Learn How to Build Powerful Correlation Searches in Splunk Enterprise Security With Extreme Search

Macy Cronkrite

Professional Services, Splunk

Jack Coates

Product Management, Splunk

splunk>

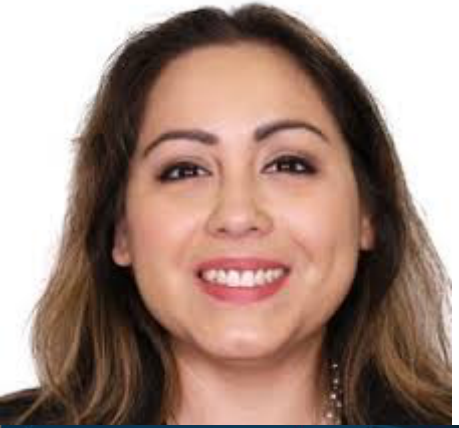
Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Personal Introduction

- Macy Cronkrite, Splunk
- Professional Service Consultant
- Public Sector Team
- Chief Meme Officer



- Jack Coates, Splunk
- Director of PM
- ES PM, 2.0 to 3.0
- Now focused on "Getting Data In and Out"
  - Add-ons, Utilities, & Other

Big Data, Analytics, and Security

"We talk about the need for analytics and business intelligence to help the business make better business decisions. It is time to bring this same technology to the information security department.

What we need is **actionable, prioritized and risk-based insight** from this sea of information. There are some emerging uses cases for information security which can only be handled with big data capabilities."

Gartner

Information Security is Becoming a Big Data Problem,
Neil MacDonald, Gartner, April 12, 2011

Agenda

- How correlations work in Enterprise Security:
  - Common Information Model
  - Assets and Identities
  - Risk Scoring Framework
- What is Extreme Search
  - Concepts and Contexts
  - Use ~~human~~ semantic phrases to frame security questions
- EXTREEEMEMEMEEEEEEEE
  - Discover and Manage security trends
  - Assess changes to enterprise risk exposure by segments of assets and users.

Splunk for Enterprise Security 3.3+

Foundational Knowledge of Enterprise environment

- Common Information Model
- Assets
- Identities
- Risk Framework
- Security Posture Domains – Access, Endpoint, Network, Threat

Extreme Search – A Concept

Qualitative Semantic Term

- A concept is a "Qualitative Semantic Term"... meaning a rich descriptive adjective or adverb that is associated with a field
- A "Semantic Term" comes from data models... the attributes (fields) of a data model are also known as Semantic Terms
- Concepts add qualitative descriptions to these attributes

Extreme Search – A Context

- A collection of concepts that are applied to a field (attribute) is known as a context

Data Models – Semantic Concepts

- The fields of an event (or a data model) are called attributes or Semantic Terms>>>
- If you want some descriptive term for a field, you create a concept.... (a concept is also known as a qualitative semantic term)
- A collection of concepts that describe a field is known as a context

Context Types

Data, User, Anomaly, Crossover

- Contexts can model different types of data.... they can be:
- Data Driven: run calculations against real data to build/update a context
- User Defined: build a context based on what you *think* the data distribution should be
- Anomaly Driven: build a context based on what value are outliers vs what values are typical
- Crossover Driven: build a context based on the "crossover" points.. usually generated by statistics
- These contexts are reduced models of large volumes of data...
- To use all this, you can create conceptual searches....

Not Your Boo

- NOT BOOLEAN
- A boolean query says "show me all of the events that MATCH this query"
- A conceptual query says "show me all of the events that are COMPATIBLE WITH this query"
- This is a key difference... compatibility gives you a range...

STD DEV is ok Not Great

- For example, you might want to get all of the events with a high std dev.
- A boolean query might be "| where stdev is >= 2.0”
- If your std dev is 1.97, it's not 2, so any std dev not at least 2.0 will fail...

- However, with a conceptual query, that wouldn't be the case
- A conceptual query might be "| xsWhere stdev is at least 2.0”
- All of the events that are COMPATIBLE with this query are returned, with a new field added, xsWhereCIX

XS WHERE CIX

COMPATIBLE

- xsWhereCIX Is the number that shows how compatible an event is with the query, so you can rank the results if necessary
- A better conceptual query might be "... | xsWhere stddev is at least high"
- Why use this? because what you're really asking for is "show me all the events that have a high standard deviation"

EXTREEEEEMEEEEEEE!!!!!

- | tstats allow_old_summaries=true count from datamodel=Intrusion_Detection by IDS_Attacks.signature
- | `drop_dm_object_name("IDS_Attacks")`
- | xswhere count from count_by_signature_1h in ids_attacks by signature is above medium

- This says show me all of the signatures over the last hour where the count by signature is above medium
- The query is different from a boolean query

MALWARE

NO BOOLEAN HERE BABY

- You can create different contexts based on different values of fields in an event.... for example, you might create a Context that describes the count of malware violations per day (signature_count_1d)....
- However, there are many different types of signatures and each can have a different range of values...
- So, you can create a different Context based on the value of the field "signature"... that lets you measure qualitatively the level of severity of a malware infestation based on the malware signature....
- A higher number of one type of malware may be more or less severe than another type of malware
- This type of context creation is known as Object Classification
- You classify which context to use based upon the value of an attribute (or field) in an event...
- If malware signature is X, then use the context signature_count_1d/X
- If malware signature is Y, then use the context signature_Count_1d/Y
- The value of malware_signature is different for each event
- Contexts are data reduction tools...
- A context models the data distribution of a field over time by class
- You can take millions of events and create/update contexts to reflect these data values...
- Contexts are small in size
- Contexts can be used on indexers and/or search heads
- Context query commands (xsWhere, xsFindBestConcept) are streaming commands
- This is important because you want to leverage the architecture that splunk provides to get the best performance possible

Questions?

Exercise 1
A Simple Rule

splunk>

Simple Detection

- `Name=WZCSVC`
- `Name=WZCSVC State=Running`
- `Name=WZCSVC State=Running host_priority=critical`
- `Name=WZCSVC State=Running host_priority=critical host_category=*pci*`

- Simple searches can simply be pasted into the Correlation Search Editor form
- Add macros to enrich the event: `get_event_id` | `map_notable_fields`

Correlation Search Editor Fields

- **Correlation Search Name**: A name, such as “My Search”. It will be automatically prepended with the Domain and appended with “– Rule”
- Drill down and throttle by should be discussed

Exercise 2
Correlate Within A Domain

splunk>

Correlate in place

- **Brute Force Access**
- `authentication
  by action |`

Does this search need updating?

The Correlation Three Step

- Gather a pool of data
- Use a table to form the data for easy numeric tests
- Perform tests to make decisions

Step 1: Gather a Pool of Events

- `authentication`

- Like most Enterprise Security Correlation Searches, this search uses a macro to simplify the final syntax.
- Macros definitions may be found at ***Manager > Advanced Search > Search Macros.***
- `tag=authentication NOT (action=success user=*$) | fillnull value=unknown action,app,src,src_user,dest,user`

- `authentication` returns raw events:

Step 2: Tabulate the Events for Comparison

- The chart command is then used to tabulate raw events for comparison
- `` `authentication`| chart count over src by action ``

	src	failure	success	unknown
1	10.11.36.1	5	2	0
2	10.11.36.2	6	1	0
3	10.11.36.3	2	2	0
4	10.11.36.4	3	4	0
5	10.11.36.5	1	2	0
6	10.11.36.6	8	4	0
7	10.11.36.7	0	1	0
8	10.11.36.8	4	1	0
9	10.11.36.9	5	2	0
10	10.11.36.10	3	1	0

Step 3: Make a Go/No-Go Decision

- The final search command filters to patterns that are worth human attention
- `` `authentication`| chart count over src by action | search failure>6 success>0 ``
- Over longer time periods, the search may generate many events

18 results in the last 24 hours (from 12:00:00 PM April 29 to 12:16:36 PM April 30, 2012)

Export Options « prev 1 2 next » 10 per page

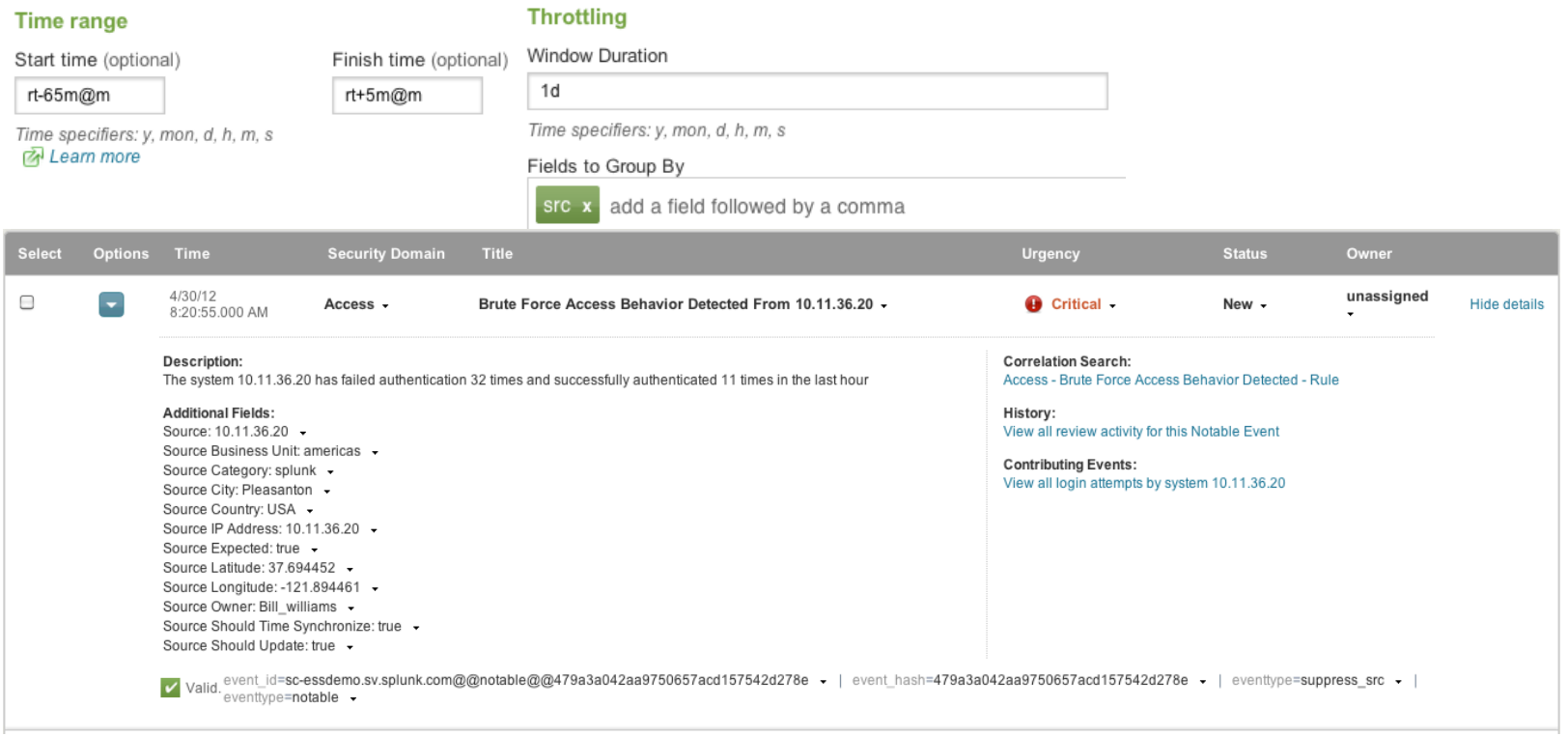
Overlay: None

	src	failure	success	unknown
1	10.11.36.1	7	3	0
2	10.11.36.2	9	1	0
3	10.11.36.6	12	7	0
4	10.11.36.9	8	2	0
5	10.11.36.13	9	2	0
6	10.11.36.14	7	2	0
7	10.11.36.17	13	1	0
8	10.11.36.20	277	93	0
9	10.11.36.27	9	2	0
10	10.11.36.33	8	1	0

« prev 1 2 next »

Manage Load with Ranges and Throttles

- Use time windows and throttling to prevent duplications or noise



Exercise 3
Correlate Across Multiple Domains

splunk>

Advanced Correlation Searches

- **The most challenging and useful searches will leverage multiple**

Should this be merged with the CIM based version?

Access Information – which user accounts are accessing which machines?

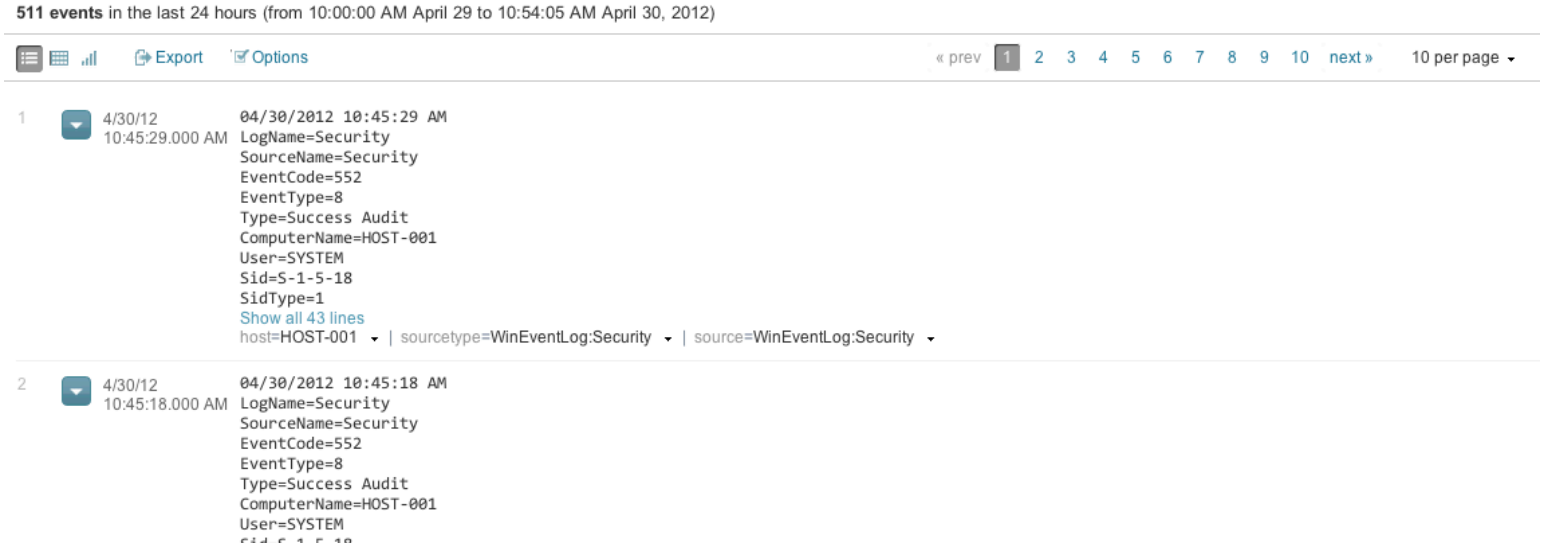
Asset and Identity Information – which users and machines are most critical?

Endpoint Information – which machines are currently infected with malware?

Did a high or critical priority user log into an infected machine?

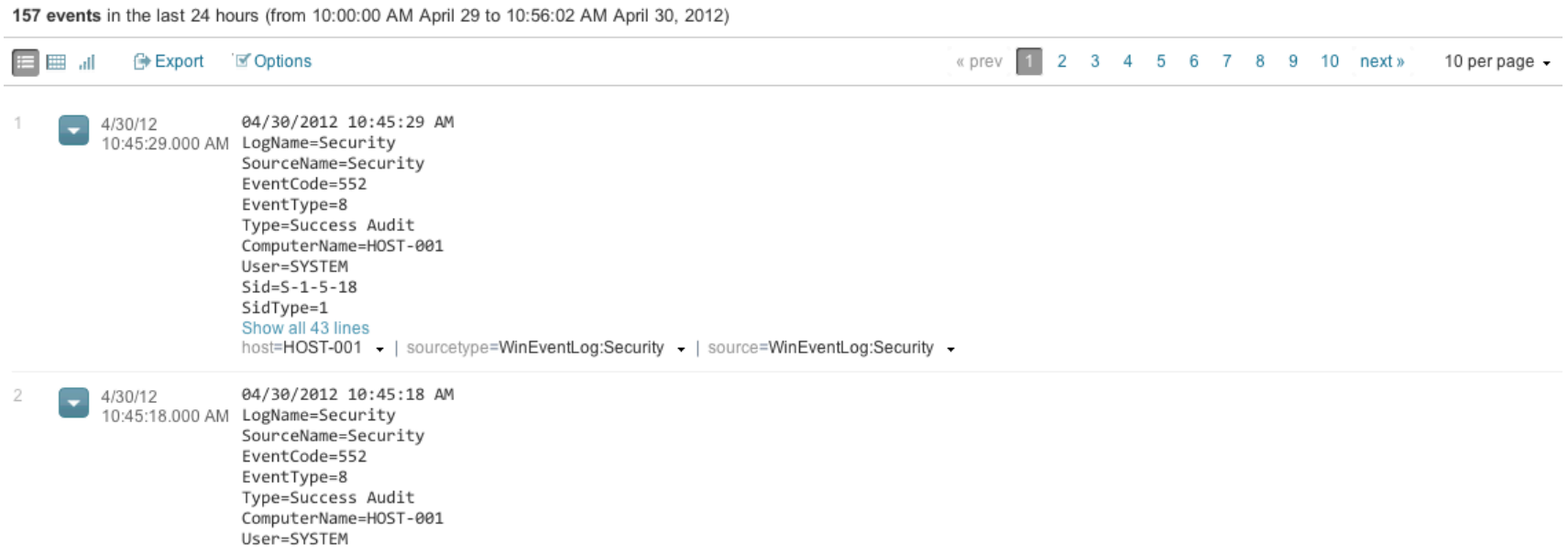
Step One: Authentications

- `authentication` macro gives all access events, but we're only looking for successful authentications… so we copy it and modify it
- `tag=authentication action=success`



Step Two: Priorities

- We have two useful fields to look at in this case, so an OR is used
- `(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high") )`



Step Three: Malware

- The `malware` macro provides useful guidance…
- `(tag=malware tag=attack action=allowed)`

1,437 events in the last 24 hours (from 11:00:00 AM April 29 to 11:02:13 AM April 30, 2012)

Export | Options | « prev 1 2 3 4 5 6 7 8 9 10 next » | 10 per page

1 4/30/12 10:16:38.000 AM

```
04/30/2012 10:16:38 AM
LogName=Application
SourceName=Trend Micro OfficeScan Server
EventCode=10
EventType=2
Type=Warning
ComputerName=ACMETREND1
User=SYSTEM
Sid=S-1-5-18
SidType=1
```

Show all 18 lines

host=sc-essdemo.sv.splunk.com | sourcetype=WinEventLog:Application:trendmicro | source=WinEventLog:Application

2 4/30/12 10:16:11.000 AM

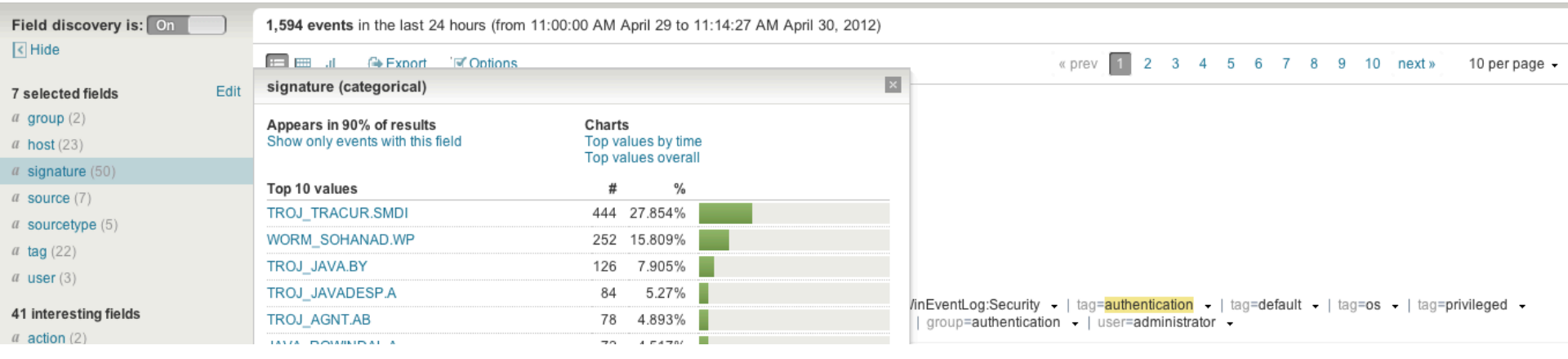
```
04/30/2012 10:16:11 AM
LogName=Application
SourceName=Trend Micro OfficeScan Server
EventCode=10
EventType=2
Type=Warning
ComputerName=ACMETREND1
User=SYSTEM
```

Step Four: Access OR Malware OR Privilege

- Construct the total pool of data to make decisions from
- `(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high") ) OR (tag=malware tag=attack action=allowed) | tags outputfield=tag | eval group=case(tag=="authentication","authentication",tag=="malware","malware")`

Step Five: Normalize Results

- All the expected fields might not be there in source data, so use evals to fill in with something sensible
- `(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high")) OR (tag=malware tag=attack action=allowed) | tags outputfield=tag | eval group=case(tag=="authentication","authentication",tag=="malware","malware") | eval user=if(tag=="malware",null(),user) | eval signature=if(tag=="authentication",null(),signature)`



Step Six: Wonder Twins, Form of Table!

- Tabulate the fields that the decision will be made from
- `(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high")) OR (tag=malware tag=attack action=allowed) | tags outputfield=tag | eval group=case(tag=="authentication","authentication",tag=="malware","malware") | eval user=if(tag=="malware",null(),user) | eval signature=if(tag=="authentication",null(),signature) | stats values(user) as user,values(signature) as signature,dc(group) as group_count by dest`

55 results in the last 24 hours (from 11:00:00 AM April 29 to 11:18:14 AM April 30, 2012)

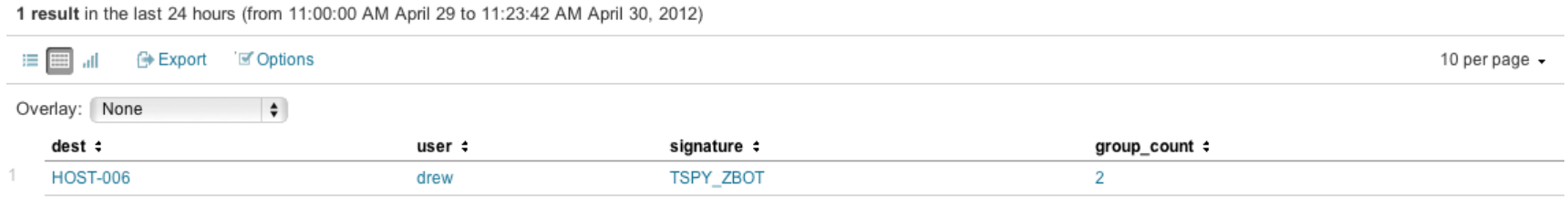
Export Options « prev 1 2 3 4 5 6 next » 10 per page

Overlay: None

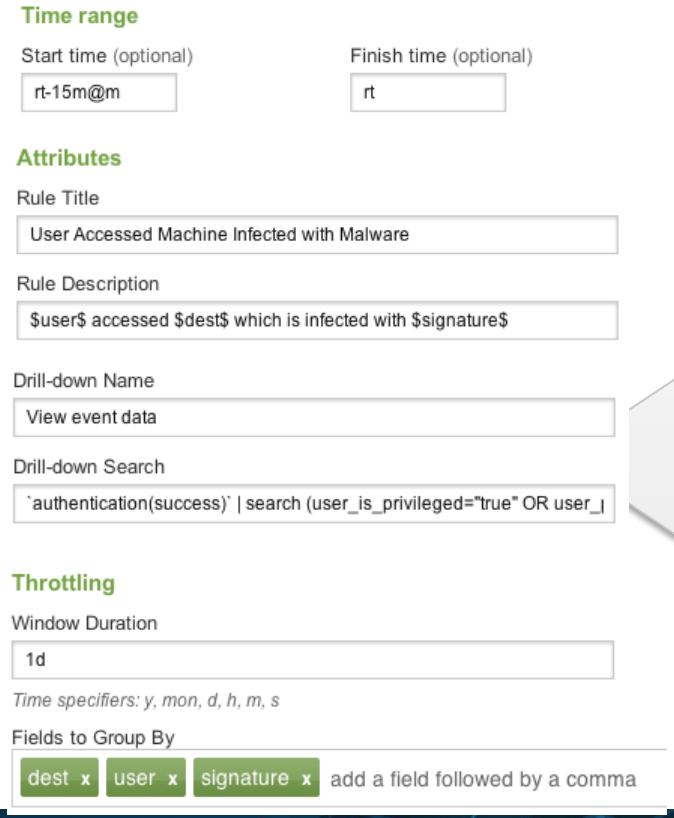
	dest	user	signature	group_count
1	10.11.36.20		Adware.Hotbar	1
2	ACME-003		Trojan.Vundo	1
3	ACME-006		SecurityRisk.eGatherer	1
4	ACME-CA0382FD		WORM_SOHANAD.WP	1
5	ACME-FE50DB		TROJ_BREDO.SMGS TROJ_FAKELRT.SMT	1
6	ACMETREND	root		1
7	ACMETREND1	root		1
8	BEE		Mal_Hifrm	1
9	BENNETTG2		TROJ_FAKEAV.SMBG	1
10	BLACKTOP		TROJ_FAKEAV.SMZQ	1

Step Seven: Detect the Condition

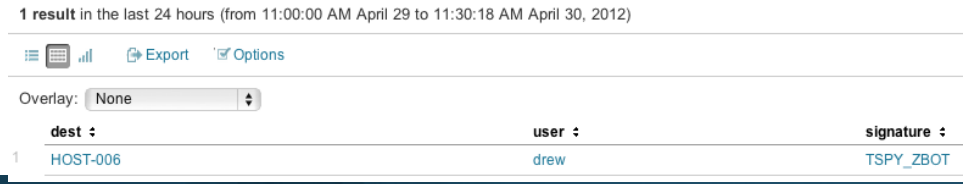
- group_count field > 1 is bad
- `(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high")) OR (tag=malware tag=attack action=allowed) | tags outputfield=tag | eval group=case(tag=="authentication","authentication",tag=="malware","malware") | eval user=if(tag=="malware",null(),user) | eval signature=if(tag=="authentication",null(),signature) | stats values(user) as user,values(signature) as signature,dc(group) as group_count by dest | search group_count>1`



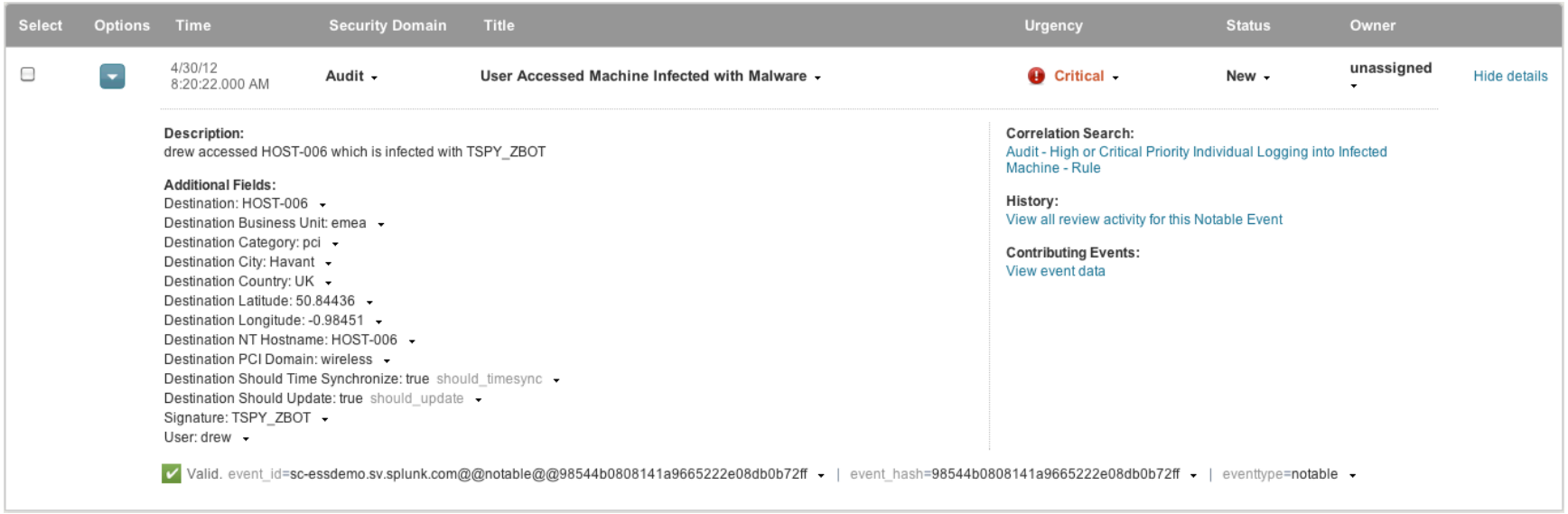
Plug it into the Search Editor



- Drill down can use a different search – efficiency is not as important, and the results desired don't require a count
- `authentication(success)` | search (user_is_privileged="true" OR user_priority="critical" OR user_priority="high") | stats values(user) as user by dest | join dest [search `malware` | search action=allowed | stats values(signature) as signature by dest]



The Resulting Notable Event

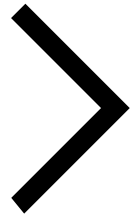


Exercise 4 –
Use the CIM, Luke!

splunk>

Normalization: Not Just a Dirty Word

(tag=malware tag=attack action=allowed)

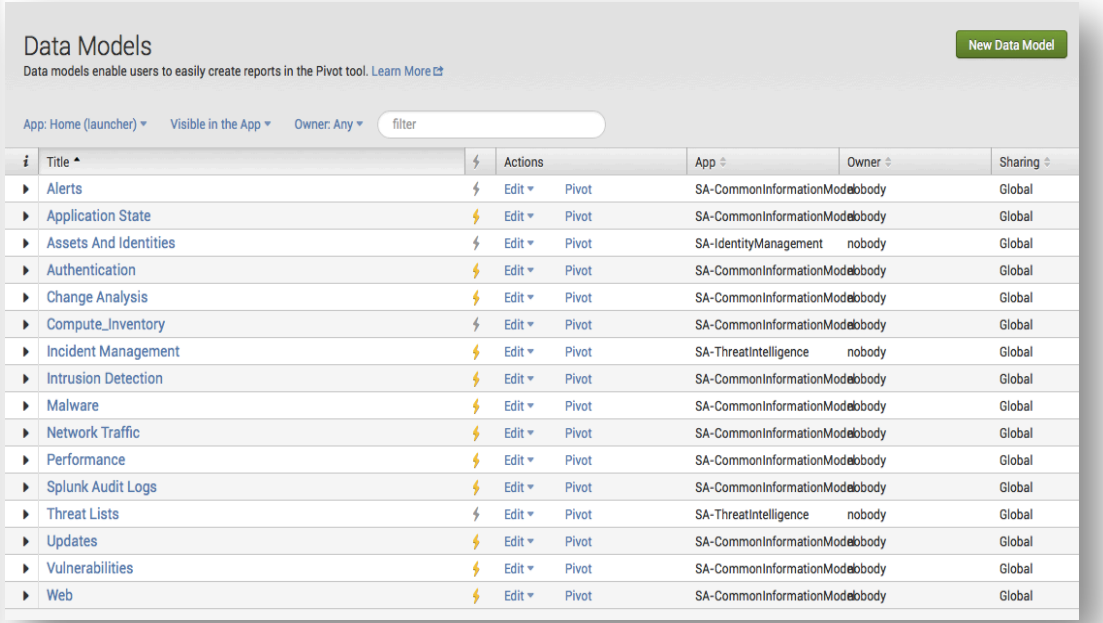


(sourcetype=SYMC “Delete failed”) OR (product=“VirusScan Enterprise” action=would*) OR (SourceName=“Trend Micro OfficeScan Server” “Action: * cannot *”)

- Normalizing at index time is pretty lame
- Normalizing the data before it’s stored is VERY lame
- Normalizing with tags and fields at search time is very AWESOME

Data Models Create Common Understanding

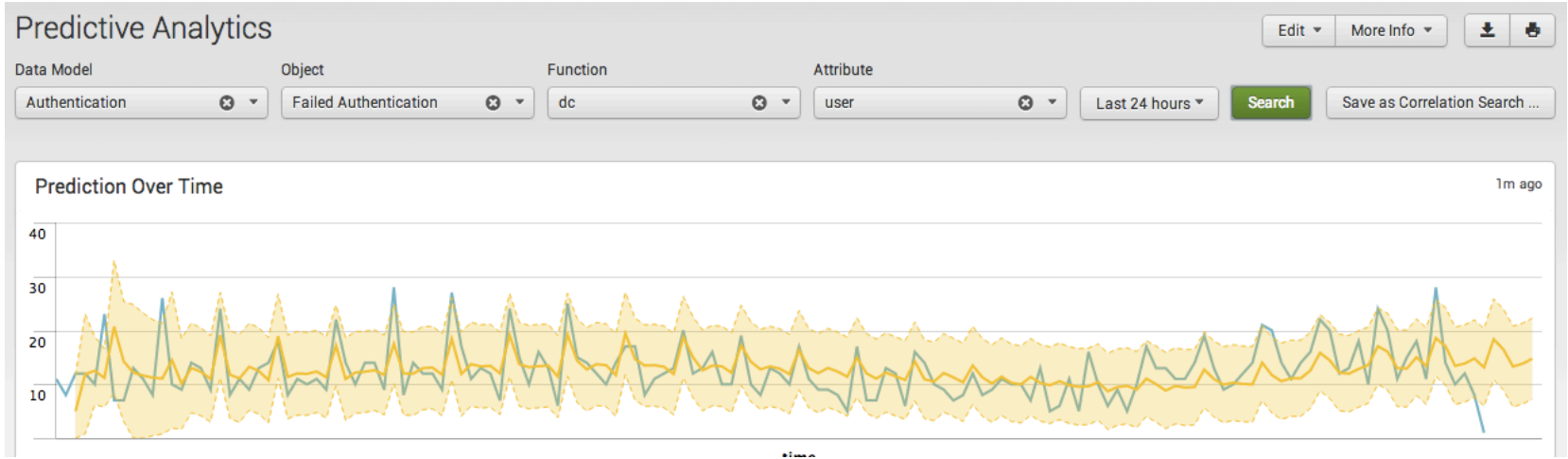
- Normalization without data reduction
- Customized for different data types
- RBAC maintained
- Data model reporting speeds optimized



Make Correlation Searches from Datamodels

Predictive Analytics -> Pointy-Clicky detection of deviations from norm

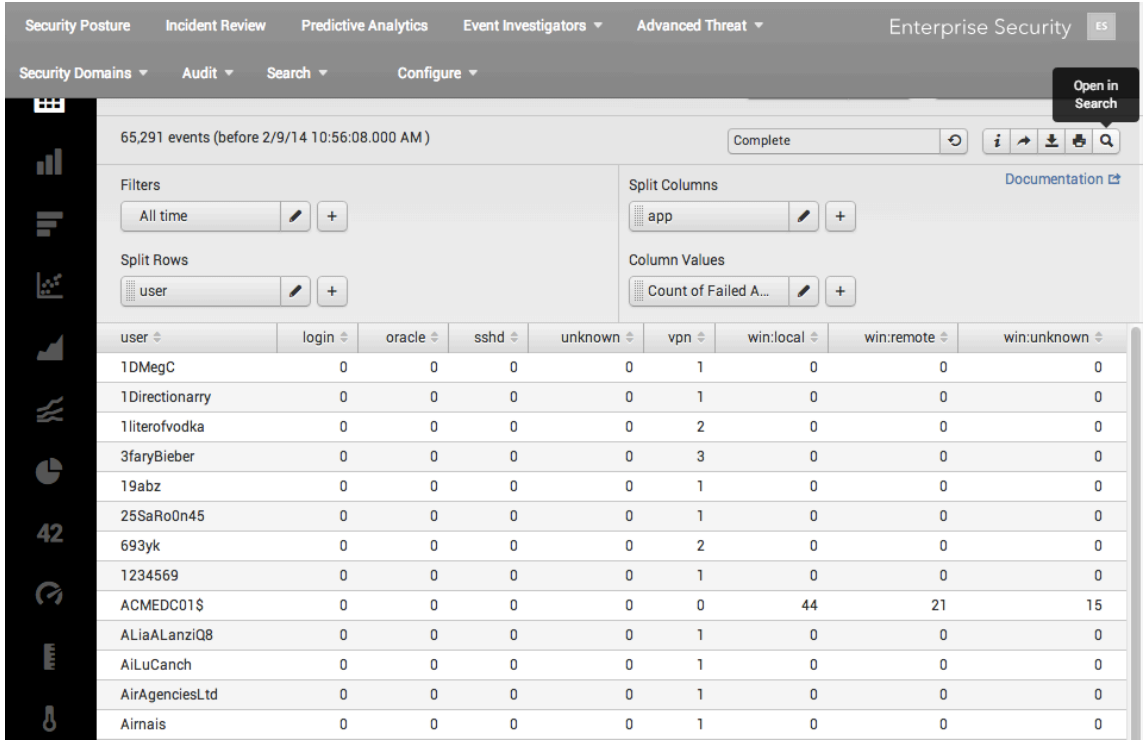
- Events and domains, not individual entities
  - YES: The number of failing accounts is different than normal
  - NO: SPLUNK/JCOATES is failing logins more than normal



Make Correlation Searches from Datamodels

Search:Pivot -> Pointy-Clicky construction of data sets

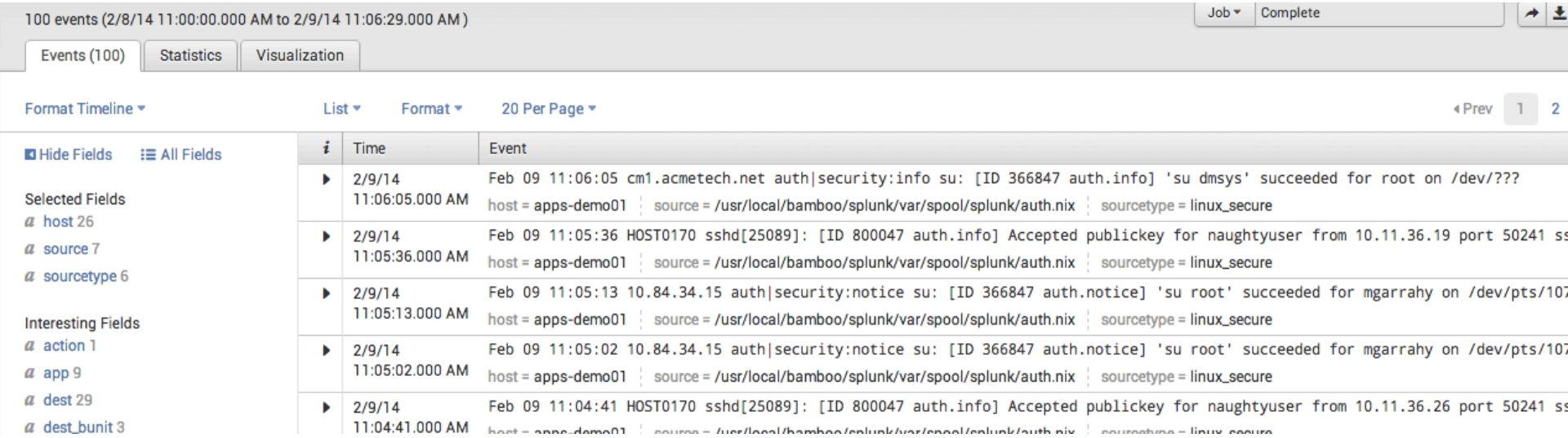
- Gather a collection of normalized data
- Join with another
- Filter for interest
- Paste into New Correlation Search form



user	login	oracle	sshd	unknown	vpn	win:local	win:remote	win:unknown
1DMegC	0	0	0	0	1	0	0	0
1Directionarry	0	0	0	0	1	0	0	0
1literofvodka	0	0	0	0	2	0	0	0
3faryBieber	0	0	0	0	3	0	0	0
19abz	0	0	0	0	1	0	0	0
25SaRo0n45	0	0	0	0	1	0	0	0
693yk	0	0	0	0	2	0	0	0
1234569	0	0	0	0	1	0	0	0
ACMEDC01$	0	0	0	0	0	44	21	15
ALiaALanziQ8	0	0	0	0	1	0	0	0
AiLuCanch	0	0	0	0	1	0	0	0
AirAgenciesLtd	0	0	0	0	1	0	0	0
Airnais	0	0	0	0	1	0	0	0

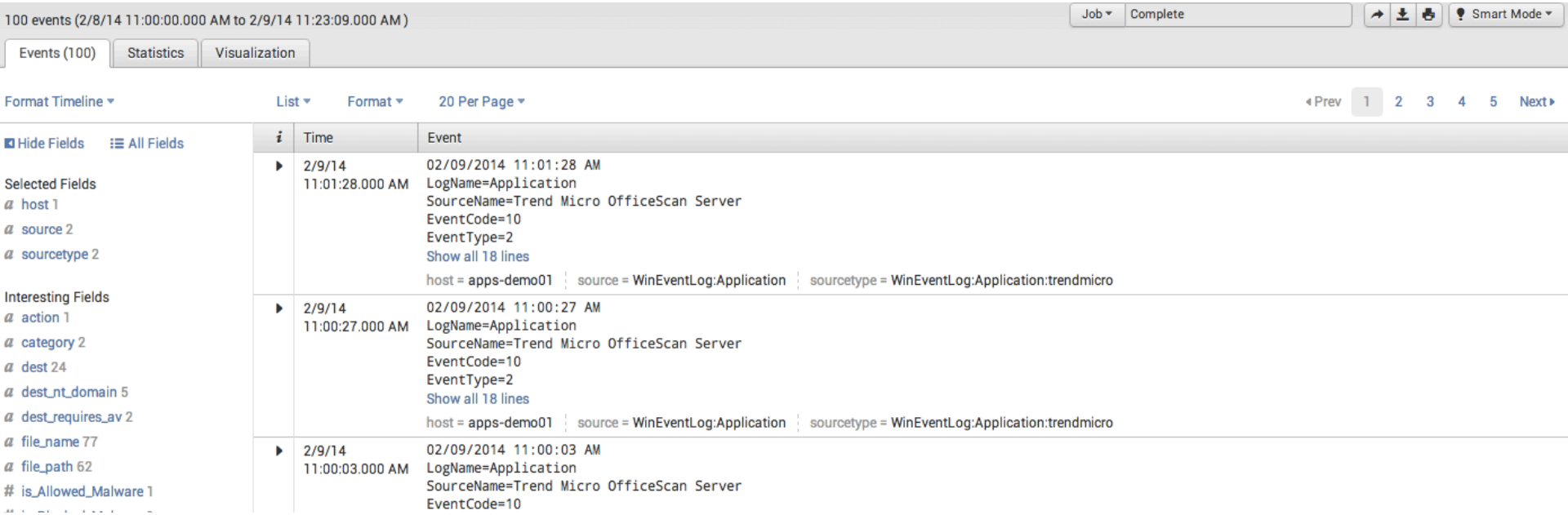
Step One: Authentications

- Authentications, successful, privileged
- `| `datamodel("Authentication", "Authentication")` | search Authentication.action="success" | `drop_dm_object_name("Authentication")` | search is_Privileged_Authentication=1`



Step Two: Malware

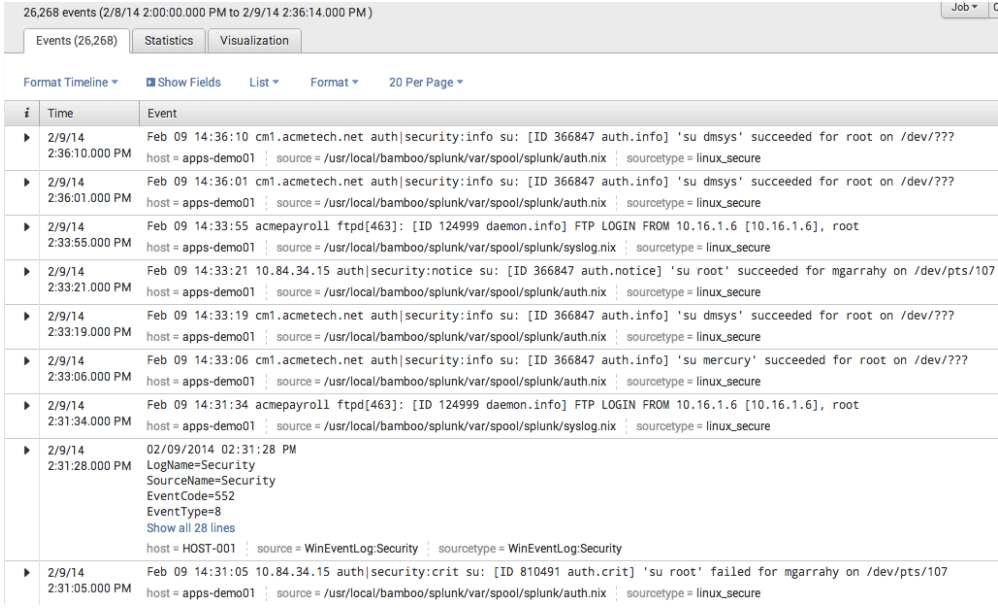
- Malware Attacks, Allowed
- `| `datamodel("Malware", "Malware_Attacks")` | search Malware_Attacks.action="allowed" | `drop_dm_object_name("Malware_Attacks")`



Step Three: Access OR Malware

- Construct the total pool of data to make decisions from

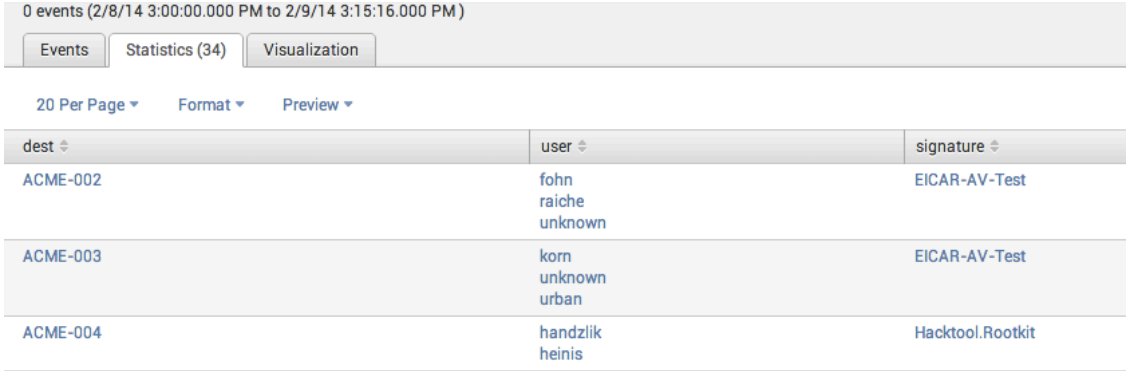
```
| `datamodel("Authentication", "Authentication")`
| search Authentication.action="success"
| `drop_dm_object_name("Authentication")`
| search is_Privileged_Authentication=1
| append [
| `datamodel("Malware", "Malware_Attacks")`
| search Malware_Attacks.action="allowed"
| `drop_dm_object_name("Malware_Attacks")`
]
```



Step Four: Count the Matches

- Tabulate the fields that the decision will be made from

- `| `datamodel("Authentication", "Authentication")` | search Authentication.action="success" | `drop_dm_object_name("Authentication")` | search is_Privileged_Authentication=1 | append [| `datamodel("Malware", "Malware_Attacks")` | search Malware_Attacks.action="allowed" | `drop_dm_object_name("Malware_Attacks")`]`
- `| eval group = case(is_Successful_Authentication==1, "authentication", is_Allowed_Malware==1, "malware")`
- `| stats values(user) as user, values(signature) as signature, distinct_count(group) as group_count by dest`



Step Five: Detect the Condition

- group_count field > 1 is bad, and there's our correlation search
- `| `datamodel("Authentication", "Authentication")` | search Authentication.action="success" | `drop_dm_object_name("Authentication")` | search is_Privileged_Authentication=1| append [| `datamodel("Malware", "Malware_Attacks")` | search Malware_Attacks.action="allowed" | `drop_dm_object_name("Malware_Attacks")`] | eval group = case(is_Successful_Authentication==1,"authentication",is_Allowed_Malware==1,"malware") | stats values(user) as user, values(signature) as signature, distinct_count(group) as group_count by dest`
- `| search group_count>1`
- `| `get_event_id``
- `| fields - group_count`

dest	user	signature
ops-sys-006	unknown	EICAR-AV-Test

Exercise 5
Create and Use Lookups

splunk>

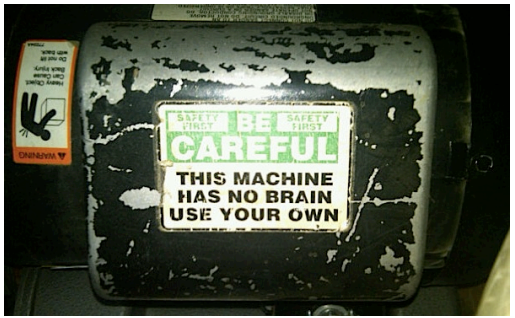
Working with Lookups

- **Some things are impractical to continually discover or recalculate in real time**

Network Traffic Flows are expensive to monitor, so save the important information in a lookup

IP ranges of prohibited sites may be stored in another lookup (several are provided by default)

Did an internal system communicate with an embargoed network?



Generating a Lookup

- Search for the material in question (tstats, raw, whatevs)
- Join with previously discovered lookup contents
- Write the new lookup

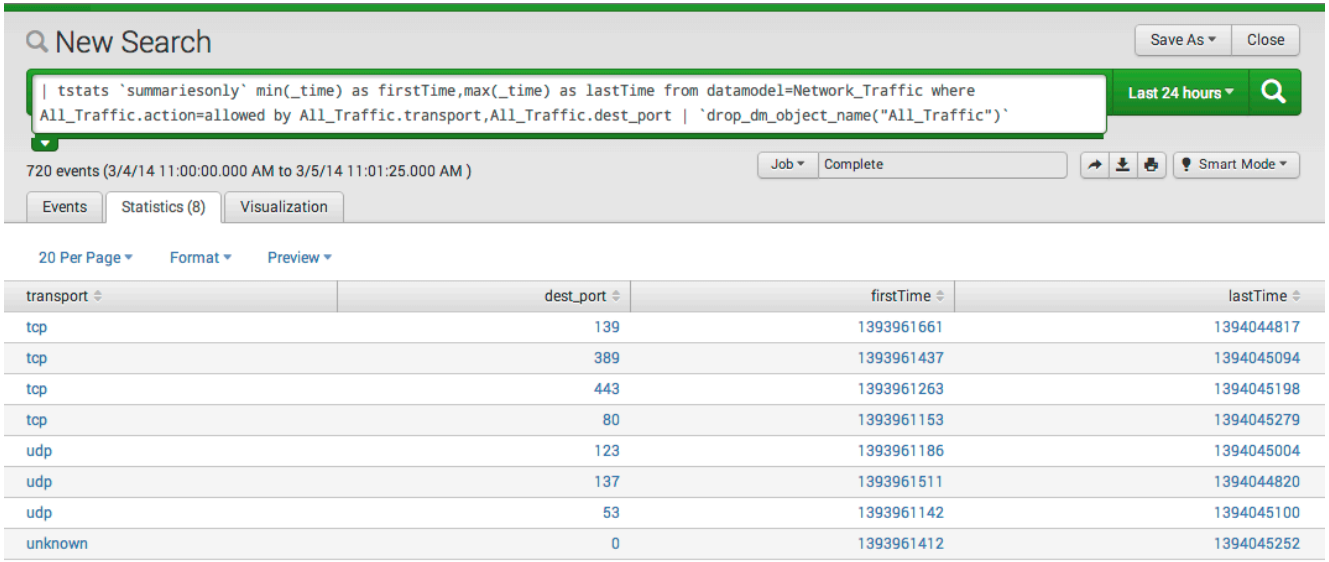
- Example: Port and Protocol Tracker, shows what transport/port combinations are in use on the network over time

```
| tstats `summariesonly` min(_time) as firstTime,max(_time)
as lastTime from datamodel=Network_Traffic where
All_Traffic.action=allowed by
All_Traffic.transport,All_Traffic.dest_port |
`drop_dm_object_name("All_Traffic")` | inputlookup append=T
port_protocol_tracker | stats min(firstTime) as
firstTime,max(lastTime) as lastTime by transport,dest_port
| outputlookup port_protocol_tracker | stats count
```

Search for the Material in Question

- In this case an accelerated data model holds the goods

```
| tstats `summariesonly` min(_time) as firstTime,max(_time) as lastTime from datamodel=Network_Traffic where
All_Traffic.action=allowed by All_Traffic.transport,All_Traffic.dest_port | `drop_dm_object_name("All_Traffic")`
```



transport	dest_port	firstTime	lastTime
tcp	139	1393961661	1394044817
tcp	389	1393961437	1394045094
tcp	443	1393961263	1394045198
tcp	80	1393961153	1394045279
udp	123	1393961186	1394045004
udp	137	1393961511	1394044820
udp	53	1393961142	1394045100
unknown	0	1393961412	1394045252

Join With the Older Lookup

- This is critical because if the new search provides no results, the old lookup will be wiped out

```
| tstats `summariesonly` min(_time) as firstTime,max(_time) as lastTime from datamodel=Network_Traffic where
All_Traffic.action=allowed by All_Traffic.transport,All_Traffic.dest_port | `drop_dm_object_name("All_Traffic")`

| inputlookup append=T port_protocol_tracker
```

Deal With Time

- Order, format, label

```
| tstats `summariesonly` min(_time) as firstTime,max(_time) as lastTime from datamodel=Network_Traffic where
All_Traffic.action=allowed by All_Traffic.transport,All_Traffic.dest_port | `drop_dm_object_name("All_Traffic")`

| inputlookup append=T port_protocol_tracker

| stats min(firstTime) as firstTime,max(lastTime) as lastTime by transport,dest_port
```

- You might also want to dedup events

```
| dedup firstTime,lastTime,transport,dest_port
```

- To keep the lookup from growing forever, you can filter by time and calculate a difference

```
| convert timeformat="%m-%d-%Y" mktime(date) as _time | eval timeDiff=now()-_time | search timeDiff<86400
```

Write the Resulting Lookup File

- Replace the lookup with the new results

```
| tstats `summariesonly` min(_time) as firstTime,max(_time) as lastTime from datamodel=Network_Traffic where
All_Traffic.action=allowed by All_Traffic.transport,All_Traffic.dest_port | `drop_dm_object_name("All_Traffic")`

| inputlookup append=T port_protocol_tracker

| stats min(firstTime) as firstTime,max(lastTime) as lastTime by transport,dest_port

| outputlookup port_protocol_tracker
```

20 Per Page ▾ Format ▾ Preview ▾

dest_port	firstTime	lastTime	transport
139	1386119390	1394044817	tcp
22	1388784710	1392695785	tcp
389	1386119665	1394045094	tcp
443	1386119183	1394045198	tcp
80	1386119185	1394045279	tcp
123	1386119355	1394045004	udp
137	1386119590	1394044820	udp
53	1386119104	1394045100	udp
0	1386119107	1394045252	unknown

Using Lookups in Correlation Searches

The Correlation Three Step

- Gather a pool of data
- Use a table to form the data for easy numeric tests
- Perform tests to make decisions

POOL:

```
| inputlookup append=T src_dest_tracker | lookup local=true ip_tor_lookup src OUTPUTNEW src_ip as src_tor_ip,src_is_tor | lookup local=true ip_tor_lookup dest OUTPUTNEW dest_ip as dest_tor_ip,dest_is_tor | search dest_is_tor=true OR src_is_tor=true
```

TABLE:

297 results in the last 24 hours (from 1:00:00 PM April 29 to 1:47:25 PM April 30, 2012)

Export Options « prev 1 2 3 4 5 6 7 8 9 10 next » 10 per page

Overlay: None

	date	dest	dest_is_tor	dest_tor_ip	sourcetype	src	src_is_tor	src_tor_ip
1	04-30-2012	10.11.36.35			netscreen:firewall	10.11.36.19		
2	04-30-2012	10.11.36.50			netscreen:firewall	10.11.36.29		
3	04-30-2012	10.11.36.45			netscreen:firewall	10.11.36.7		
4	04-30-2012	unknown			airdefense	0b:4a:fe:06:36:92		

The data is already tabular!

TEST:

```
| eval tor_ip=if(dest_is_tor=="true",dest_tor_ip,tor_ip) | eval tor_ip=if(src_is_tor=="true",src_tor_ip,tor_ip) | fields + sourcetype,src,dest,tor_ip
```

FINAL:

```
| inputlookup append=T src_dest_tracker | lookup local=true ip_tor_lookup src OUTPUTNEW src_ip as src_tor_ip,src_is_tor | lookup local=true ip_tor_lookup dest OUTPUTNEW dest_ip as dest_tor_ip,dest_is_tor | search dest_is_tor=true OR src_is_tor=true | eval tor_ip=if(dest_is_tor=="true",dest_tor_ip,tor_ip) | eval tor_ip=if(src_is_tor=="true",src_tor_ip,tor_ip) | fields + sourcetype,src,dest,tor_ip
```

Using a Static Lookup

- You can define a non-dynamic lookup even more easily

Embargo definition, used by a custom block-listed traffic correlation search

```
## SA-ThreatIntelligence/local/transforms.conf
[ip_embargo]
file=embargo.csv
max_matches = 1
match_type = CIDR(src)
```

Embargo.CSV content sample

```
"src","blocked"
"2.144.0.0/14",1
"2.176.0.0/12",1
"31.7.64.0/18",1
"31.7.128.0/20",1
"0.0.0.0/0",0
```

Embargoed Country Activity Correlation Search

```
| inputlookup append=T src_dest_tracker | lookup local=true ip_embargo src
OUTPUTNEW src_ip as src_embargo_ip,src_is_embargoed | lookup local=true
ip_embargo dest OUTPUTNEW dest_ip as dest_embargo_ip,dest_is_embargo | search
dest_is_embargo=true OR src_is_embargo=true | eval
embargo_ip=if(dest_is_embargo=="true",dest_embargo_ip,embargo_ip) | eval
embargo_ip=if(src_is_embargo=="true",src_embargo_ip,embargo_ip) | fields +
sourcetype,src,dest,embargo_ip
```

THANK YOU

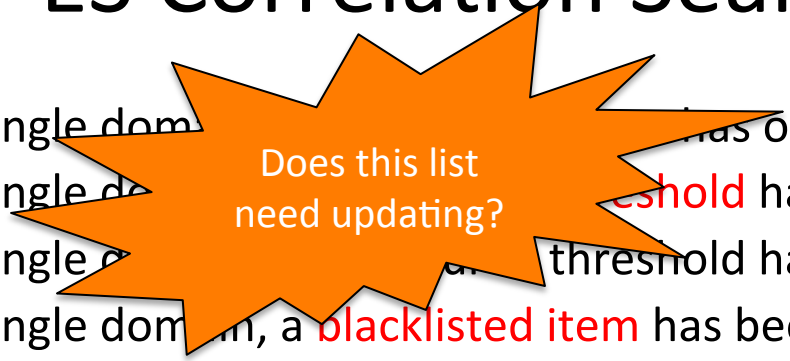
splunk>

Reference Material

Types of Correlation Searches

splunk>

ES Correlation Search Types



- Within a single dom… …as occurred.
- Within a single d… …eshold has been breached once.
- Within a single … …threshold has been breached by many.
- Within a single dom…n, a blacklisted item has been matched.
- Within a single domain, a desired item has not been matched.
- Within a single domain, a known bad event correlation has been matched.
- Within a single domain, a desired event correlation has not been matched.
- Within a single domain, a measured value is anomalous from recorded history.
- Within a single domain, a measured value is anomalous from computed baseline.
- Across multiple domains, a single recognized event has occurred.
- Across multiple domains, a correlation of recognized events has been matched.

Within a Single Domain, a Recognized Event has Occurred

- Threat - Watchlisted Events – Rule

- `tag=watchlist NOT sourcetype=stash | `get_event_id` | `map_notable_fields``

- Events that look bad, add fields and tags.

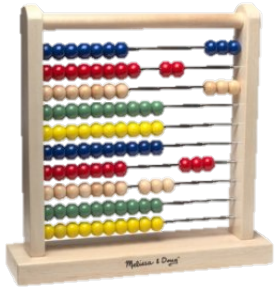
Within a Single Domain, a Configured Threshold has Been Breached Once

- Access - Excessive Failed Logins - Rule

- ```
  `authentication(failure)` | tags outputfield=tag | stats values(tag) as tag,dc(user) as user_count,dc(dest) as dest_count,count by app,src | `settags("access")` | search count>=6
  ```

- Events that look bad, add fields and tags, count them, add fields and tags, test if greater than threshold.






  ```

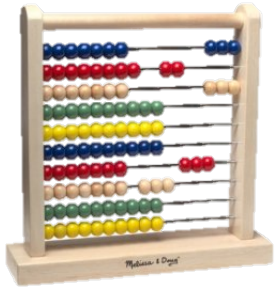
Within a Single Domain, a Configured Threshold has Been Breached by Many

- Network - Vulnerability Scanner Detection (by targets) - Rule

- `tag=attack | tags outputfield=tag | stats values(tag) as tag,dc(dest) as count by src | search count>25 | `settags("network")``

- Events that look bad, add fields and tags, count them, add fields and tags, test if greater than threshold



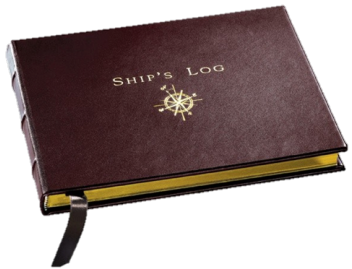



Within a Single Domain, a Blacklisted Item has Been Matched

- Endpoint - Prohibited Process Detection - Rule

- `NOT sourcetype=stash` `localprocesses` | `get_interesting_processes` | search is_prohibited=true | `get_event_id` | `map_notable_fields` | fields + orig_event_id,orig_raw,dest,process,note

- All the useful records, match the blacklist, add fields and tags

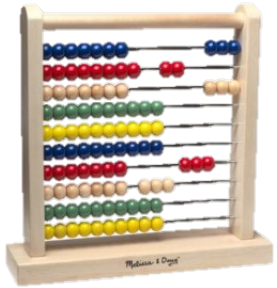


Within a Single Domain, a Desired Item has Not Been Matched

- Audit - Potential Gap in Data - Rule

- `index=_internal * sourcetype="scheduler" status=success (app=SA-* OR app=DA-* OR app=SplunkEnterpriseSecuritySuite OR app=SplunkPCIComplianceSuite) | head 1 | stats count(sourcetype) as count | where count=0 | eval problem="true”`

- Relevant events that are good, count them, test if count = 0

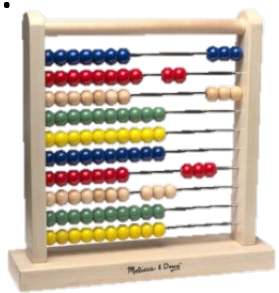


Within a Single Domain, a Known Bad Event Correlation has Been Matched

- Access - Brute Force Access Behavior Detected - Rule

- `authentication` | tags outputfield=tag | stats values(tag) as tag,count(eval(action=="failure")) as failure,count(eval(action=="success")) as success by src | search failure>6 success>0 | `settags("access")`

- Relevant events, count by types, test if two thresholds are breached, add fields and tags.

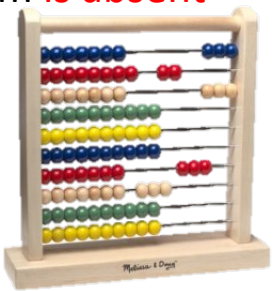




Within A Single Domain, A Desired Event Correlation Has Not Been Matched

- No shipping rule matches this pattern

- N/A

- Customers often will do something like this:
  - Access to important system
  - Change Event on that system
  - Service Desk ticket referencing the system is absent

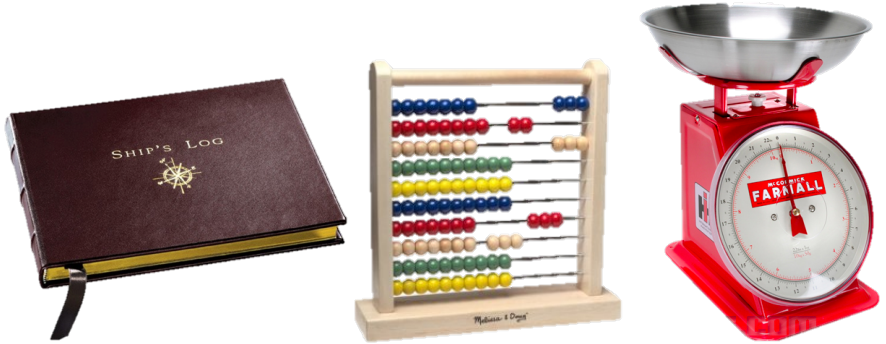




Within a Single Domain, a Measured Value is Anomalous from Recorded History

- Endpoint - Anomalous New Processes - Rule

- `| inputlookup append=T localprocesses_tracker | eval _time=firstTime | \`hoursago(24)\` | stats dc(dest) as dest_count by process | search dest_count>9`

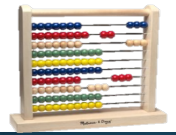
- Load recorded history, bucket by time, count and compare to threshold



Within a Single Domain, a Measured Value is Anomalous From Computed Baseline

- Access - Brute Force Access Behavior Detected – Rule
- ```
  | tstats `summariesonly` count from datamodel=Intrusion_Detection by
  _time,IDS_Attacks.signature span=30m | `drop_dm_object_name("IDS_Attacks")` |
  `timeDiff` | appendpipe [search timeDiff<=86400 | stats max(_time) as
  _time,sum(count) as count by signature | eval group="Last 24 hours"] | eval
  group=if(_time<relative_time(time(),"@d") AND timeDiff<=5184000,"Last 60
  days",group) | bin _time span=1d | stats sum(count) as count by
  _time,group,signature | eval temp=if(group="Last 60 days",signature,null()) |
  eventstats stdev(count) as stdev,avg(count) as avg by temp | eventstats
  max(stdev) as stdev,max(avg) as avg by signature | dedup signature sortby -_time
  | eval limit=(3*stdev)+avg | eval diff=count-limit | search diff>0
  ```

- Relevant events, sort types into a table, test if two thresholds are breached, add fields and tags.



  ```

Across Multiple Domains, a Recognized Event has Occurred

- Threat - Threat List Activity - Rule
- `| `src_dest_tracker("allowed")` | `threatlist_lookup(src)` | `threatlist_lookup(dest)` | eval threat_ip=if(isnotnull(src_threatlist_name), src, if(isnotnull(dest_threatlist_name), dest, threat_ip)) | search threat_ip=* | fields + sourcetype,threat_ip,src,src_threatlist_category,src_threatlist_description,src_threatlist_name,dest,dest_threatlist_category,dest_threatlist_description,dest_threatlist_name`

- Events of the right type, recognized bad things, add fields and tags, filter by match, add more fields and tags.


Across Multiple Domains, a Correlation of Recognized Events has Been Matched

- Access - High or Critical Priority Individual Logging into Infected Machine - Rule
-
```
(tag=authentication action=success (user_is_privileged="true" OR user_priority="critical" OR user_priority="high") ) OR (tag=malware tag=attack action=allowed) | tags outputfield=tag | eval group=case(tag=="authentication","authentication",tag=="malware","malware") | eval user=if(tag=="malware",null(),user) | eval signature=if(tag=="authentication",null(),signature) | stats values(user) as user,values(signature) as signature,dc(group) as group_count by dest | search group_count>1
```

- All of the potentially interesting events, add fields and tags, count and test.