(_time) as last_event dc(host) as Number_DCs by domain
```

If the number of computers is similar to the expected number of computers by domain and the number of DCs is <4 the existing data can be used. If no results are returned or the results are in question continue to collecting ActiveDirectory Events

### 3.4.2 Collecting Active Directory Events

Splunk `sourcetype=ActiveDirectory` events are generated when a object change or replication of an object change occurs. SecKit typicaly utilizes the index `appmsadmon` for these events if another index or multiple indexes will be used ensure the search macro `seckit_idm_windows_adindex` is properly updated to reference the indexes.

SecKit Windows contains prescriptive guidance for collection of Windows data available here https://seckit.readthedocs.io/projects/splunk-ta-windows-seckit/en/latest/index.html If the entire SecKit Windows guidance is not used ensure the following activities are performed.

- Implement the Universal Forwarder managed by a deployment server or alternative configuration management solution for all Active Directory Servers in all domains.
- Ensure WinEventLog:Security with `renderxml=true` is collected on all Active Directory servers using Splunk_TA_windows v5.0.0 or later
- Create a new index `appsmsadmon` or name selected based on internal standards exists on the indexer cluster configured for at least 90 days of storage.
- Using deployment server (or other config management solution) place the following config stanza on two selected Active Directory servers per domain. ** DO NOT DEPLOY TO ALL SERVERS ** Update the index as required and remove any other `admon://` stanzas from previous configuration.

```
[admon://seckit]
disabled = 0
baseline = 1
monitorSubtree = 1
index=appmsadmon
```

Verify by repeating the steps to determine if events have been collected for all domains.

### 3.4.3 Load Object Collections

SecKit IDM uses collections to cache Active Directory objects for later use.

- Run the search:

```
|  from savedsearch:seckit_idm_windows_activedirectory_computers_load
```

This will build the initial cache of computer objects the number returned as a result should represent the number of computers in the domain.

- Run the search:

```
|  from avedsearch:seckit_idm_windows_activedirectory_persons_load
```

This will build the initial cache of user objects the number returned as a result should represent the number of users in the domain.

### 3.4.4 Build initial asset and identity lookups

If a new domain is on boarded, the normally scheduled searches will consolidate and merge the information within 24 hours. To expedite the process this can be safely repeated.

Run the following searches in order:

#### Windows Identities

**Build:**

```
| from savedsearch:seckit_idm_windows_identities_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_identities_lookup
```

#### Windows Assets

**Build :**

```
| from savedsearch:seckit_idm_windows_assets_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_assets_lookup
```

#### Windows Computer Identities

**Build:**

```
| from savedsearch:seckit_idm_windows_assets_identities_lookup_gen
```

**Verification:**

```
| inputlookup seckit_idm_windows_assets_identities_lookup
```

### Force Enterprise Asset and Identity Merge

**Run the search:**

```
| from savedsearch:"Identity - Asset String Matches - Lookup Gen"
```

**Run the search:**

```
| from savedsearch:"Identity - Asset CIDR Matches - Lookup Gen"
```

**Run the search:**

```
| from savedsearch:"Identity - Identity Matches - Lookup Gen"
```

## 3.5 Quick Start Tutorial

The quick start procedure is simply to demonstrate the application of this solution continue reading in the using guide once your first use is complete. If you have not already done so follow the install and data-activedirectory portions of the documentation first.

### 3.5.1 Identifying important assets and identities

- Working with a knowledgeable identify a specific Windows server (or servers) that by name or name pattern can be identified as high or critical. For example the file server used by the CEO.
- Working with a knowledgeable Active Directory administrator identify one group with current members that grants privileged access to Active Directory or Member Servers OTHER than the default groups created by Active Directory such as Enterprise Admins, Domain Admins or Administrators

### 3.5.2 Configure categorization for a single server

- Login to Enterprise Security
- Navigate to Enterprise security
- Select Configure menu
- Select Content management
- Select "SecKit SA IDM Common for ES" in the app drop down
- Find "SecKit IDM Common static hosts"
- Under actions for this row select export
- Using a csv editor of your choice Add the following information and save. wild card values for static name are allowed as are specific host names. Wild cards should use should be limited to the end of the host name to avoid accidental match to unintended hosts.

| static_name | static_category | static_pci_domain | static_priority | static_expected |
|---|---|---|---|---|
| srvexevfs* | estaff | trust | high | true |

- return to Enterprise Security
- Under Actions for this row click Update File
- Select the modified file

**Run the following searches:**

**Windows Assets:**

Run the search:

```
| savedsearch seckit_idm_windows_identities_lookup_gen
```

**Force Enterprise Identity Merge:**

- Run the search:

```
| from savedsearch:"Identity - Asset Identity Matches - Lookup Gen"
```

### 3.5.3 Verify categorization for a single server

- Return to Enterprise Security
- Select the Security Domains menu
- Select Identity
- Select Asset Center
- Enter a specific host matching static name above
- Verify the category, pci_domain and priority fields match above

### 3.5.4 Configure categorization for privileged group

- Login to Enterprise Security
- Navigate to Enterprise security
- Select Configure menu
- Select Content management
- Select "SecKit SA IDM Windows for ES" in the app drop down
- Find "SecKit IDM Windows AD Identity Classification By memberOf Group"
- Under actions for this row select export
- Using a csv editor of your choice Add the following information and save. wild card values for static name are allowed as are specific host names. Wild cards should use should be limited to the end of the host name to avoid accidental match to unintended hosts.

| memberOf | member_category | member_priority | member_watchlist |
|---|---|---|---|
| CN=System Managed Accounts Group,CN=Builtin,* | nha | critical | high |

- return to Enterprise Security
- Under Actions for this row click Update File

- Select the modified file

**Run the following searches:**

**Windows Identities**

Build:

```
| savedsearch seckit_idm_windows_assets_lookup_gen
```

**Force Enterprise Asset and Identity Merge:**

- Run the search:

```
| from savedsearch:"Identity - Asset String Matches - Lookup Gen"
```

- Run the search:

```
| from savedsearch:"Identity - Asset CIDR Matches - Lookup Gen"
```

### 3.5.5 Verify categorization for privileged group

- Return to Enterprise Security
- Select the Security Domains menu
- Select Identity
- Select Identity Center
- Enter a specific user included in the group
- Verify the category and priority fields match above

## 3.6 Using Windows Assets and Identities

Before continuing with this section ensure you have completed the quickstart tutorial.

### 3.6.1 Enrichment Lookups

#### seckit_idm_windows_identities_accounts_lookup

Utilizes the sAMAccountName field of the ActiveDirectory sync event to enrich the record. This is often used to identify privileged and service accounts.

##### account

Case insensitive and wild card match.

##### account_category

Additional categories to apply

#### account_priority

Priority to apply note the highest priority is utilized

#### account_watchlist

Include in watchlist this should only be applied to accounts which should not be seen in events.

#### seckit_idm_windows_identities_bunit_lookup

Utilizes the calculated bunit field of the ActiveDirectory sync event to enrich the record. This is often used to identify users with access to sensitive data. See macro for bunit calculation.

#### bunit

Case insensitive and wild card match.

#### bunit_category

Additional categories to apply

#### bunit_priority

Priority to apply note the highest priority is utilized

#### seckit_idm_windows_identities_members_lookup

Utilizes the sAMAccountName field of the ActiveDirectory sync event to enrich the record. This is often used to identify privileged and service accounts.

#### account

Case insensitive and wild card match.

#### account_category

Additional categories to apply

#### account_priority

Priority to apply note the highest priority is utilized

#### account_watchlist

Include in watchlist this should only be applied to accounts which should not be seen in events.

### seckit_idm_windows_identities_bunit_lookup

Utilizes the memberOf field of the ActiveDirectory sync event to enrich the record. This is often used to identify users with access to sensitive data or privileged.

#### memberOf

Case insensitive and wild card match.

#### member_category

Additional categories to apply

#### member_priority

Priority to apply note the highest priority is utilized

#### member_watchlist

Priority to apply note the highest priority is utilized

### seckit_idm_windows_identities_nha_lookup

Utilizes the sAMAccountName field of the ActiveDirectory sync event to enrich the record. This is often used to identify privileged and service accounts. This list works exactly as the account lookup. It is provided as a separate lookup to help change management where the contents are subject to audit review.

#### identity

Case insensitive and wild card match.

#### nha_category

Additional categories to apply

#### nha_priority

Priority to apply note the highest priority is utilized

#### nha_watchlist

Include in watchlist this should only be applied to accounts which should not be seen in events.

#### seckit_idm_windows_identities_org_lookup

Utilizes the organizational unit to enrich identities

##### org

Case insensitive and wild card match.

##### org_category

Additional categories to apply

##### org_priority

Priority to apply note the highest priority is utilized

#### seckit_idm_windows_identities_title_lookup

Utilizes the person record's job title to enrich identities. Typically this is utilized to identify important persons and persons which may have access to sensitive information such as attorneys and execs.

##### title

Case insensitive and wild card match.

##### title_category

Additional categories to apply

##### title_priority

Priority to apply note the highest priority is utilized

##### title_watchlist

Include in watchlist this should only be applied to accounts which should not be seen in events.

#### seckit_idm_windows_assets_bunit_lookup

Utilizes the calculated bunit field of the ActiveDirectory sync event to enrich the record. This is often used to usage of assets. See Macro . . . .

##### bunit

Case insensitive and wild card match.

#### bunit_category

Additional categories to apply

#### bunit_priority

Priority to apply note the highest priority is utilized

#### bunit_expected

Identify this asset as expected for ES

### seckit_idm_windows_os_lookup

The lookup enriches the asset object with operating system information and support status. Changes to this lookup should be submitted via issue or PR in the project repository.

### seckit_idm_windows_os_win_interfaces_lookup

The lookup enriches optional data to detect the ip of assets where the interface is identified as dhcp assigned however the ip is effectively static as it is assigned by a virtualization management solution. Changes to this lookup should be submitted via issue or PR in the project repository.

### seckit_idm_windows_pf_roles_lookup

This lookip enriches the category of assets based on the installed roles as collected by Splunk_TA_windows.

#### role_name

Case insensitive and wild card match.

#### role_category

Additional categories to apply

#### role_priority

Priority to apply note the highest priority is utilized

#### role_expected

Identify this asset as expected for ES

#### role_pci_domain

Frequently used to define systems whose roles such as Active Directory services Mandate inclusion in the trust domain for PCI.

#### seckit_idm_windows_assets_org_lookup

This lookup utilizes the organizational unit of the asset.

#### org

Case insensitive and wild card match.

#### org_category

Additional categories to apply

#### org_priority

Priority to apply note the highest priority is utilized

#### org_expected

Identify this asset as expected for ES

### 3.6.2 Apply the updated configuration to your assets

*Update the configuration files using Enterprise Security Content Management*

- As a es_admin login to Splunk Enterprise Security
- Navigate to the configure menu
- Select Content Management
- Select "SecKit SA IDM Common" from the app menu
- Find "SecKit IDM Common network location" by name and click update file upload the file created above `seckit_idm_pre_cidr_location.csv`
- Find "SecKit IDM Common network categories" by name and click update file upload the file created above `seckit_idm_pre_cidr_category.csv`

*Force Merge of Assets*

The following process can be used at any time to force immediate updates of asset files

- Navigate to a Splunk Search window
- Run the search:

```
| savedsearch "seckit_idm_common_assets_networks_lookup_gen"
```

- Run the search:

```
| from savedsearch:"Identity - Asset String Matches - Lookup Gen"
```

- Run the search:

```
| from savedsearch:"Identity - Identity Matches - Lookup Gen"
```

*Verification*

- As an ES user (or above) navigate to Enterprise security
- Select Security Domains from the menu
- Select Identity from the drop down
- Select Asset Center
- View the record as defined above if additional records are displayed from other sources sort/scroll to locate

## 3.6.3 Scheduled Searches and Enabled Input Tasks

### Inputs

#### identity_manager://seckit_idm_common_assets_networks

Utilized to enable the usage of the main combined lookup by Enterprise Security Identity Manager

### Scheduled Searches

#### seckit_idm_common_assets_networks_lookup_gen

Produces the lookup `seckit_idm_common_assets_networks_lookup` used as input in `identity_manager://seckit_idm_common_assets_networks`. The default schedule will produce a new lookup every 4 hours.

#### seckit_idm_combined_cidr_category_by_str_lookup_gen

Combines the csv lookup `seckit_idm_pre_cidr_category_by_str_lookup` and search managed collection `seckit_idm_common_event_cidr_category` to produced the lookup `seckit_idm_combined_cidr_category_by_str_lookup`. This lookup is utilized by the saved search `seckit_idm_common_assets_networks_lookup_gen` to produce the network assets file. The default schedule will produce a new file every 30 min.

#### seckit_idm_common_event_cidr_category_from_dm_network_session_dhcp

Utilizes the network session data model to identify network segments managed using DHCP to automatically categorize subnets. The default schedule will detect new subnets every 4 hours.

#### seckit_idm_common_event_cidr_category_age

Ages entries in the lookup `seckit_idm_common_event_cidr_category_age` where last is non zero and not updated in the prior year. The default schedule will trim the lookup once per day

#### seckit_idm_common_assets_expected_tracker_gen

Updates the lookup `seckit_idm_common_assets_host_expected_tracker_lookup` based on universal forwarder internal logs to identify hosts which should be set as is_expected. The default schedule search runs at the top of the hour using only the last 15m of prior data.

#### seckit_idm_common_assets_expected_tracker_age

Ages entries in the lookup `seckit_idm_common_assets_host_expected_tracker_lookup` not updated in the prior year. The default schedule will trim the lookup once per day.

#### seckit_idm_pre_cidr_category_by_str_lookup_ftr

Ensures the lookup `seckit_idm_pre_cidr_category_by_str_lookup` exists and contains the correct fields. The default schedule of the search uses a special configuration option run_on_startup and run_n_times to ensure the search runs on only once.

#### seckit_idm_common_assets_networks_lookup_ftr

Ensures the lookup `seckit_idm_common_assets_networks_lookup` exists and contains the correct fields. The default schedule of the search uses a special configuration option run_on_startup and run_n_times to ensure the search runs on only once.

#### seckit_idm_pre_host_static_lookup_ftr

Ensures the lookup `seckit_idm_pre_host_static_lookup` exists and contains the correct fields. The default schedule of the search uses a special configuration option run_on_startup and run_n_times to ensure the search runs on only once.

### 3.6.4 Scheduled Searches and Enabled Input Tasks

#### Inputs

#### identity_manager://seckit_idm_windows_assets

Utilized to enable the usage of the assets discovered from ActiveDirectory lookup by Enterprise Security Identity Manager

#### identity_manager://seckit_idm_windows_assets_identities

Utilized to enable the usage of the identities discovered from ActiveDirectory for computer accounts lookup by Enterprise Security Identity Manager

#### identity_manager://seckit_idm_windows_identities

Utilized to enable the usage of the identities discovered from ActiveDirectory lookup by Enterprise Security Identity Manager.

### Scheduled Searches

#### seckit_idm_windows_assets_identities_lookup_gen

Produces the lookup `seckit_idm_windows_assets_identities_lookup` used as input in `identity_manager://seckit_idm_windows_assets_identities`. The default schedule will produce a new lookup every 4 hours.

#### seckit_idm_windows_assets_lookup_gen

Produces the lookup `seckit_idm_windows_assets_lookup` used as input in `identity_manager://seckit_idm_windows_assets`. The default schedule will produce a new lookup every 4 hours.

#### seckit_idm_windows_identities_lookup_gen

Produces the lookup `seckit_idm_windows_identities_lookup` used as input in `identity_manager://seckit_idm_windows_identities`. The default schedule will produce a new lookup every 4 hours.

#### seckit_idm_windows_pf_roles_lookup_gen

Manages the collection lookup `seckit_idm_windows_asset_os_role_lookup` The default schedule will update the collection hourly.

#### seckit_idm_windows_activedirectory_computers_load

Manually executed to load the collection using historical events or reload collection (unusual). This is not scheduled and should not be under normal conditions.

#### seckit_idm_windows_activedirectory_computers_tracker

Manages the collection lookup `seckit_idm_windows_activedirectory_computers_lookup` The default schedule will update every 5 minutes using the prior five minutes.

#### seckit_idm_windows_activedirectory_people_load

Manually executed to load the collection using historical events or reload collection (unusual). This is not scheduled and should not be under normal conditions.

#### seckit_idm_windows_activedirectory_persons_tracker

Manages the collection lookup `seckit_idm_windows_activedirectory_persons_lookup` The default schedule will update every 5 minutes using the prior five minutes.

#### seckit_idm_windows_asset_interface_load

Manually executed to load the collection using historical events or reload collection (unusual). This is not scheduled and should not be under normal conditions.

#### seckit_idm_windows_asset_interface_tracker

Manages the collection lookup `seckit_idm_windows_asset_interface_lookup` The default schedule will update every 5 minutes using the prior five minutes.

#### seckit_idm_windows_asset_os_role_load

Manually executed to load the collection using historical events or reload collection (unusual). This is not scheduled and should not be under normal conditions.

#### seckit_idm_windows_asset_os_role_tracker

Manages the collection lookup `seckit_idm_windows_asset_os_role_lookup` The default schedule will update every 5 minutes using the prior five minutes.

## 3.7 Customizing the add on

The add on support customization using macros

### 3.7.1 Index Location Macros

- seckit_idm_windows_adindex The index(es) where sourcetype=ActiveDirectory can be found
- seckit_idm_windows_winhostmon_adindex The index(es) where sourcetype=WinHostMon can be found
- seckit_idm_windows_winscripts_adindex The index(es) where scripted inputs for the Windows OS can be found this is used to identify static IP addresses
- seckit_idm_windows_winevents_adindex The index(es) where Windows Security events can be located
- seckit_idm_windows_nixscripts_adindex The index(es) where scripted inputs for the Nix OS can be found this is used to identify static IP addresses for domain joined Nix assets

### 3.7.2 seckit_idm_windows_get_asset_bunit

This macro is utilized to customize the format of the bunit field for ES Assets it should set the bunit field.

### 3.7.3 seckit_idm_windows_get_asset_owner

This macro is utilized to customize the owner field for the asset this should set the value of owner

### 3.7.4 seckit_idm_windows_get_asset_priority

This macro allows customization of the priority field for the asset this should the field custom_priority

### 3.7.5 seckit_idm_windows_get_asset_category

This macro allows customization of the category for the asset it should set/build the value of the mvfield custom_category

### 3.7.6 seckit_idm_windows_get_identity_bunit

This macro is utilized to customize the format of the bunit field for ES Identity it should set the bunit field.

### 3.7.7 seckit_idm_windows_get_identity_priority

This macro is set used to set the identity priority is should set a field value of custom_priority

### 3.7.8 seckit_idm_windows_get_identity_manager

This macro is utilized to set the manager of the identity record generally this is in the form of the managers username or primary email. The field should set the field manager

### 3.7.9 seckit_idm_windows_get_asset_ip_custom

This macro is set used to set the identity priority is should set a field value of ip_custom which can be multivalue. Do not set the ip value for hosts which are DHCP managed.

### 3.7.10 seckit_idm_windows_get_asset_mac_custom

This macro is set used to set the asset MAC field is should set a field value of mac_custom which can be multivalue.

## 3.8 Windows Categories

The following categories are commonly defined in the categories configuration. The shortest reasonable string should be used for all values. Note only values matching the regex [A-Za-z0-9-_] should be used.

### 3.8.1 pf:<value> BOTH

The PF or primary function of a device is a specific identifier relates to the role of a asset in a service. This is most commonly applied to a specific asset but may apply to a CIDR

### 3.8.2 svc:<value> BOTH

The SVC or Service is a identifier indicating the service this asset participates in providing for example. The service DNS “svc:DNS” would typically be made up of a combination of “pf:ms_dns” or “pf:BIND” AND “pf:dns_rbl” “pf:dns_recursive”

## 3.9 Auditing

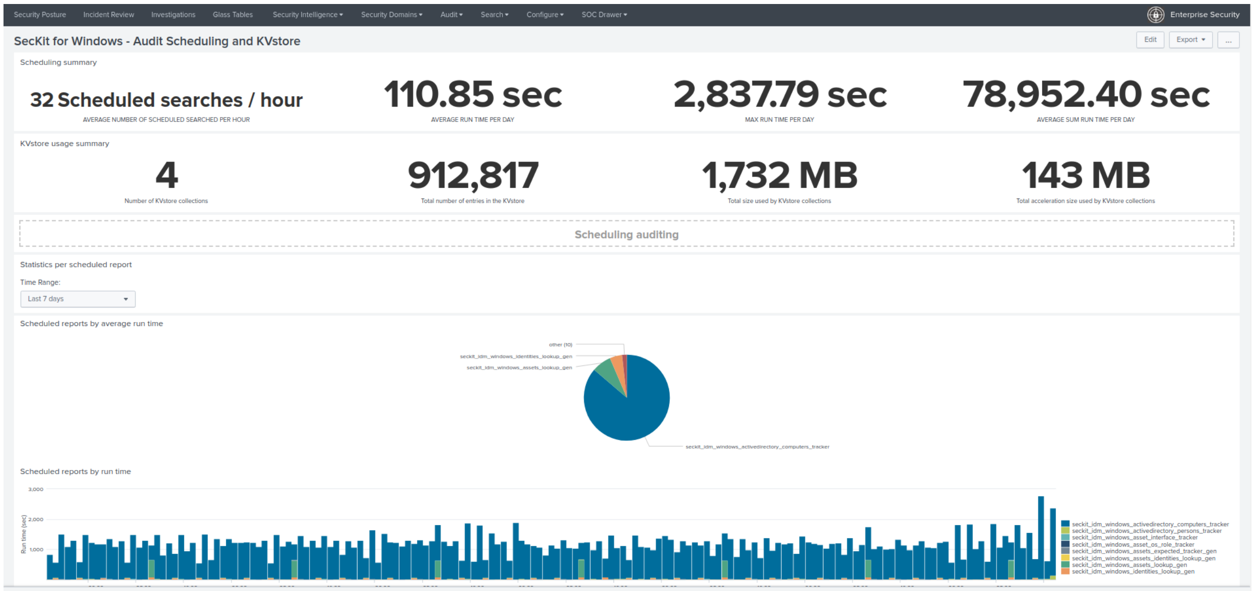
### 3.9.1 Scheduling and KVstore audit

**The SecKit SA IDM for Windows assets provides out of the box an audit dashboard for scheduling activity and KVstore collections:**

- **SecKit for Windows - Audit Scheduling and KVstore collections**

**This dashboard is available in any application and provides key statistics and visualizations on top of:**

- Main aggregated statistics for scheduling: average number of scheduled searches, run time statistics, etc
- Main aggregated statistics for KVstore collections: number of collections, cumulated size of collections and accelerations, etc
- Over time visualization of scheduled reports activity
- Most time expensive scheduled reports, frequency
- Details table of scheduled reports with statistics and configuration
- Details table of KVstore collections hosted in the application



SecKit for Windows - Audit Scheduling and KVstore

Scheduling summary

32 Scheduled searches / hour
AVERAGE NUMBER OF SCHEDULED SEARCHED PER HOUR

110.85 sec
AVERAGE RUN TIME PER DAY

2,837.79 sec
MAX RUN TIME PER DAY

78,952.40 sec
AVERAGE SUM RUN TIME PER DAY

KVstore usage summary

4
Number of KVstore collections

912,817
Total number of entries in the KVstore

1,732 MB
Total size used by KVstore collections

143 MB
Total acceleration size used by KVstore collections

Scheduling auditing

Statistics per scheduled report

Time Range: Last 7 days ▾

Scheduled reports by average run time

other (10)
seckit_idm_windows_identities_lookup_gen
seckit_idm_windows_assets_lookup_gen
seckit_idm_windows_activedirectory_computers_tracker

Scheduled reports by run time

3,000
2,000
1,000
Run time (sec)
seckit_idm_windows_activedirectory_computers_tracker
seckit_idm_windows_activedirectory_persons_tracker
seckit_idm_windows_asset_interface_tracker
seckit_idm_windows_asset_os_role_tracker
seckit_idm_windows_assets_expected_tracker_gen
seckit_idm_windows_assets_identities_lookup_gen
seckit_idm_windows_assets_lookup_gen
seckit_idm_windows_identities_lookup_gen