Use Security Analytics local event source assets database for enrichment with multi-indexed feed

Miha Mesojedec, System Engineer EMEA EE

Miha.Mesojedec@rsa.com

Table of Contents

1. Overview

RSA Security Analytics has built-in Event Source Management (ESM) capability which provides an easy way to manage event sources and configure alerting policies for your event sources.

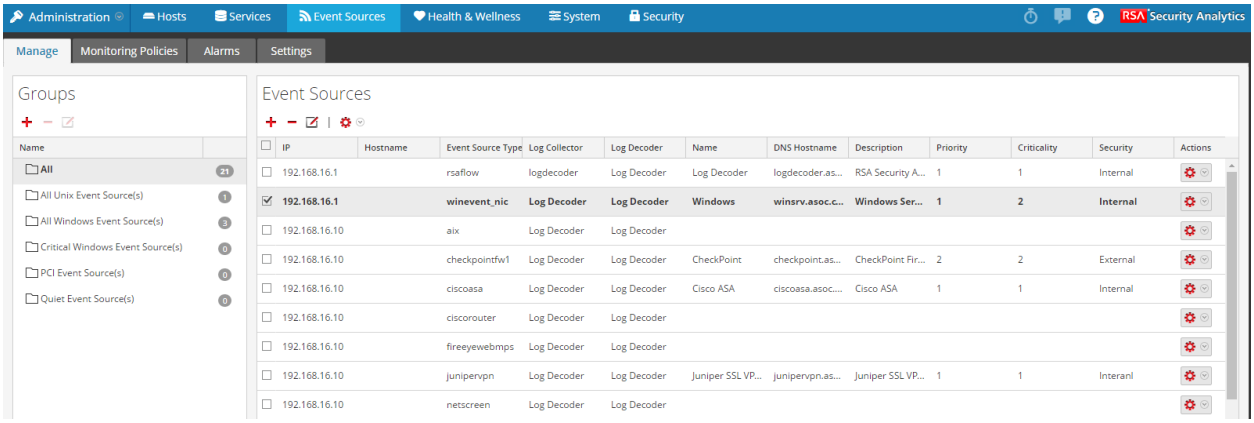
More details about ESM: http://sadocs.emc.com/0_en-us/088_SA106/90_EventSM

Unfortunately ESM module has limitation and can't be used for Reporting and RBAC.

This guide will explain how you can use event source attributes as Multi-indexed feed and later use it for Reporting and RBAC.

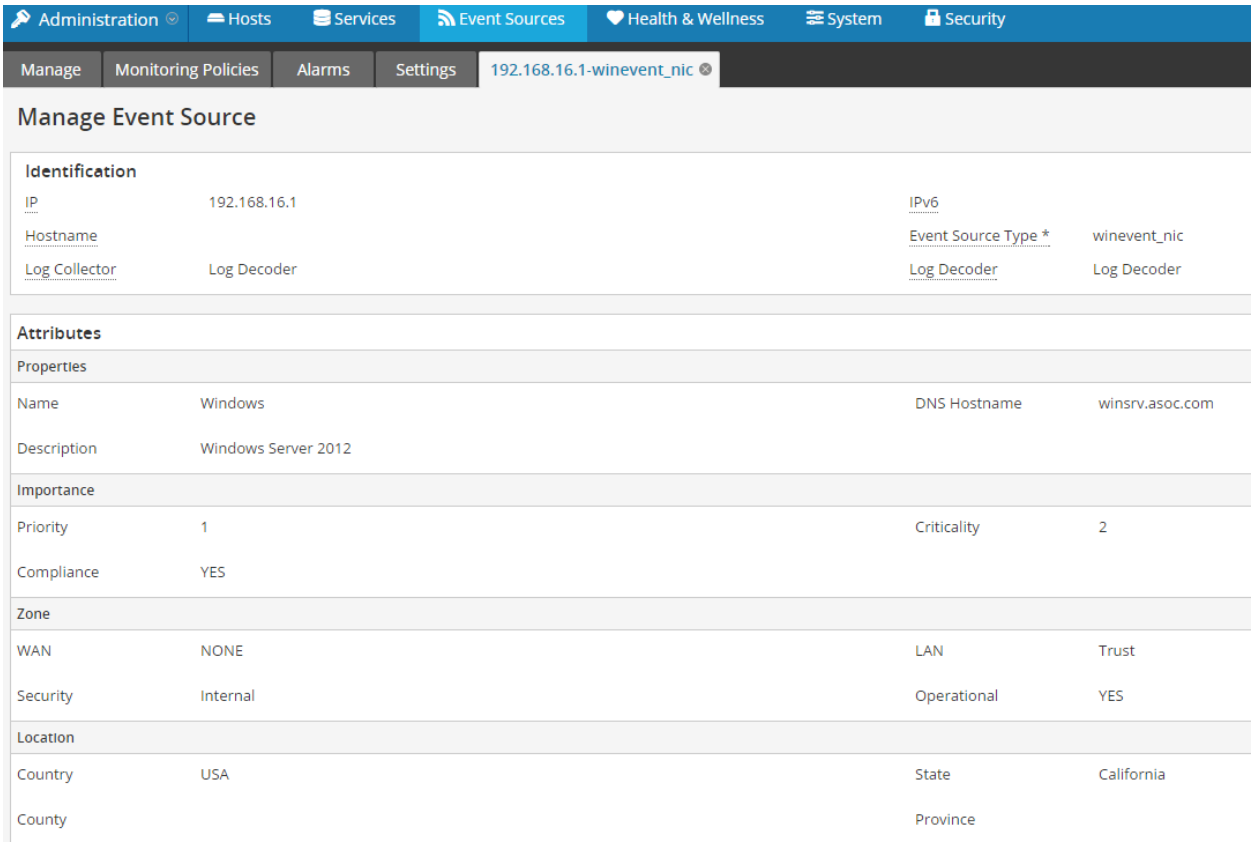
2. Adding/Editing Event Source Attributes

Event sources attributes can be added to event source manually or imported through SA UI.



Documentation explains all possible attributes categories.

http://sadocs.emc.com/0_en-us/088_SA106/90_EventSM/30_Ref/zzEvtSrcDetTb



3. Automatically exporting event source attributes details

In documentation you can find Export option from where you can manually create CSV file of all your event source assets.

Goal is to automatically export ESM attributes which are stored in Mongo database and save it to CSV file, which can be later used for Multi-indexed feed.

ESM attributes are saved in Security Analytics server Mongo "esm" database and with collection name "eventsources".

To see all stored ESM attributes you can run following command in your SA server (below output example).

```
[root@saserver ~]# echo 'db.eventsources.find()' | mongo esm
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: esm
{ "_id" : "192.168.16.1-rsaflow", "_class" : "EsmEventSource", "attributes" : { "asoc-es-security" :
"Internal", "asoc-es-country" : "USA", "asoc-es-bu" : "IT", "asoc-es-type" : "rsaflow", "asoc-es-
criticality" : NumberLong(1), "asoc-es-lan" : "Trust", "asoc-es-vendor" : "RSA", "asoc-es-division" :
"RSA", "asoc-es-city" : "Boston", "asoc-es-wan" : "None", "asoc-es-operational" : "YES", "asoc-es-ip"
: "192.168.16.1", "asoc-es-color" : "black", "asoc-es-contact" : "John Black", "asoc-es-compliance" :
```

"YES", "asoc-es-upsProtected" : true, "asoc-es-role" : "Log Collection", "asoc-es-logDecoder" : "Log Decoder", "asoc-es-voltage" : NumberLong(220), "asoc-es-room" : "2018", "asoc-es-department" : "Cyber Defence", "asoc-es-desc" : "RSA Security Analytics Log Decoder", "asoc-es-assetTag" : "3333333333", "asoc-es-logDecoderUuid" : "f07c11a2-6136-4d8b-bec3-a9a7e283e536", "asoc-es-manager" : "Peter White", "asoc-es-domain" : "ASOC.COM", "asoc-es-dnsHostname" : "logdecoder.asoc.com", "asoc-es-backupAdmin" : "Tim Short", "asoc-es-custom1" : "Critical", "asoc-es-logCollector" : "logdecoder", "asoc-es-company" : "EMC", "asoc-es-postalCode" : "01207", "asoc-es-primaryAdmin" : "John Black", "asoc-es-priority" : NumberLong(1), "asoc-es-name" : "Log Decoder", "asoc-es-building" : "RSA", "asoc-es-uid" : "192.168.16.1-rsaflow", "asoc-es-rackHeight" : NumberLong(1), "asoc-es-state" : "Massachusetts", "asoc-es-floor" : "4th", "asoc-es-email" : "john.black@asoc.com", "asoc-es-serialNumber" : "888888888", "asoc-es-systemName" : "Log Decoder", "asoc-es-logCollectorUuid" : "logdecoder", "asoc-es-systemDesc" : "RSA Security Analytics Log Decoder" }, "searchAttributes" : { "asoc-es-security" : "INTERNAL", "asoc-es-country" : "USA", "asoc-es-bu" : "IT", "asoc-es-type" : "RSAFLOW", "asoc-es-criticality" : NumberLong(1), "asoc-es-lan" : "TRUST", "asoc-es-vendor" : "RSA", "asoc-es-division" : "RSA", "asoc-es-city" : "BOSTON", "asoc-es-wan" : "NONE", "asoc-es-operational" : "YES", "asoc-es-ip" : NumberLong("3232239617"), "asoc-es-color" : "BLACK", "asoc-es-contact" : "JOHN BLACK", "asoc-es-compliance" : "YES", "asoc-es-upsProtected" : true, "asoc-es-role" : "LOG COLLECTION", "asoc-es-logDecoder" : "LOG DECODER", "asoc-es-voltage" : NumberLong(220), "asoc-es-room" : "2018", "asoc-es-department" : "CYBER DEFENCE", "asoc-es-desc" : "RSA SECURITY ANALYTICS LOG DECODER", "asoc-es-assetTag" : "3333333333", "asoc-es-logDecoderUuid" : "F07C11A2-6136-4D8B-BEC3-A9A7E283E536", "asoc-es-manager" : "PETER WHITE", "asoc-es-domain" : "ASOC.COM", "asoc-es-dnsHostname" : "LOGDECODER.ASOC.COM", "asoc-es-backupAdmin" : "TIM SHORT", "asoc-es-custom1" : "CRITICAL", "asoc-es-logCollector" : "LOGDECODER", "asoc-es-company" : "EMC", "asoc-es-postalCode" : "01207", "asoc-es-primaryAdmin" : "JOHN BLACK", "asoc-es-priority" : NumberLong(1), "asoc-es-name" : "LOG DECODER", "asoc-es-building" : "RSA", "asoc-es-uid" : "192.168.16.1-RSAFLOW", "asoc-es-rackHeight" : NumberLong(1), "asoc-es-state" : "MASSACHUSETTS", "asoc-es-floor" : "4TH", "asoc-es-email" : "JOHN.BLACK@ASOC.COM", "asoc-es-serialNumber" : "888888888", "asoc-es-systemName" : "LOG DECODER", "asoc-es-logCollectorUuid" : "LOGDECODER", "asoc-es-systemDesc" : "RSA SECURITY ANALYTICS LOG DECODER" } }

To store ESM attributes in CSV file you can use “mongoexport” option in SA server.

```
[root@saserver ~]# mongoexport
connected to: 127.0.0.1
no collection specified!
Export MongoDB data to CSV, TSV or JSON files.
options:
  --help                                produce help message
  -v [ --verbose ]                      be more verbose (include multiple times
                                        for more verbosity e.g. -vvvvv)
  --version                             print the program's version and exit
  -h [ --host ] arg                     mongo host to connect to ( <set
                                        name>/s1,s2 for sets)
  --port arg                            server port. Can also use --host
                                        hostname:port
  --ipv6                                enable IPv6 support (disabled by
                                        default)
  -u [ --username ] arg                 username
```

```
  -p [ --password ] arg                 password
  --authenticationDatabase arg          user source (defaults to dbname)
  --authenticationMechanism arg (=MONGODB-CR)
                                        authentication mechanism
  --dbpath arg                          directly access mongod database files
                                        in the given path, instead of
                                        connecting to a mongod  server - needs
                                        to lock the data directory, so cannot
                                        be used if a mongod is currently
                                        accessing the same path
  --directoryperdb                      each db is in a separate directly
                                        (relevant only if dbpath specified)
  -d [ --db ] arg                       database to use
  -c [ --collection ] arg               collection to use (some commands)
  -f [ --fields ] arg                   comma separated list of field names
                                        e.g. -f name,age
  --fieldFile arg                       file with fields names - 1 per line
  -q [ --query ] arg                    query filter, as a JSON string
  --csv                                 export to csv instead of json
  -o [ --out ] arg                      output file; if not specified, stdout
                                        is used
  --jsonArray                           output to a json array rather than one
                                        object per line
  -k [ --slaveOk ] arg (=1)             use secondaries for export if
                                        available, default true
  --forceTableScan                      deprecated
```

With “mongoexport” option you can filter necessary attributes and save it into CSV file.

Example below is representing ESM attributes you can use in your filter and output to CSV file.

```
{
	"_id": "192.168.16.1-rsaflow",
	"_class": "EsmEventSource",
	"attributes": {
		"asoc-es-security": "Internal",
		"asoc-es-country": "USA",
		"asoc-es-bu": "IT",
		"asoc-es-type": "rsaflow",
		"asoc-es-criticality": NumberLong(1),
		"asoc-es-lan": "Trust",
		"asoc-es-vendor": "RSA",
		"asoc-es-division": "RSA",
		"asoc-es-city": "Boston",
```

```
        "asoc-es-wan": "None",
        "asoc-es-operational": "YES",
        "asoc-es-ip": "192.168.16.1",
        "asoc-es-color": "black",
        "asoc-es-contact": "John Black",
        "asoc-es-compliance": "YES",
        "asoc-es-upsProtected": true,
        "asoc-es-role": "Log Collection",
        "asoc-es-logDecoder": "Log Decoder",
        "asoc-es-voltage": NumberLong(220),
        "asoc-es-room": "2018",
        "asoc-es-department": "Cyber Defence",
        "asoc-es-desc": "RSA Security Analytics Log Decoder",
        "asoc-es-assetTag": "3333333333",
        "asoc-es-logDecoderUuid": "f07c11a2-6136-4d8b-bec3-a9a7e283e536",
        "asoc-es-manager": "Peter White",
        "asoc-es-domain": "ASOC.COM",
        "asoc-es-dnsHostname": "logdecoder.asoc.com",
        "asoc-es-backupAdmin": "Tim Short",
        "asoc-es-custom1": "Critical",
        "asoc-es-logCollector": "logdecoder",
        "asoc-es-company": "EMC",
        "asoc-es-postalCode": "01207",
        "asoc-es-primaryAdmin": "John Black",
        "asoc-es-priority": NumberLong(1),
        "asoc-es-name": "Log Decoder",
        "asoc-es-building": "RSA",
        "asoc-es-uid": "192.168.16.1-rsaflow",
        "asoc-es-rackHeight": NumberLong(1),
        "asoc-es-state": "Massachusetts",
        "asoc-es-floor": "4th",
        "asoc-es-email": "john.black@asoc.com",
        "asoc-es-serialNumber": "888888888",
        "asoc-es-systemName": "Log Decoder",
        "asoc-es-logCollectorUuid": "logdecoder",
        "asoc-es-systemDesc": "RSA Security Analytics Log Decoder"
},
"searchAttributes": {
        "asoc-es-security": "INTERNAL",
        "asoc-es-country": "USA",
        "asoc-es-bu": "IT",
        "asoc-es-type": "RSAFLOW",
        "asoc-es-criticality": NumberLong(1),
        "asoc-es-lan": "TRUST",
        "asoc-es-vendor": "RSA",
        "asoc-es-division": "RSA",
        "asoc-es-city": "BOSTON",
        "asoc-es-wan": "NONE",
        "asoc-es-operational": "YES",
        "asoc-es-ip": NumberLong("3232239617"),
        "asoc-es-color": "BLACK",
        "asoc-es-contact": "JOHN BLACK",
        "asoc-es-compliance": "YES",
        "asoc-es-upsProtected": true,
        "asoc-es-role": "LOG COLLECTION",
        "asoc-es-logDecoder": "LOG DECODER",
        "asoc-es-voltage": NumberLong(220),
        "asoc-es-room": "2018",
        "asoc-es-department": "CYBER DEFENCE",
```

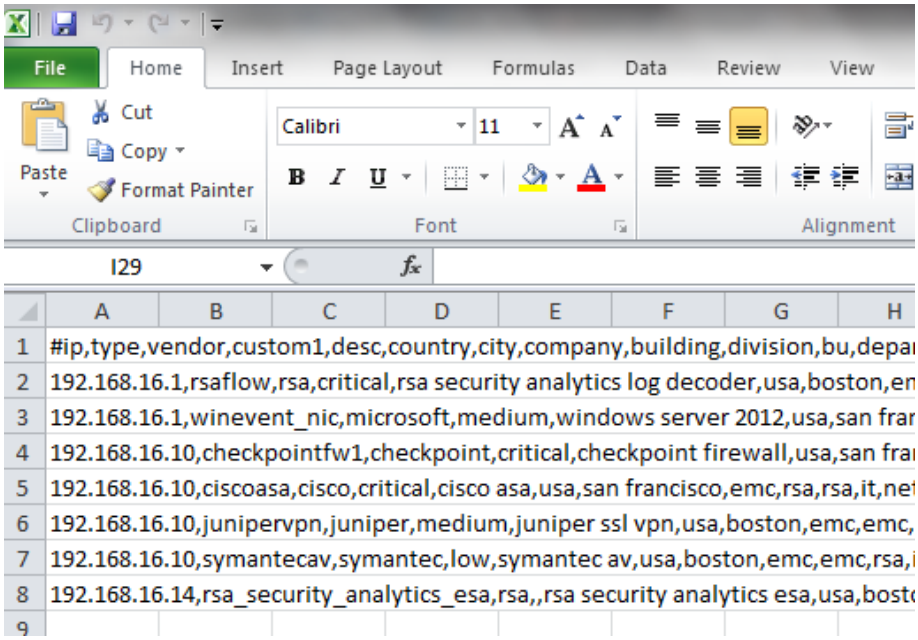
```
                "asoc-es-desc": "RSA SECURITY ANALYTICS LOG DECODER",
                "asoc-es-assetTag": "3333333333",
                "asoc-es-logDecoderUuid": "F07C11A2-6136-4D8B-BEC3-A9A7E283E536",
                "asoc-es-manager": "PETER WHITE",
                "asoc-es-domain": "ASOC.COM",
                "asoc-es-dnsHostname": "LOGDECODER.ASOC.COM",
                "asoc-es-backupAdmin": "TIM SHORT",
                "asoc-es-custom1": "CRITICAL",
                "asoc-es-logCollector": "LOGDECODER",
                "asoc-es-company": "EMC",
                "asoc-es-postalCode": "01207",
                "asoc-es-primaryAdmin": "JOHN BLACK",
                "asoc-es-priority": NumberLong(1),
                "asoc-es-name": "LOG DECODER",
                "asoc-es-building": "RSA",
                "asoc-es-uid": "192.168.16.1-RSAFLOW",
                "asoc-es-rackHeight": NumberLong(1),
                "asoc-es-state": "MASSACHUSETTS",
                "asoc-es-floor": "4TH",
                "asoc-es-email": "JOHN.BLACK@ASOC.COM",
                "asoc-es-serialNumber": "888888888",
                "asoc-es-systemName": "LOG DECODER",
                "asoc-es-logCollectorUuid": "LOGDECODER",
                "asoc-es-systemDesc": "RSA SECURITY ANALYTICS LOG DECODER"
        }
}
```


Developed “esm_mongo_export.sh” script export ESM Asset Attribute information from Mongo ESM DB on SA Server and create CSV file. CSV file is then copied to Log Decoder via SCP and used for Multi-indexed feed file.

esm_mongo_export.sh

Script will create “assets.csv” file located in “/var/netwitness/srv/www/feeds” folder and output log messages to “mongoexport.log” file located in “/var/log/” directory in SA Server.

Example of CSV file is shown below.



First line in assets.csv file represents different attributes collected from Mongo ESM DB and attributes are described below.

ip -> Asset IP address

type -> Device Type collection

vendor -> Asset Vendor

custom1 -> Asset Criticality

desc -> Asset Description

country -> Asset Country

city -> Asset City

company -> Asset Company

building -> Asset Building

division -> Asset Division

bu -> Asset Business Unit

department -> Asset Department

manager -> Asset Manager

contact -> Asset Contact

email -> Asset Contact Email

security -> Asset Security

compliance -> Asset Compliance

wan -> Asset Wan zone

lan -> Asset Lan zone

dnshostname -> Asset FQDN

Attributes collected from Mongo ESM DB can be changed when modifying “fields” filter in “mongoexport” command.

4. Automatically copy Asset.csv file to Log Decoder via SCP

This additional step is required because currently multi-indexed feed can’t be created via SA UI, (even if multi-indexed XML file and CSV are loaded through SA UI).

To copy file via SCP without password prompt, Trust relationship needs to be establish between SA Server and Log Decoder.

SCP copy command is already included in “esm_mongo_export.sh” script.

To achieve trust relationship you need to generate new RSA key pair and copy public key to Log Decoder. If “root” user account is not used in a system, then use user account which will have permission to copy CSV file to Log Decoder and run script with same user. Below is example how to enable trust relationship for root account.

```
[root@saserver .ssh]# ssh-keygen -b 2048 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
df:f2:de:35:13:2b:72:69:bf:65:6d:62:c6:83:f5:c8 root@saserver
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|                 |
|        S     .. |
|         . . =.o+|
```

```
|          o.o=E+B|
|           o++o*+|
|            .o .o.|
+-----------------+
[root@saserver .ssh]#
[root@saserver .ssh]# cd /root/.ssh/
[root@saserver .ssh]# ll
total 12
-rw-------. 1 root root 1675 Mar 25 13:39 id_rsa
-rw-r--r--. 1 root root  395 Mar 25 13:39 id_rsa.pub
-rw-r--r--. 1 root root  395 Feb 18 22:24 known_hosts
[root@saserver .ssh]#
```

Copy public key to Log Decoder.

```
[root@saserver .ssh]# scp id_rsa.pub root@192.168.16.11:/root/.ssh/authorized_keys
root@192.168.16.11's password:
id_rsa.pub
100%  395     0.4KB/s   00:00
[root@saserver .ssh]#
```

Try ssh connection to Log decoder and you should be connected without entering password.

```
[root@saserver .ssh]# ssh root@192.168.16.11
Last login: Fri Mar 25 12:00:03 2016 from puppetmaster.local
[root@logdecoder ~]# exit
logout
Connection to 192.168.16.11 closed.
[root@saserver .ssh]#
```

5. Create Multi-indexed XML file

Multi-indexed XML file below is build based on attributes exported from Mongo ESM DB and will be used to enrich data when there is metadata match of “Device IP” and “Device Type”.

For example if we have single event source where we collect Red Hat Linux and Apache logs then we need to map IP address to a service type for log parsing. As well we need to separate log when doing enrichment and combination of “Device IP” and “Device Type” is best option in this case.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FDF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="feed-definitions.xsd">
	<FlatFileFeed comment="#" separator="," path="assets.csv" name="assetsfeed">
		<MetaCallback name="Device IP" valuetype="IPv4">
			<Meta name="device.ip"/>
		</MetaCallback>
		<MetaCallback name="DeviceType" valuetype="Text" ignorecase="true">
			<Meta name="device.type"/>
		</MetaCallback>
		<LanguageKeys>
			<LanguageKey valuetype="Text" name="vendor"/>
			<LanguageKey valuetype="Text" name="assetcriticality"/>
			<LanguageKey valuetype="Text" name="desc"/>
			<LanguageKey valuetype="Text" name="country"/>
			<LanguageKey valuetype="Text" name="city"/>
			<LanguageKey valuetype="Text" name="company"/>
			<LanguageKey valuetype="Text" name="building"/>
			<LanguageKey valuetype="Text" name="division"/>
			<LanguageKey valuetype="Text" name="bu"/>
			<LanguageKey valuetype="Text" name="department"/>
			<LanguageKey valuetype="Text" name="manager"/>
			<LanguageKey valuetype="Text" name="contact"/>
			<LanguageKey valuetype="Text" name="email.cont"/>
			<LanguageKey valuetype="Text" name="security"/>
			<LanguageKey valuetype="Text" name="compliance"/>
			<LanguageKey valuetype="Text" name="wanzone"/>
			<LanguageKey valuetype="Text" name="lanzone"/>
			<LanguageKey valuetype="Text" name="fqdn"/>
		</LanguageKeys>
		<Fields>
```

```
                    <Field type="index" index="1" key="Device IP"/>
                    <Field type="index" index="2" key="DeviceType"/>
                    <Field key="vendor" type="value" index="3"/>
                    <Field key="assetcriticality" type="value" index="4"/>
                    <Field key="desc" type="value" index="5"/>
                    <Field key="country" type="value" index="6"/>
                    <Field key="city" type="value" index="7"/>
                    <Field key="company" type="value" index="8"/>
                    <Field key="building" type="value" index="9"/>
                    <Field key="division" type="value" index="10"/>
                    <Field key="bu" type="value" index="11"/>
                    <Field key="department" type="value" index="12"/>
                    <Field key="manager" type="value" index="13"/>
                    <Field key="contact" type="value" index="14"/>
                    <Field key="email.cont" type="value" index="15"/>
                    <Field key="security" type="value" index="16"/>
                    <Field key="compliance" type="value" index="17"/>
                    <Field key="wanzone" type="value" index="18"/>
                    <Field key="lanzone" type="value" index="19"/>
                    <Field key="fqdn" type="value" index="20"/>
            </Fields>
        </FlatFileFeed>
</FDF>
```

Attached “assets.xml” file.

assets.xml

6. Manually create multi-indexed feed in Log Decoder

Currently there is no option to create multi-indexed feed via SA UI.

Only possible way is to create it manually on Log Decoder with NWConsole command.
Copy multi-indexed XML and assets CSV file into “/etc/netwitness/ng/feeds” on Log Decoder and then execute NwConsole command to create feed file. Status of feed file can be verified by NwConsole command. Process for creating and checking feed file is described below.

```
[root@logdecoder feeds]# pwd
/etc/netwitness/ng/feeds
[root@logdecoder feeds]# ll
total 28
-rw-r--r--. 1 root root 1431 Mar 24 08:58 assets.csv
-rw-r--r--. 1 root root 2385 Mar 24 10:30 assets.xml
-rw-------. 1 root root 2164 Mar 23 22:51 aveksa.feed
-rw-r--r--. 1 root root  393 Mar 23 22:51 aveksa.feed-attr.xml
-rw-------. 1 root root  383 Mar 24 08:58 esmfeed.feed
-rw-r--r--. 1 root root  133 Mar 24 08:58 esmfeed.feed-attr.xml
-rw-r--r--. 1 root root 4041 Feb 10 13:22 feed-definitions.xsd
[root@logdecoder feeds]#
[root@logdecoder feeds]#
[root@logdecoder feeds]# NwConsole -c "feed create assets.xml"
RSA Security Analytics Console 10.6.0.0.6993
Copyright 2001-2016, RSA Security Inc.  All Rights Reserved.

>feed create assets.xml
Creating feed assetsfeed...
done.  125 entries, 0 invalid records
All feeds complete.

Check Feed status.

[root@logdecoder feeds]# ll
total 40
-rw-r--r--. 1 root root 1485 Mar 25 20:14 assets.csv
-rw-------. 1 root root 5520 Mar 24 14:23 assets.feed
-rw-r--r--. 1 root root 2505 Mar 24 14:19 assets.xml
-rw-------. 1 root root 2164 Mar 23 22:51 aveksa.feed
-rw-r--r--. 1 root root  393 Mar 23 22:51 aveksa.feed-attr.xml
-rw-------. 1 root root  383 Mar 24 14:10 esmfeed.feed
-rw-r--r--. 1 root root  133 Mar 24 14:10 esmfeed.feed-attr.xml
-rw-r--r--. 1 root root 4041 Feb 10 13:22 feed-definitions.xsd
-rw-r--r--. 1 root root 2479 Mar 24 19:55 mongo.sh
[root@logdecoder feeds]#
[root@logdecoder feeds]# NwConsole -c "feed stats assets.feed"
RSA Security Analytics Console 10.6.0.0.6993
Copyright 2001-2016, RSA Security Inc.  All Rights Reserved.

>feed stats assets.feed
assetsfeed stats:
        version     : 0
        keys count  : 18
        values count: 55
        record count: 125
        meta key    : device.ip,device.type
        language keys:
                assetcriticality        Text
```

```
                bu      Text
                building        Text
                city    Text
                company Text
                compliance      Text
                contact Text
                country Text
                department      Text
                desc    Text
                division        Text
                email.cont      Text
                fqdn    Text
                lanzone Text
                manager Text
                security        Text
                vendor  Text
                wanzone Text

[root@logdecoder feeds]#
```

The “feed dump” command generates a normalized, key-value pair listing of an un-obfuscated feed file. You can use the resulting file to validate a feed file or assist in determining which records were considered invalid when the feed was created. Specifying an obfuscated feed file will result in an error. If *outfile* exists, the command will abort without overwriting the existing file.

```
[root@logdecoder feeds]# NwConsole -c "feed dump assets.feed assets.txt"
RSA Security Analytics Console 10.6.0.0.6993
Copyright 2001-2016, RSA Security Inc.  All Rights Reserved.

>feed dump assets.feed assets.txt

[root@logdecoder feeds]#
```

Below is partial example of normalized feed file.

```
[root@logdecoder feeds]# cat assets.txt
192.168.16.1,"rsaflow",vendor/vendor,rsa
192.168.16.1,"rsaflow",assetcriticality/assetcriticality,critical
192.168.16.1,"rsaflow",desc/desc,rsa security analytics log decoder
192.168.16.1,"rsaflow",country/country,usa
192.168.16.1,"rsaflow",city/city,boston
192.168.16.1,"rsaflow",company/company,emc
192.168.16.1,"rsaflow",building/building,rsa
192.168.16.1,"rsaflow",division/division,rsa
192.168.16.1,"rsaflow",bu/bu,it
192.168.16.1,"rsaflow",department/department,cyber defence
192.168.16.1,"rsaflow",manager/manager,peter white
192.168.16.1,"rsaflow",contact/contact,john black
192.168.16.1,"rsaflow",email.cont/email.cont,john.black@asoc.com
192.168.16.1,"rsaflow",security/security,internal
192.168.16.1,"rsaflow",compliance/compliance,yes
192.168.16.1,"rsaflow",wanzone/wanzone,none
192.168.16.1,"rsaflow",lanzone/lanzone,trust
192.168.16.1,"rsaflow",fqdn/fqdn,logdecoder.asoc.com
192.168.16.1,"winevent_nic",vendor/vendor,microsoft
192.168.16.1,"winevent_nic",assetcriticality/assetcriticality,medium
192.168.16.1,"winevent_nic",desc/desc,windows server 2012
```

7. Automatically create multi-indexed feed in Log Decoder

Only way to automatically create new feed file, when there are changes in CSV file is to use script.

```
#!/bin/bash/

#Change directory where multi-indexed XML and assets CSV files are stored in Log Decoder.
cd /etc/netwitness/ng/feeds/

#Create new feed file.
NwConsole -c "feed create assets.xml"
sleep 5

#Reload Decoder Feeds
NwConsole -c login localhost:50002 admin netwitness -c /decoder/parsers feed op=reload -c quit
```

Script attached below.

8. Customize “index-concentrator-custom.xml” file

To use Assets information in investigation, reporting and for RBAC you need to add following lines to you “index-concentrator-cutom.xml” file.

You can change information you are adding to “index-concentrator-cutom.xml”, if your “mongoexport” query include different event source attributes. You can also change name description below.

If you are using Broker device then you need to add same changes to your “index-broker-custom.xml” file.

```
<!-- RSA Security Analytics Local Event Sources Assets -->
<key description="Asset Vendor" format="Text" level="IndexValues" name="vendor" valueMax="1000" defaultAction="Open"/>
<key description="Asset Description" format="Text" level="IndexValues" name="desc" valueMax="1000" defaultAction="Open"/>
<key description="Asset Country" format="Text" level="IndexValues" name="country" valueMax="1000" defaultAction="Open"/>
<key description="Asset City" format="Text" level="IndexValues" name="city" valueMax="1000" defaultAction="Open"/>
<key description="Asset Company" format="Text" level="IndexValues" name="company" valueMax="1000" defaultAction="Open"/>
<key description="Asset Building" format="Text" level="IndexValues" name="building" valueMax="1000" defaultAction="Open"/>
<key description="Asset Division" format="Text" level="IndexValues" name="division" valueMax="1000" defaultAction="Open"/>
<key description="Asset Business Unit" format="Text" level="IndexValues" name="bu" valueMax="1000" defaultAction="Open"/>
<key description="Asset Department" format="Text" level="IndexValues" name="department" valueMax="1000" defaultAction="Open"/>
<key description="Asset Manager" format="Text" level="IndexValues" name="manager" valueMax="1000" defaultAction="Open"/>
<key description="Asset Contact" format="Text" level="IndexValues" name="contact" valueMax="1000" defaultAction="Open"/>
<key description="Asset Contact email" format="Text" level="IndexValues" name="email.cont" valueMax="1000" defaultAction="Open"/>
<key description="Asset Security" format="Text" level="IndexValues" name="security" valueMax="1000" defaultAction="Open"/>
<key description="Asset Compliance" format="Text" level="IndexValues" name="compliance" valueMax="1000" defaultAction="Open"/>
<key description="Asset Wan zone" format="Text" level="IndexValues" name="wanzone" valueMax="1000" defaultAction="Open"/>
<key description="Asset Lan zone" format="Text" level="IndexValues" name="lanzone" valueMax="1000" defaultAction="Open"/>
<key description="Asset FQDN" format="Text" level="IndexValues" name="fqdn" valueMax="1000" defaultAction="Open"/>
<key description="Asset Criticality" format="Text" level="IndexValues" name="assetcriticality" valueMax="1000"
defaultAction="Open"/>
```

9. Create Cron job task

To automatically run “esm_mongo_export.sh” and “assetsfeed.sh” scripts on SA Server and Log Decoder, cron job task needs to be created.

Based on your needs you can modify time when cron job task is executed.

Add executable permission for “esm_mongo_export.sh” and “assetsfeed.sh” scripts.

a. SA Server

```
[root@saserver feeds]#cd /var/netwitness/srv/www/feeds
[root@saserver feeds]# chmod +x esm_mongo_export.sh
```

b. Log Decoder

```
[root@logdecoder feeds]# cd /etc/netwitness/ng/feeds
[root@logdecoder feeds]# chmod +x assetsfeed.sh
```

For testing purposes following cron job tasks were created.

a. SA Server

```
#Run script every 5 min and export ESM assets from Mongo DB and copy it to Log Decoder
*/5 * * * * sh /var/netwitness/srv/www/feeds/esm_mongo_export.sh
```

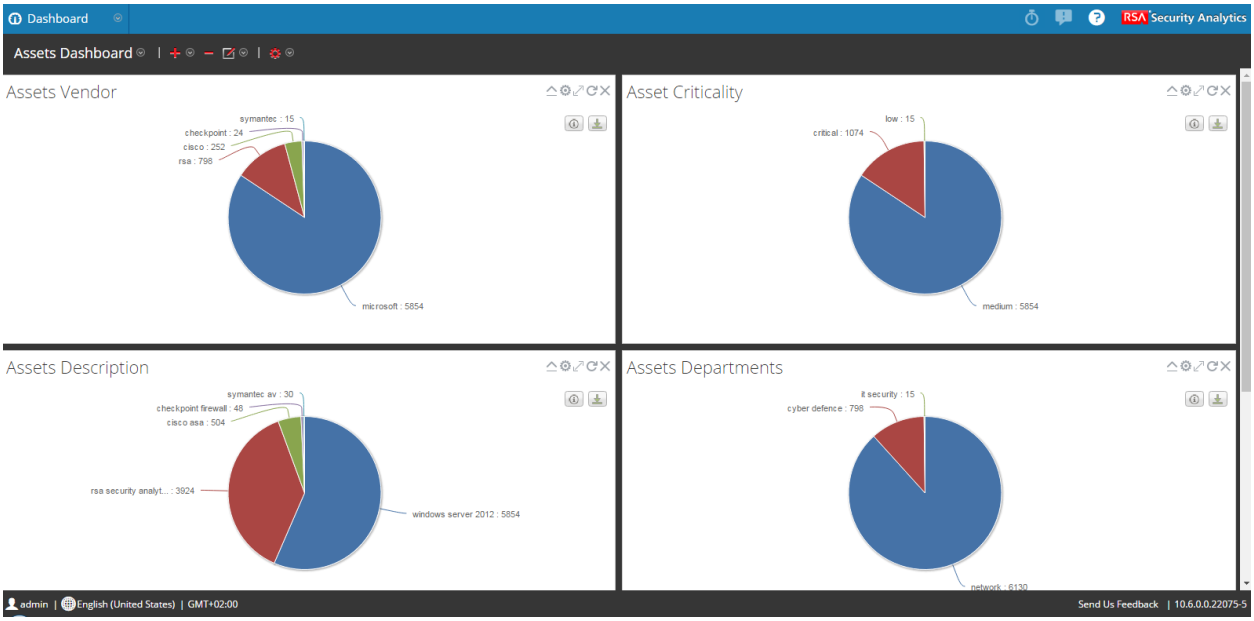
b. Log Decoder

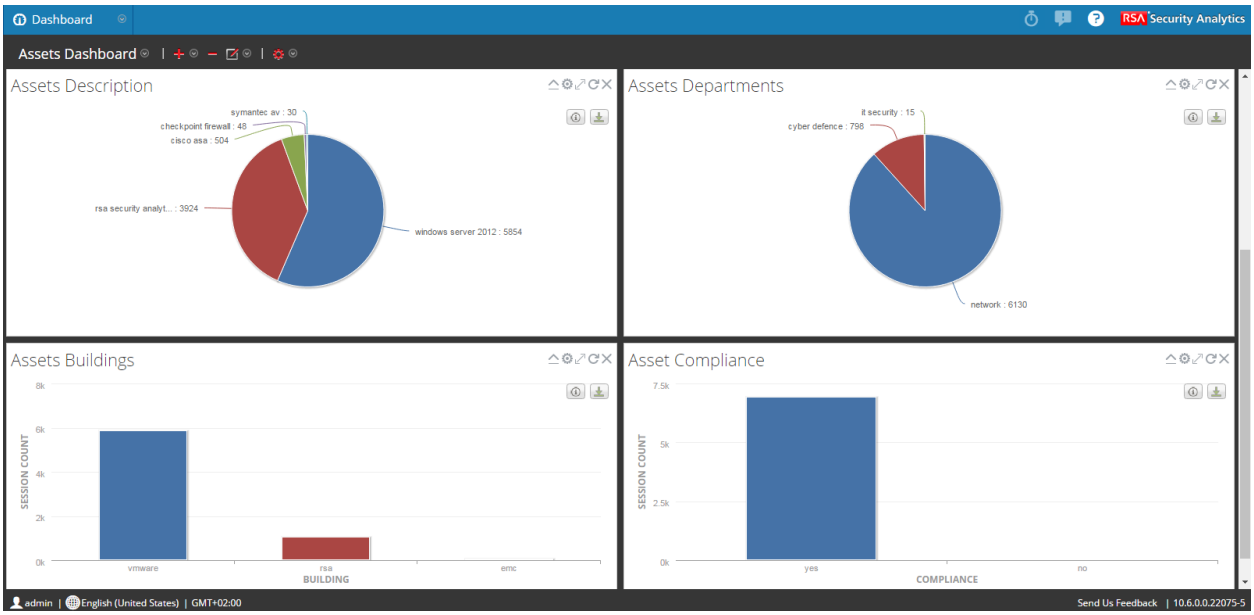
```
#Run script every 5 min to create and reload Log Decoder feeds
*/6 * * * * sh /etc/netwitness/ng/feeds/assetsfeed.sh
```

10. Security Analytics assets view

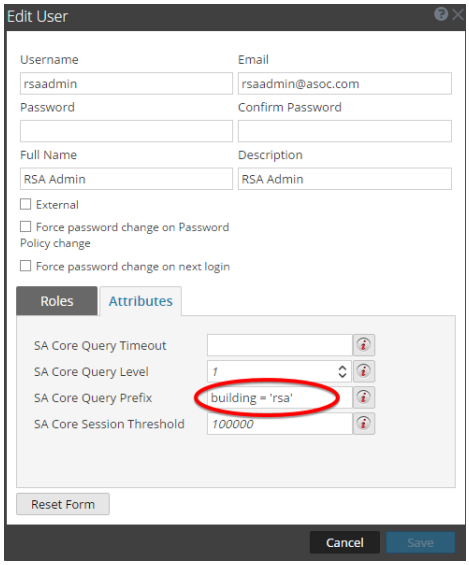
After all configurations steps you can configure different Reports, Charts, create RBAC or investigate collected data. When configuring RBAC control for users you can use "Query Prefix" option to limit metadata view for specific user.

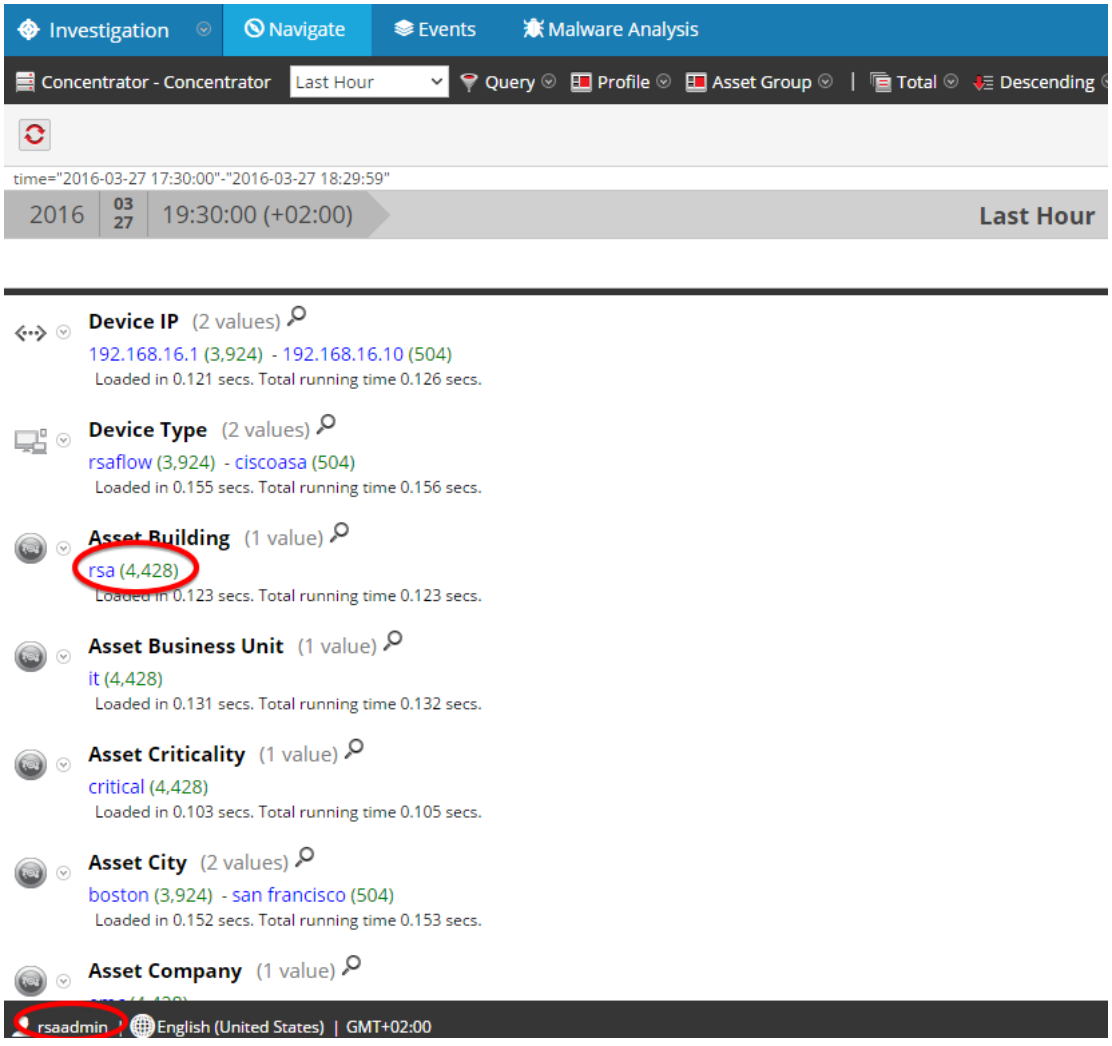
Below are simple examples of Dashboard, RBAC and investigation.





"Query Prefix" settings for user "rsaadmin" based on "building" metadata limitation.





Investigation with “admin” user without restriction.