ESA TROUBLESHOOTING (ESATOOL v2.0)

PABLO TRIGO

Table of Contents

1. REVISION HISTORY

Revision Number	Date	Revision
1	December 2015	First guide release

2. PREFACE

ABOUT THIS GUIDE

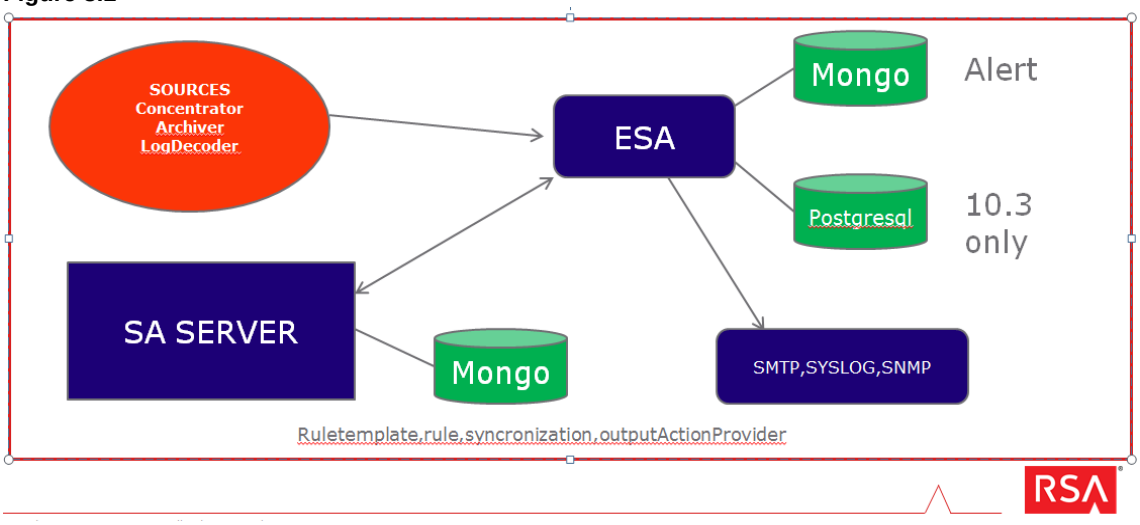
This guide describes how to troubleshoot ESA and how to use ESATOOL.

3. Chapter 1: ESA ARCHITECTURE

3.1 Overview

Esa has different architectures depending on the version installed, basically these are the main variations:

Figure 3.1

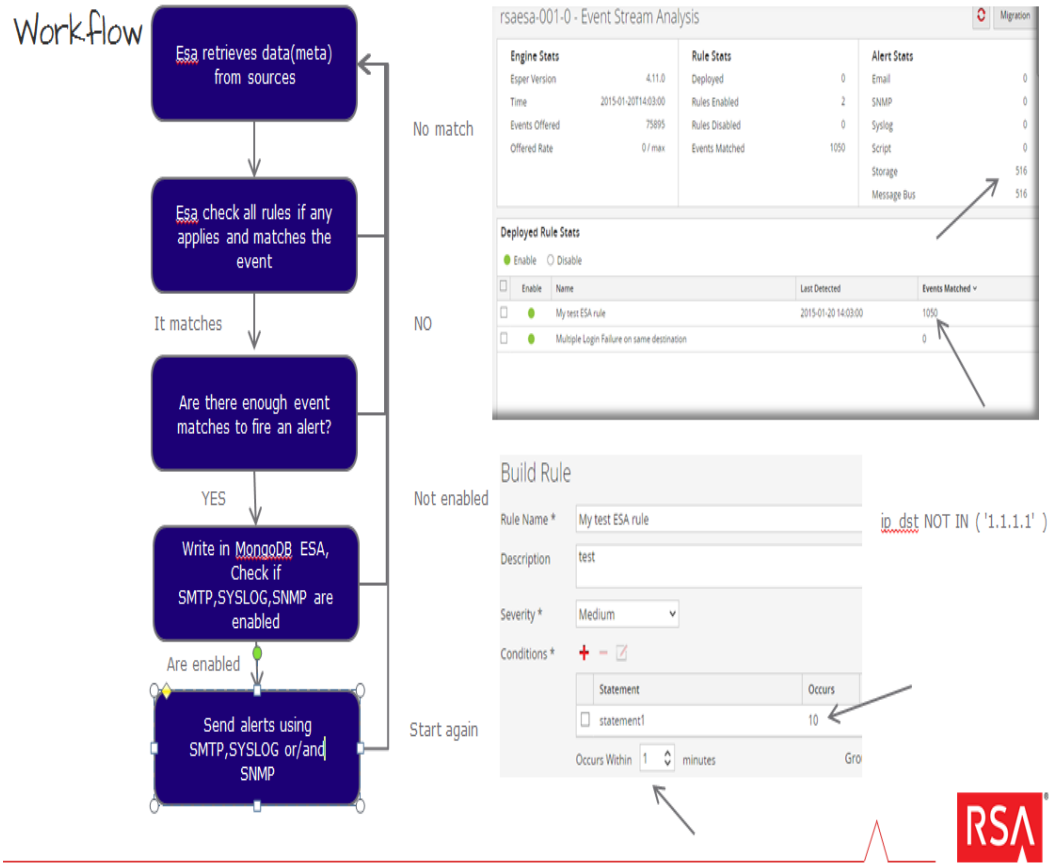


The Security Analytics Event Stream Analysis (ESA) service provides advanced stream analytics such as correlation and complex event processing at high throughputs and low latency. It is capable of processing large volumes of disparate event data from different sources.

ESA's advanced Event Processing Language allows you to express filtering, aggregation, joins, pattern recognition and correlation across multiple disparate event streams. Event Stream Analysis helps to perform powerful incident detection and alerting.

3.2 Basic WORKFLOW OF ESA

Figure 3.2 Workflow ESA



NOTE: EVENTS MATCHED IS NOT ALERTS FIRED

VERSION 10.3:

DATABASES : Postgresql (ONLY ESA)

RULES : ESAA FILES in ESA (ZIP FORMAT)

LICENSE: ESA

VERSION 10.4:

DATABASES: MONGODB (ESA AND SA SERVER)

RULES: SA SERVER (MONGODB)

LICENSE: ESA

VERSION 10.5

DATABASES: MONGODB (ESA AND SA SERVER)

RULES: SA SERVER (MONGODB)

LICENSE: SA SERVER(MONGODB)

Log files:

/opt/rsa/esa/logs/audit/audit.log

/opt/rsa/esa/logs/esa.log (esa log)

/opt/rsa/esa/wrapper.log (rsa-esa service log)

/usr/lib/rpm/rpm.log

/var/log/CAS.log

/var/log/mcollective.log

/var/log/rabbitmq/sa@localhost-sasl.log

/var/log/rabbitmq/sa@localhost.log

/var/log/tokumx/tokumx.log

/var/log/yum.log

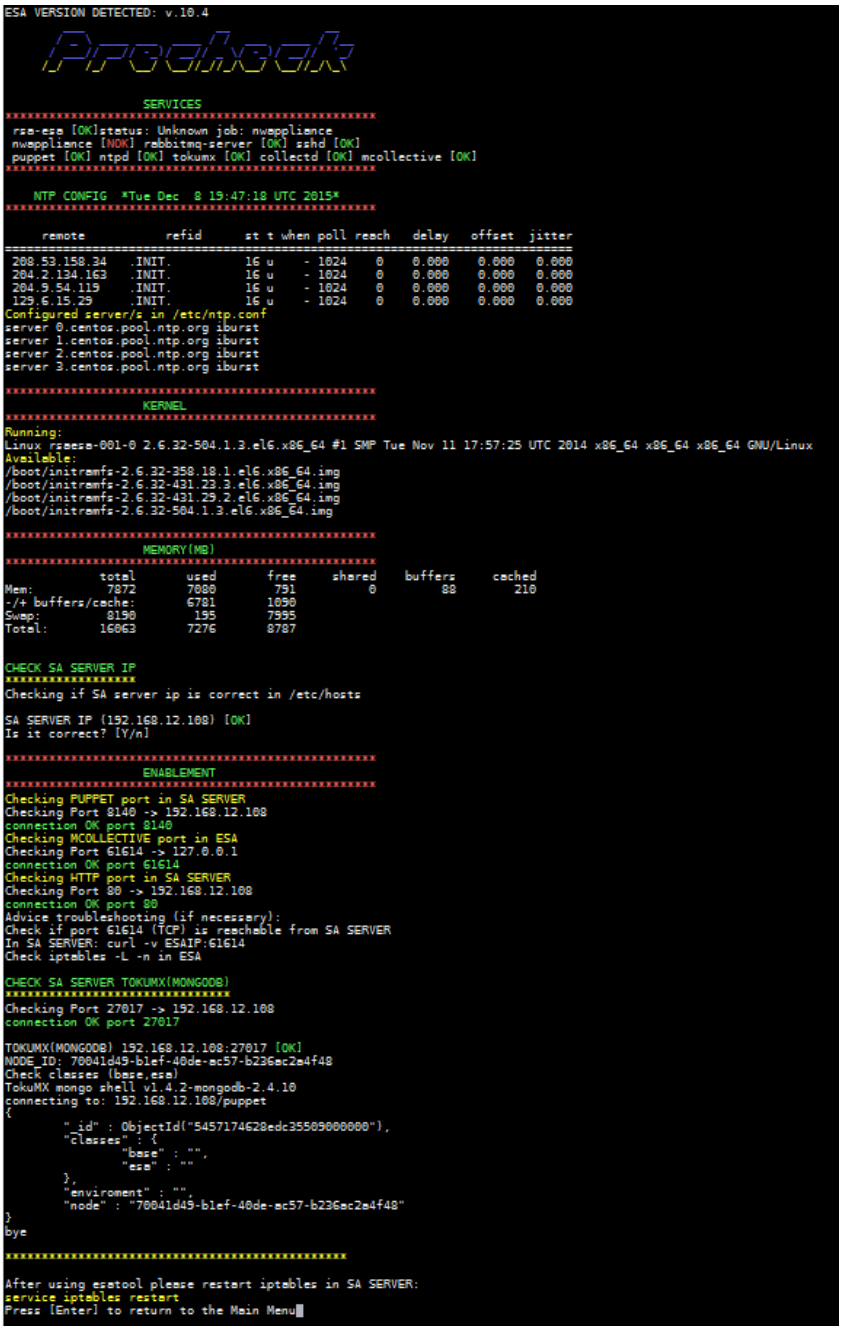
/var/log/messages

4. Chapter 2: Troubleshooting esa 10.4

4.1 EXECUTION of ESATOOL (PRECHECK)

This is the first screen after executing esatool:

Figure 4.1



We can see different stages of the precheck in the last screenshoot:

ESA VERSION DETECTED

Esatool now checks the version running to avoid any incompatibility with future versions.

SERVICES

Esatool checks the most relevant services running in the appliance

NTP

Exhaustive check of the ntp config and the current time in the system

KERNEL

Check of the current kernel and the available ones.

MEMORY

Memory check in MB (Physical + Swap)

SA SERVER IP CHECK

Esatool needs the current ip of the SA server to connect to mongodb. Rules of ESA are in mongodb in SA SERVER for instance. If SA SERVER mongodb is not reachable esatool will try to open this port remotely passing a command to SA SERVER through ssh.

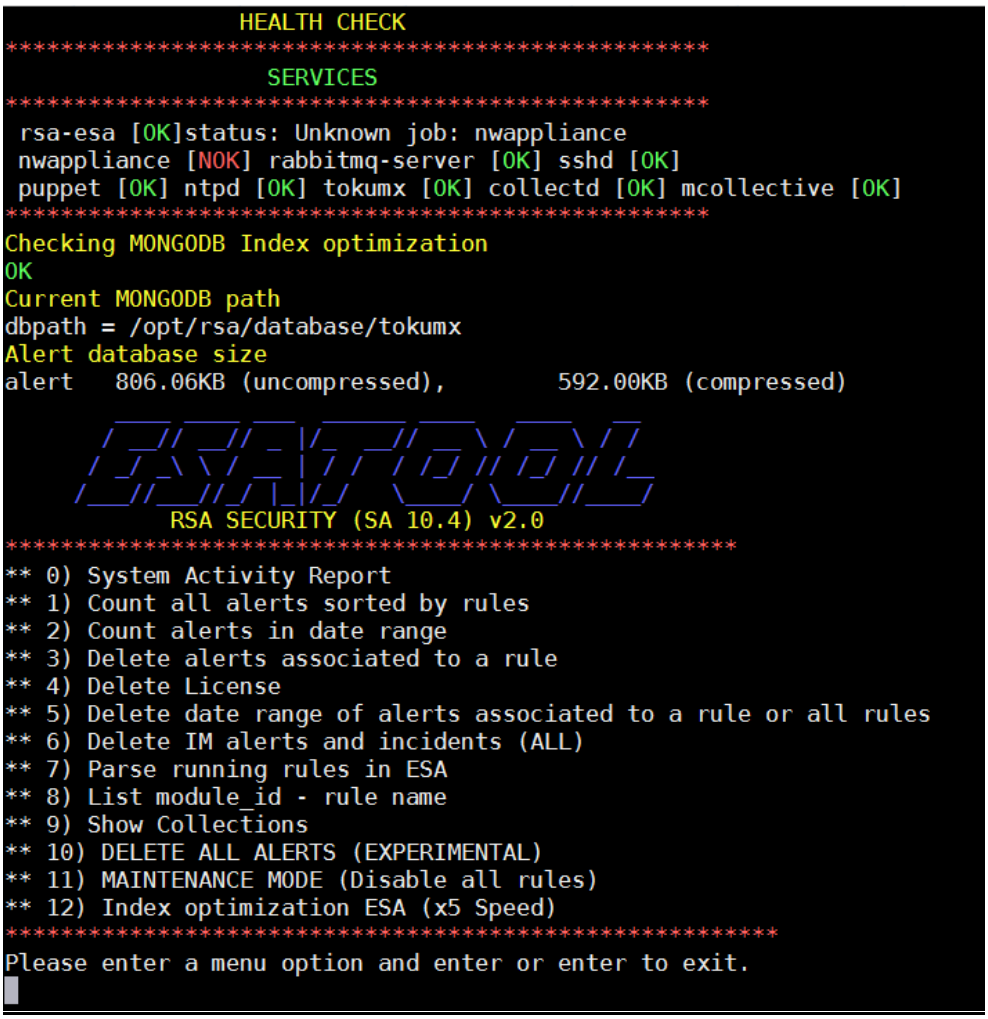
ENABLEMENT

Several checks in this stage:

- Checking puppetmaster connection in SA SERVER (Port 8140)
- Checking port 80 in SA SERVER (checking if the http server of SA SERVER is reachable)
- Checking if mcollective is up (port 61614) in esa, this is a mcollective server running in ESA and must be reachable from SA SERVER. Check if the iptables in ESA are blocking the connection (It can be also a physical firewall between SA SERVER and ESA).
- Check SA SERVER tokumx(mongodb) to verify if the entry of esa exists (classes must have base and esa). If esa or base class is missing check /etc/puppet/scripts/ in SA SERVER to fix the issue.
- If mongodb port in SA SERVER was opened by esatool in an early stage, please remember to restart iptables in SA SERVER to revert the configuration to the previous one.

4.2 Main screen esatool

This is the main screen of esatool after the system precheck. This is the main page:



The different sections are:

Health check

- Services: This sections is identical to the precheck, esatool checks the main processes in ESA.
    - Rsa-esa : rsa esa service
    - Nwappliance: rsa service to perform some actions in UI (Reboot,restart services,etc.)
    - Rabbitmq-server : Core messaging channel
    - Sshd: ssh server
    - Puppet: Configuration management utility.

- Ntpd: Network Time Protocol Daemon
- Tokumx: Enterprise version of Mongodb
- Collectd: collects, transfers and stores performance data of computers and network equipment.
- Mcollective: framework to build server orchestration or parallel job execution systems.

- MongoDB optimization: ESA index was not improved in some versions in 10.4, this checks if the index improvement is already in the database.
- Current mongodb path: By default the db path is /opt/rsa/database/tokumx , if the path is different then it's a problem(duplicated databases, alerts missing etc.)
- Alert database size: Problems start when the database is bigger than 10Gb, check the alert count to isolate the rule that is generating too much data in the database.

OPTIONS

0- System Activity Report

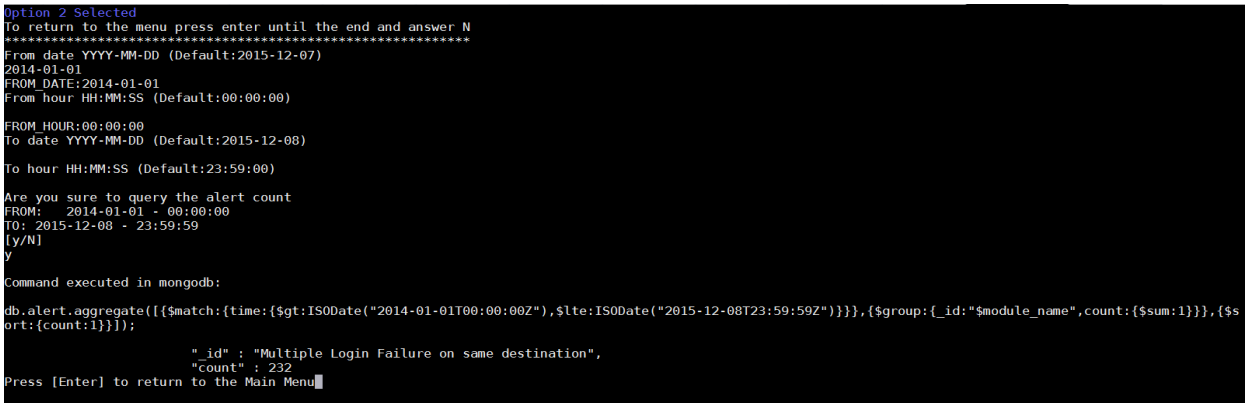
The system activity report writes to standard output the contents of selected cumulative activity counters in the operating system. The accounting system, based on the values in the count and interval parameters. Example (few lines of the option 0):

20:00:01	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
20:10:01	dev8-0	0.10	0.00	0.93	9.33	0.00	1.75	1.67	0.02
20:10:01	dev8-16	1.70	0.00	21.90	12.85	0.00	1.81	0.45	0.08
20:10:01	dev8-32	0.22	0.00	2.66	11.96	0.00	1.54	1.06	0.02
20:10:01	dev253-0	0.28	0.00	2.66	9.58	0.00	1.46	0.86	0.02
20:10:01	dev253-1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20:10:01	dev253-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20:10:01	dev253-3	1.51	0.00	12.08	8.00	0.00	1.79	0.14	0.02
20:10:01	dev253-4	0.07	0.00	0.52	8.00	0.00	4.90	1.10	0.01
20:10:01	dev253-5	1.16	0.00	9.29	8.00	0.00	1.71	0.44	0.05
20:10:01	dev253-6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20:10:01	dev253-7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20:20:01	dev8-0	0.10	0.00	0.93	9.33	0.00	1.42	1.23	0.01
20:20:01	dev8-16	1.68	0.00	21.63	12.91	0.00	1.83	0.47	0.08
20:20:01	dev8-32	0.21	0.00	2.48	11.79	0.00	1.25	1.04	0.02
20:20:01	dev253-0	0.28	0.00	2.48	8.89	0.00	1.20	0.78	0.02

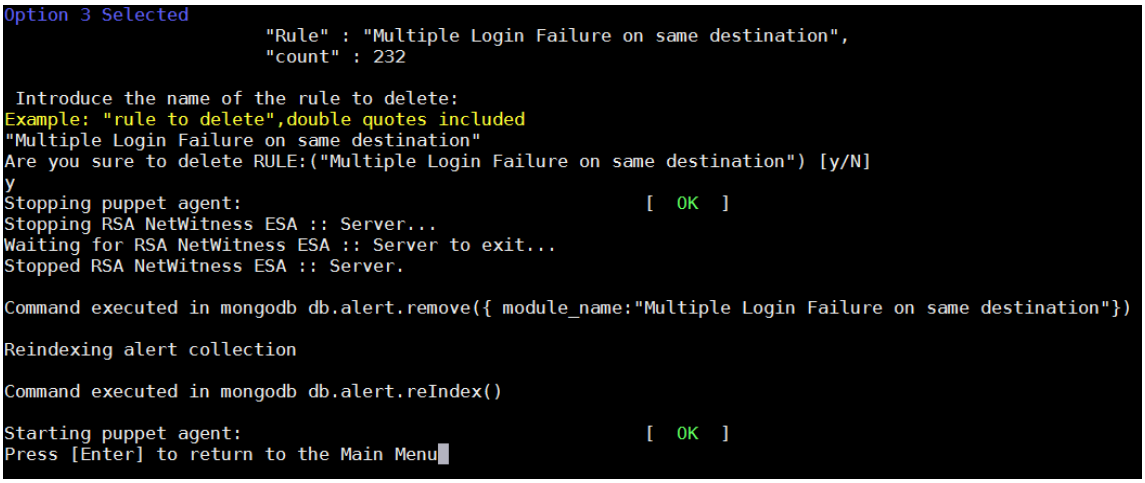
1- Count all alerts sorted by rules (Count all alerts in the system and sort them by rule)

```
Option 1 Selected
RULE_NAME :: MODULE_ID :: COUNT

Multiple Login Failure on same destination :: "Module_5aa794ec_bae8_4976_8ca7_189f00134eab" :: 232

Press [Enter] to return to the Main Menu
```

2- Count alerts in date range (Set the range that you want to count the alerts)

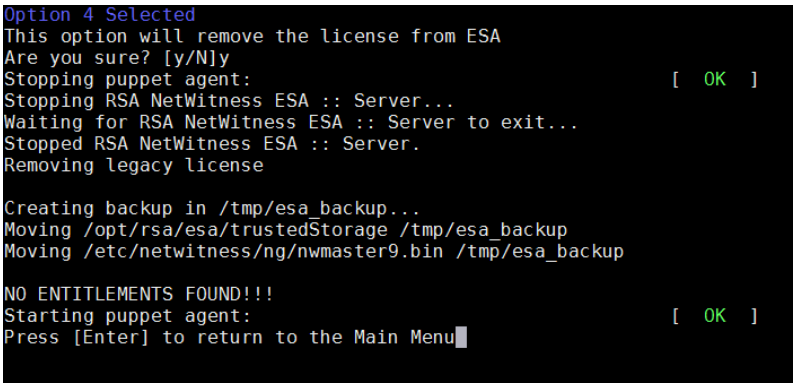


3- Delete alerts associated to a rule

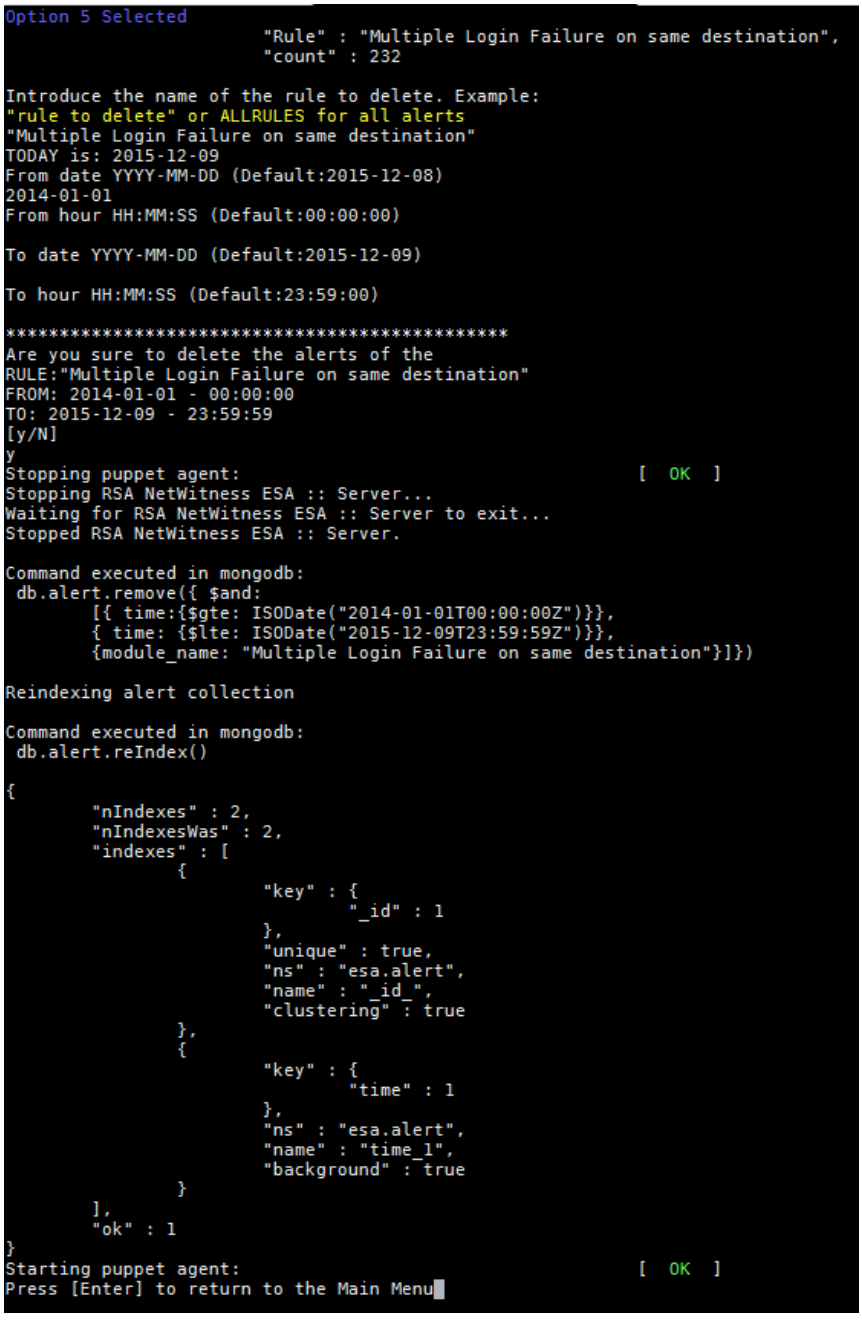


4- Remove License:

This feature is different in 10.4 and 10.5. In 10.4 the license is installed in ESA, however in 10.5 is in SA SERVER (entry in mongodb). In 10.4 it will delete the files only, but in 10.5 it will delete also the entry in SA SERVER database.



5- Delete date range of alerts associated to a rule or all rules:



6- Delete all alerts and incidents from IM

```
Option 6 Selected
Do you want to delete all alerts and incidents from IM[y/N]
y
Command executed in Mongodb: db.incident.remove()
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: im
bye
Command executed in Mongodb: db.alert.remove()
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: im
bye
Press [Enter] to return to the Main Menu
```

7- Parse running rules in ESA

```
Option 7 Selected
{"type": "Dictionary","dictionary": {"entry": [{"key": "MaxConstituentEvents","value": {"type": "Number","number": {"type": "INT_32","int32": 100
}}},{"key": "MessageBusEnabled","value": {"type": "Boolean","boolean": true}},{"key": "SerializedModules","value": {"type": "Array","array": {"el
ement": [{"type": "String","string": "{"identifier": "54585131e4b054918e5f6866","epl": "/*
This basic template is a placeholder for defining basic EPL content that can be
installed and executed in ESA. The sample below is the minimum that would be required
to get started.
*/


/*
Module debug section. If this is empty then debugging is off.
*/

/* EPL section. If there is no text here it means there were no statements. */

    module Module_54585131e4b054918e5f6866;



        @Name('Module_54585131e4b054918e5f6866_Alert')
        @Description('')
        @RSAAlert(oneInSeconds=0)

        SELECT * FROM Event(
            /* Statement: Statement 1 */
            (ec_outcome LIKE '%Failure%' AND ec_subject LIKE '%User%' AND ec_theme LIKE '%Authentication%' AND ip_src LIKE '%.%' AND ip_dst LIKE
'%66.135.192.83%')

        )
        .std:groupwin(ip_dst)
        .win:time_length_batch(1 Minutes, 10)
        GROUP BY ip_dst
        HAVING COUNT(*) >= 10;



","enabled": true,"name": "Multiple Login Failure on same destination","severity": 9}"}]}}},{"key": "DebugModules","value": {"type": "Boolean","b
oolean": false}}]}}


Created /root/parsed_rules.txt

*****************************************************

Press [Enter] to return to the Main Menu
```

8- List module id - rule

```
Option 8 Selected
{ "module_id" : "54585131e4b054918e5f6866", "module_name" : "Multiple Login Failure on same destination" }
{ "module_id" : "Module_5aa794ec_bae8_4976_8ca7_189f00134eab", "module_name" : "Multiple Login Failure on same destination" }
Press [Enter] to return to the Main Menu
```

9- Show collections of SA server and ESA.

```
Option 9 Selected
Command executed in mongodb ESA:
show collections
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: esa
system.indexes  476.00B (uncompressed), 32.00KB (compressed)
system.users    NaNundefined (uncompressed),    NaNundefined (compressed)
alert   806.06KB (uncompressed),        592.00KB (compressed)
bye

Command executed in mongodb SA SERVER:
show collections
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: 192.168.12.108:27017/sa
system.indexes  845.00B (uncompressed), 32.00KB (compressed)
enrichmentSource        402.00B (uncompressed), 32.00KB (compressed)
template        6.53KB (uncompressed),  32.00KB (compressed)
ruleTemplate    17.25KB (uncompressed), 64.00KB (compressed)
metaType        41.50KB (uncompressed), 64.00KB (compressed)
rule    2.81KB (uncompressed),  32.00KB (compressed)
synchronization 452.00B (uncompressed), 32.00KB (compressed)
outputActionProvider    915.00B (uncompressed), 32.00KB (compressed)
outputType      863.00B (uncompressed), 32.00KB (compressed)
bye
Press [Enter] to return to the Main Menu
```

10- Delete all alerts

```
Option 10 Selected
SECURITY NUMBER:457736
Introduce the number above to DELETE ALL ALERTS:
457736
Stopping puppet agent:                                     [  OK  ]
Stopping RSA NetWitness ESA :: Server...
Waiting for RSA NetWitness ESA :: Server to exit...
Stopped RSA NetWitness ESA :: Server.
TokuMX mongo shell v1.4.2-mongodb-2.4.10
connecting to: esa
true
bye
Starting puppet agent:                                     [  OK  ]
Press [Enter] to return to the Main Menu
```

11- Disable rules (Disable rules in 10.4 and instructions to remove the running ones in 10.5)

```
Option 11 Selected
Stopping puppet agent:                                     [  OK  ]
Stopping RSA NetWitness ESA :: Server...
Waiting for RSA NetWitness ESA :: Server to exit...
Stopped RSA NetWitness ESA :: Server.
All rules disabled in /opt/rsa/esa/conf/eplModuleManager.json
Starting puppet agent:                                     [  OK  ]
Press [Enter] to return to the Main Menu
```

12- Create index to optimize database.

```
Option 12 Selected
Do you want to optimize alert collection index? [y/N]
y
 Command executed in mongodb: db.alert.ensureIndex({time:1}, {background:true})
Press [Enter] to return to the Main Menu
```

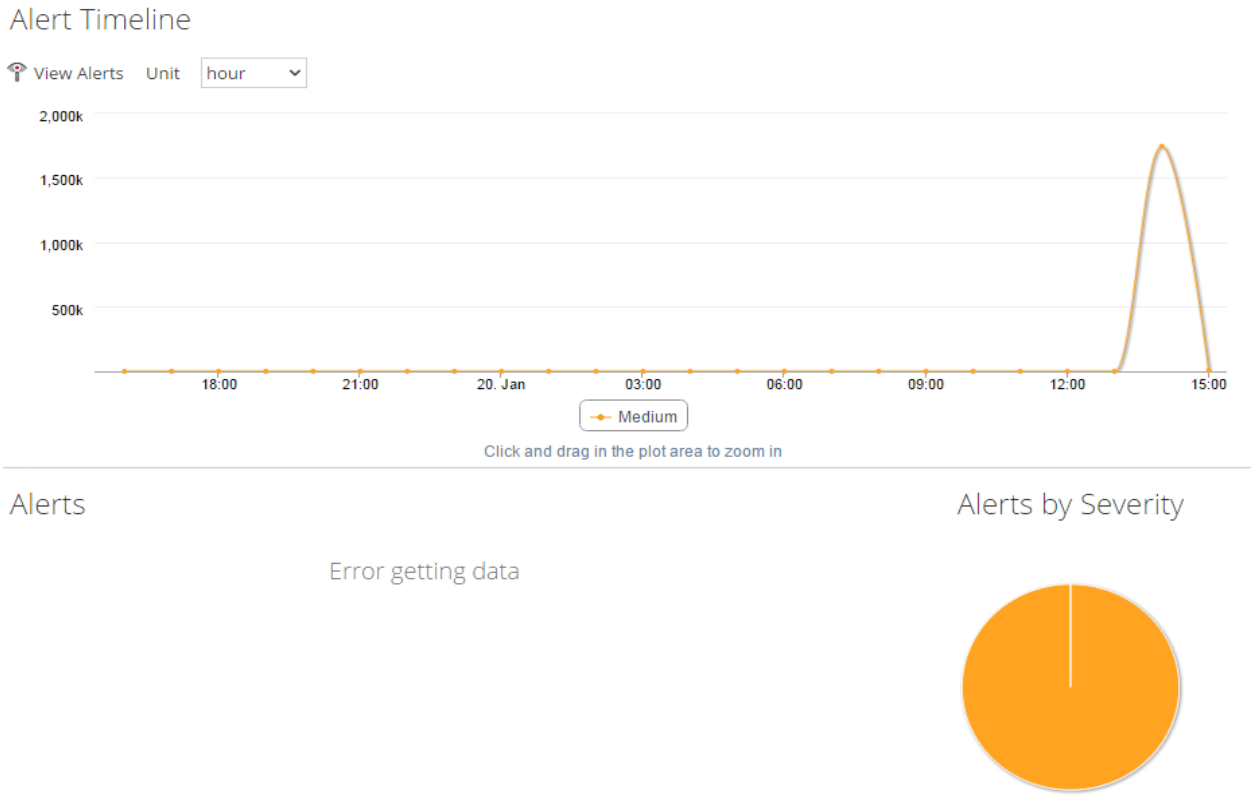
5. USE CASES

5.1 Error getting data

This is the most frequent issue in ESA, when the UI query the alert collection in mongodb it timeouts waiting an answer. This behavior is because there are rules (sometimes a simple rule) which triggers too much and cripples the system. Disabling the rule makes no difference.

SOLUTION:

- Option 1 to determine which rule/s are triggering too much.
- Option 3 to delete the rule.
- Check if the UI is ok.



5.2 After reimaging SA SERVER NO RULES in the UI

It seems counterintuitive that the rules are stored in SA SERVER instead of ESA. After reimaging SA SERVER if the collection rule in sa database is not backed up and restored all rules are gone. However in esa will be running in “memory”, we can recover the syntax of these rules and recreate them again in the UI.

SOLUTION:

- Use option 7 to parse the running rules in ESA.
- Create manually the rules in the UI with the information provided in the previous step.

5.3 ESA SERVICE RESTARTS EVERY X hours

Another frequent issue in ESA.

POSSIBLE REASONS:

- Rules with big windows time which exhausts the memory, that's why the service restarts(Needs memory so it's the last option that the service has).
- Low performing rules (bad designed rules).

POSSIBLE SOLUTION:

- Disable in blocks rules (example, groups of 3 rules) and check if the service is stable. Normally there is a rule which is allocating too much memory, therefore the esa service crashes.

6. Chapter 4: ESA 10.3 TROUBLESHOOTING

After the little introduction about esa architecture we are going to see how to troubleshoot ESA. This version is supported in esatool but is not automatized(Only displays the information regarding the commands that you need to use).

Useful files and folder location:

/opt/rsa/esa/modules/ (rules .esaa)

/opt/rsa/esa/content/ (templates)

/opt/rsa/esa/conf/ (current configuration of esa)

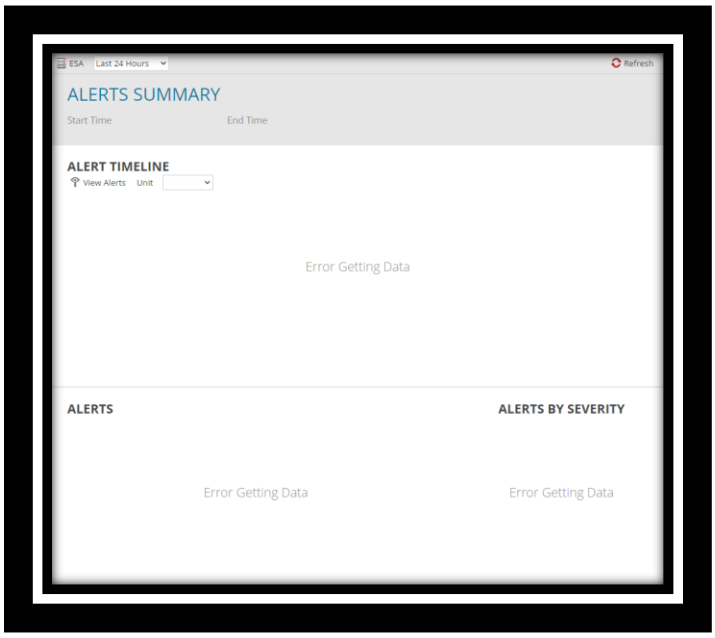
/opt/rsa/esa/logs (esa.log log regarding rules, aggregation etc.)

/opt/rsa/esa/wrapper.log (esa service log)

/opt/rsa/esa/conf/wrapper.conf (Configuration of esa service, memory allocation etc.)

/var/log/messages (General logs of the appliance)

Now, we will see the most common problem, rules normally bad designed/configured that generate a lot of alerts. The following commands would be useful to troubleshoot database/alert generation problems:



COMMANDS:

Connecting to postgresql and enabling “pretty” output: (to exit \q)

```
[root@rsaesa-001-0 ~]# psql -h localhost -U esa
Password for user esa: esa
esa=> \x on
Expanded display is on.
```

COUNT ALERTS

```
[root@rsaesa-001-0 ~]# psql -h localhost -U esa -c "SELECT module_name, COUNT(*) FROM alert GROUP BY module_name ORDER BY count DESC"

 module_name  | count

--------------+-------

 Bad Rule     |  3481

 Juniper data |   300

 SecOps_test1 |    68

(3 rows)
```

Newest alert:

```
esa=> SELECT * FROM alert ORDER BY time DESC limit 1;
-[ RECORD 1 ]-

id              | 71928313-17b2-489b-b02b-6360ffd30d49
time            | 2015-12-07 14:32:22.604
severity        | 1
module_id       | Module_ef3e29c4_53c7_4521_8da5_7ef3eac955e9
module_name     | bad rule
module_type     | basic
statement_name  | Module_ef3e29c4_53c7_4521_8da5_7ef3eac955e9_Alert
event_source_id | 192.168.12.108:50005:92654376
events          | {"event_cat_name=>\"Content.Web
Traffic\",esa_time=>1449498742595,device_type=>\"ciscoiportwsa\",did=>\"rsaaio\",sessionid=>92654376,medium=>32,size=>269,result_cod
e=>\"200\",time=>1449498719,header_id=>\"0001\",device_class=>\"Web
Logs\",level=>6,rid=>92654376,msg_id=>\"GET\",action=>\"\\\"TCP_HIT\\\"\",policy_name=>\"ALLOW_WBRS-DefaultGroup-
DefaultRouting\",parse_error=>\"EVENTTIME\",event_source_id=>\"192.168.12.108:50005:92654376\",ip_src=>\"10.10.51.70\",device_ip=>\"
127.0.0.1\""}
```

Oldest alert

```
esa=> SELECT * FROM alert ORDER BY time ASC limit 1;
-[ RECORD 1]-

id              | c6c65799-46ad-4aa7-8c0c-571bee73e055
time            | 2014-11-03 04:16:26.677
severity        | 9
module_id       | Module_5aa794ec_bae8_4976_8ca7_189f00134eab
module_name     | Multiple Login Failure on same destination
module_type     | basic
```

```
statement_name  | Module_5aa794ec_bae8_4976_8ca7_189f00134eab_Alert
event_source_id | 192.168.12.108:50005:259527
events          | {"ec_theme=>\"Authentication\",event_cat_name=>\"User.Activity.Privileged
Use.Denied\",esa_time=>1414988186651,user_dst=>\"cdion\",ec_subject=>\"User\",device_type=>\"ciscopix\",did=>\"rsaaio\",sessionid=>25
9527,medium=>32,size=>124,ec_outcome=>\"Failure\",time=>1414988177,header_id=>\"0004\",device_class=>\"Firewall\",ec_activity=>\"Lo
gon\",level=>6,rid=>259527,action=>\"\\\"User Authentication ..]
```

Delete all alerts generated by the alert 'bad rule' (total deleted 6908)

```
esa=> DELETE FROM alert WHERE module_name='bad rule';
DELETE 6908
```

Count between dates for a specific rule.

```
esa=> SELECT count(*) FROM alert where module_name='bad rule' and time > '2015-12-07 15:08:39.999' and time < '2015-12-07 15:09:42.00';
-[ RECORD 1 ]
count | 183
```

Similar to last query but now deleting the alerts.(total deleted 183)

```
esa=> DELETE FROM alert WHERE module_name='bad rule' AND time > '2015-12-07 15:08:39.999' AND time < '2015-12-07 15:09:42.00';
DELETE 183
```

Esatool execution:

```
ESA VERSION DETECTED: v.10.3

Version 10.3 detected skipping precheck
**************************************

CONNECT TO DATABASE(password esa):

psql -h localhost -U esa

** COUNT ALERTS **

Example:

psql -h localhost -U esa -c "SELECT module_name, COUNT(*) FROM alert GROUP BY module_name ORDER BY count DESC"

** DELETE ALERTS FOR A SPECIFIC RULE BETWEEN DATES**

Example:

 DELETE FROM alert WHERE module_name='bad rule' AND time > '2015-12-07 15:08:39.999' AND time < '2015-12-07 15:09:42.00';

For more information contact with RSA support
```

Esaa files structure:

```
10.3 root@rsaaio:~ # file esa000057.esaa
esa000057.esaa: Zip archive data, at least v2.0 to extract

10.3 root@rsaaio:~ # unzip esa000057.esaa
Archive:  esa000057.esaa
  inflating: META-INF/MANIFEST.MF
   creating: templates/esa000057/
  inflating: templates/esa000057/module.xml
  inflating: templates/esa000057/module.properties
  inflating: templates/esa000057/module.ftl

10.3 root@rsaaio:~ # cat templates/esa000057/module.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<module xmlns="http://www.netwitness.com/rta/1.0">
   <id>esa000057</id>
   <name ref="$module_name"/>
   <desc ref="$module_desc"/>
   <type>ootb</type>
   <severity>5</severity>
   <enabled>false</enabled>
   <last_modified>1435870404324</last_modified>
   <revision>2</revision>
   <variables>
      <variable is_array="false" id="time_window">
         <name ref="$time_window_name"/>
         <desc ref="$time_window_desc"/>
         <value type="int">60</value>
      </variable>
      <variable is_array="false" id="head_count">
         <name ref="$head_count_name"/>
         <desc ref="$head_count_desc"/>
         <value type="int">30</value>
      </variable>
   </variables>
   <alerters suppress="0"/>
   <support_files/>
</module>
```

A. Appendix Mongodb commands

ENABLE RULE EXAMPLE (ONLY 10.4)

```
service rsa-esa stop && sed -i 's/\\\"enabled\\\": true,\\\"name\\\": \\\"test123\\\"/\\\"enabled\\\": false,\\\"name\\\": \\\"test123\\\"/g' /opt/rsa/esa/conf/eplModuleManager.json && service rsa-esa start
```

DISABLE RULE EXAMPLE (ONLY 10.4)

```
service rsa-esa stop && sed -i 's/\\\"enabled\\\": false,\\\"name\\\": \\\"test123\\\"/\\\"enabled\\\": true,\\\"name\\\": \\\"test123\\\"/g' /opt/rsa/esa/conf/eplModuleManager.json && service rsa-esa start
```

DISABLE ALL RULES (ONLY 10.4)

```
service rsa-esa stop && sed -i 's/\\\"enabled\\\": true/\\\"enabled\\\": false /g' /opt/rsa/esa/conf/eplModuleManager.json && service rsa-esa start
```

COUNT ALERTS

```
echo 'db.alert.aggregate([{ $group: { _id: "$module_name", count: { $sum: 1 }}}
, { $sort: { count: 1 } }])' | mongo esa -u esa -p esa
```

COUNT ALERTS DATE RANGE

```
echo 'db.alert.aggregate([{$match : { time : { $gt : ISODate("2015-05-03T00:00:00Z"), $lte : ISODate("2015-05-04T23:59:59Z")}}},{$group: { _id: "$module_name", count: { $sum: 1 }}},{ $sort: { count: 1 } }]);' | mongo esa -u esa -p esa
```

DELETE LICENSE

```
mkdir -p /tmp/esa_backup && service rsa-esa stop && mv /opt/rsa/esa/trustedStorage /tmp/esa_backup && mv /etc/netwitness/ng/nwmaster9.bin /tmp/esa_backup && service rsa-esa start
```

DELETE RULE

```
echo 'db.alert.remove({ module_name:"myrule"})' | mongo esa -u esa -p esa
```

REINDEX COLLECTION

```
echo 'db.alert.reIndex()' | mongo esa -u esa -p esa
```

LIST MODULE_ID - RULES

```
for moduleid in $(echo 'db.alert.aggregate([{ $group : { _id : "$module_id" }}])' | mongo esa -u esa -p esa | grep id | cut -d':' -f2);do echo "db.alert.find({
module_id:"$moduleid"},{module_name:1,module_id:1,_id:0}).limit(1)" | mongo esa -u esa -p esa | grep module_id; done
```

DELETE ALL ALERTS

echo 'db.alert.drop()' | mongo esa -u esa -p esa

SQL-COMPARISON

SQL SELECT Statements	MongoDB find() Statements
SELECT * FROM alert	db.alert.find()
SELECT id, module_id, statement FROM alert	db.alert.find({ }, { module_id: 1, statement: 1 })
SELECT module_id, statement FROM alert	db.alert.find({ }, { module_id: 1, statement: 1, _id: 0 })
SELECT * FROM alert WHERE statement = "A"	db.alert.find({ statement: "A" })
SELECT module_id, statement FROM alert WHERE statement = "A"	db.alert.find({ statement: "A" }, { module_id: 1, statement: 1, _id: 0 })
SELECT * FROM alert WHERE statement != "A"	db.alert.find({ statement: { $ne: "A" } })

SQL SELECT Statements	MongoDB find() Statements
SELECT * FROM alert WHERE statement = "A" AND severity = 50	db.alert.find({ statement: "A", severity: 50 })
SELECT * FROM alert WHERE statement = "A" OR severity = 50	db.alert.find({ $or: [{ statement: "A" } , { severity: 50 }] })
SELECT * FROM alert WHERE severity > 25	db.alert.find({ severity: { $gt: 25 } })
SELECT * FROM alert WHERE severity < 25	db.alert.find({ severity: { $lt: 25 } })
SELECT * FROM alert WHERE severity > 25 AND severity <= 50	db.alert.find({ severity: { $gt: 25, $lte: 50 } })
SELECT * FROM alert WHERE module_id like "%bc%"	db.alert.find({ module_id: /bc/ })
SELECT * FROM alert WHERE module_id like "bc%"	db.alert.find({ module_id: /^bc/ })

SQL SELECT Statements	MongoDB find() Statements
SELECT * FROM alert WHERE statement = "A" ORDER BY module_id ASC	db.alert.find({ statement: "A" }).sort({ module_id: 1 })
SELECT * FROM alert WHERE statement = "A" ORDER BY module_id DESC	db.alert.find({ statement: "A" }).sort({ module_id: -1 })
SELECT COUNT(*) FROM alert	db.alert.count() *or* db.alert.find().count()
SELECT COUNT(module_id) FROM alert	db.alert.count({ module_id: { $exists: true } }) *or* db.alert.find({ module_id: { $exists: true } }).count()
SELECT COUNT(*) FROM alert WHERE severity > 30	db.alert.count({ severity: { $gt: 30 } }) *or* db.alert.find({ severity: { $gt: 30 } }).count()

SQL SELECT Statements	MongoDB find() Statements
SELECT DISTINCT(statement) **FROM** alert	**db.alert.distinct**("statement")
SELECT * **FROM** alert **LIMIT** 1	**db.alert.findOne**() *or* **db.alert.find**().**limit**(1)
SELECT * **FROM** alert **LIMIT** 5 SKIP 10	**db.alert.find**().**limit**(5).**skip**(10)
EXPLAIN SELECT * **FROM** alert **WHERE** statement = "A"	**db.alert.find**({ **statement**: "A" }).**explain**()