Use Case: Number of events occur within a time interval as long as absence of a specific event detected

Alert after receiving 10 different IDS events from the same source within 10 minutes but only if within those 10 minutes we don't see a TCP RST sent from the destination IP. This is an example of correlating packet and log data. Our F5's will do a TCP RST on the inbound web requests for unknown paths, so in this instance I only want to be alerted when a source receives 10 unique attacks to a single destination and that destination hasn't responded to the web requests.

Solution:

SELECT * FROM pattern @SuppressOverlappingMatches

Intrushield event followed by 9 others each with a unique policy_name and the same ip_src and ip_dst. The unique policy_name is controlled by the clause

where b.distinctOf(i => i.policy_name).countOf() = 9

The 10 minute time window following the first event is expressed by

timer:interval(600 seconds)

Both the statement for event b and event c must evaluate to true for the syntax to match. In other words, no TCP RST can occur to match the pattern.

AND NOT c=Event (medium=1 AND tcp_flags_seen ='rst' AND ip_dst=a.ip_dst)

```
@RSAAlert

SELECT * FROM pattern @SuppressOverlappingMatches

[

every a=Event (

device_type IN ( 'intrushield' )

AND ip_src is not null

AND ip_dst is not null

AND policy_name is not null

AND policy_name NOT LIKE '%P2P%'
```

```
)

-> (timer:interval(600 seconds)

AND

[9] b= Event (

device_type IN ( 'intrushield' )

AND ip_src = a.ip_src

AND ip_dst = a.ip_dst

AND policy_name is not null

AND policy_name NOT LIKE '%P2P%'

AND policy_name != a.policy_name

)

AND NOT

c=Event (medium=1 AND tcp_flags_seen ='rst' AND ip_dst=a.ip_dst)

)

] where b.distinctOf(i => i.policy_name).countOf() = 9;
```

Use Case: How to Correlate events that arrive out of order?

Correlates 3 events that populate the same ip_dst and occur within 30 of each other in any order

http://espertech.com/esper/release-5.3.0/esper-reference/html_single/index.html#epl-join-inner

```
/*
Intrusion Detection with Nonstandard HTTPS Traffic and ECAT Alert
Single host generates IPS alert on destination IP on port TCP/443
accompanied by traffic to TCP/443 that is not HTTPS with the target
host generating an ECAT alert within 5 minutes.
*/
```

```
/*
Create a window to store the IPS, nonstandard traffic and ECAT alerts
*/
@Name('create')
Create Window HttpsJoinedWindow.win:time(15 minutes)(device_class string, ip_dstport
integer, service integer , tcp_dstport integer, device_type string, ip_dst string);

/*
Insert into the window the IPS, nonstandard traffic and ECAT alerts
*/
@Name('insert')
INSERT INTO HttpsJoinedWindow
SELECT * FROM
Event
(
(ip_dst IS NOT NULL and device_class IN ('IPS', 'IDS', 'Firewall') AND ip_dstport=443)
OR
(ip_dst IS NOT NULL and service!=443 and tcp_dstport=443)
OR
(ip_dst IS NOT NULL and device_type='rsaecat')
);

/*
Alert to the combination of all three events: IPS, nonstandard traffic and ECAT alerts
*/
@RSAAlert
INSERT INTO HttpsIntrusionTrigger
SELECT * FROM
        HttpsJoinedWindow(ip_dst IS NOT NULL and device_class IN ('IPS', 'IDS', 'Firewall') AND
ip_dstport=443) as s1,
        HttpsJoinedWindow(ip_dst IS NOT NULL and service!=443 and tcp_dstport=443) as s2,
        HttpsJoinedWindow(ip_dst IS NOT NULL and device_type='rsaecat') as s3
        where s1.ip_dst = s2.ip_dst and s1.ip_dst = s3.ip_dst;

/*
Delete all events from the joined window that caused the alert so they won't be reused
*/
@Name('delete')
on HttpsIntrusionTrigger delete from HttpsJoinedWindow as j where s1.ip_dst=j.ip_dst;
```

Use case -- Only fire rules that are within business hours

http://espertech.com/esper/release-5.3.0/esper-reference/html_single/index.html#context_def_nonoverlapping

Non working hours.

- Set the working hours as '09:00' – '18:00'
- Any 'event.cat.name LIKE system.config%' after the working hours will trigger.

```
create context NotWorkingHours start (0, 18, *, *, *) end (0, 9, *, *, *);

context NotWorkingHours select * from Event(event_cat_name LIKE ‘system.config%’);
```

Use Case: How to leverage referencing lists via databases/files from ESA Rules such as domain or IPs

Create a Named Window to store IPs and update the window based on matching filter criteria. Only trigger if a second event occurs and the IP is on the watchlist. The user is only kept on the watchlist for 15 minutes. Use may delete from a named window based on a triggering event.

http://espertech.com/esper/release-5.3.0/esper-reference/html_single/index.html#named_delete

```
create window WatchListIPs.win:time(15 min) (ip_src string);

insert into WatchListIPs select ip_src from Event(category LIKE '%scan%');

 @RSAAlert
select * from Event(category LIKE '%malicious%') WHERE ip_src in (SELECT ip_src from WatchListIPs );
```

```
SELECT * FROM Event(

        (ip_dst IS NOT NULL ) AND NOT EXISTS (SELECT * FROM GeoIpLookup WHERE ( ipv4 = Event.ip_dst ) ))

      )
```

Use Case: How to computer percentages/rations/averages/counts/min/max within a given time window

Note: Computations over a large number of events and/or time periods are performance and memory intensive. Use caution when deploying the rules. See the ESA Enablement Guide for *10.5*. https://sadocs.emc.com/@api/deki/files/53102/ESA-enablement-guide-10_5.pdf

One way is to use named windows. However, this stores events in memory, which may cause issues if storing over a long period or large number of events.

```
CREATE WINDOW SizePerIP.win:length(100) (ip_src string,size long);

INSERT INTO SizePerIP SELECT ip_src AS ip_src, sum(size) AS size FROM Event.win:time_batch(1 minute) GROUP BY ip_src;

@RSAAlert(oneInSeconds=0)
SELECT ip_src FROM SizePerIP GROUP BY ip_src HAVING size > avg(size)*2;
```

Using a non-overlapping context does not retain events in memory and should be the preferred solution.

```
/*
Create a non-overlapping context to store data by second
*/
create context PerSecond start @now end after 1 second;

/*
Sum session size per second
*/

context PerSecond

insert into OneSecondBucket

select ip_src, sum(size) as size

from Event group by ip_src output snapshot when terminated;



/*
Alert if the total size for one second within an hour is two times greater than average
*/

@RSAAlert
```

```
select  ip_src from OneSecondBucket.win:time(1 hour) group by ip_src HAVING size >
avg(size)*2;
```

Question: What Regex filter support is supported?

http://espertech.com/esper/release-5.3.0/esper-reference/html_single/index.html#epl-operator-ref-keyword-regexp

The rexexp function matches the entire region against the pattern
via java.util.regex.Matcher.matches()method. Please consult the Java API documentation for more information or refer to Regular Expression Flavors.

Question: What is the difference between the count VS time length batch

```
SELECT * FROM Event(filter_criteria)
.std:groupwin(ip_src)
.win:time_batch(1 minute)
GROUP BY ip_src
HAVING count(*) > 10;
```

When the time window of 1 minute is reached it will output everything within it that matches an ip_src. The HAVING count clause instructs the engine to only output after the time window if the count of events is greater than 10. The GROUP BY ip_src aggregation instructs the count to apply to only a single ip_src instead of across all ip_src that match the filter criteria.

Question: How to identify periods of times that exceed memory usage

With 10.5 and above, I believe the Health and Wellness monitoring and alarms may be sufficient to monitor memory use.

https://sadocs.emc.com/0_en-us/089_105InfCtr/215_SysAdm/MonitorHlth

See also the ESA Enablement Guide for ***10.5***. It gives a good overview of

- Deployment best practices using trial rules
- Alarms for memory utilization and service status
- Per rule memory usage reporting

https://sadocs.emc.com/@api/deki/files/53102/ESA-enablement-guide-10_5.pdf

Question: How to feed rules into collector and create rules on top of rules

In 10.4, you would need to output the ESA alerts to syslog and create a custom parser. In 10.5 and above, you could deploy the CEF parser from Live and use this to parse the ESA alerts sent

over syslog in that format. The additional ESA rules or reports would need to be written against the parser output for the ESA alerts.