



... and now we can SPL "(?<foo>s[hi]{2}t)"

Mary Cordova

@cyphoid_mary

ShellCon 2020

Something(s) about Mary



- Splunk Trust Member, Splunk Certified Architect
- SIEM 2013-16
@ <insert biggest gaming company you can think of here>
- SOAR 2016-18
@ <insert Hollywood agency for your favorite A-lister here>
- IR 2019-present
@ <insert your 2nd favorite (or maybe 3rd) comic book movie studio here>
- Creds
SANS GIAC⁶, CCNA, SSCP, ISC² Exam Developer
- Education
B.S. Computer Information Systems
- Groups
WSC, DC310, ISSA, ...



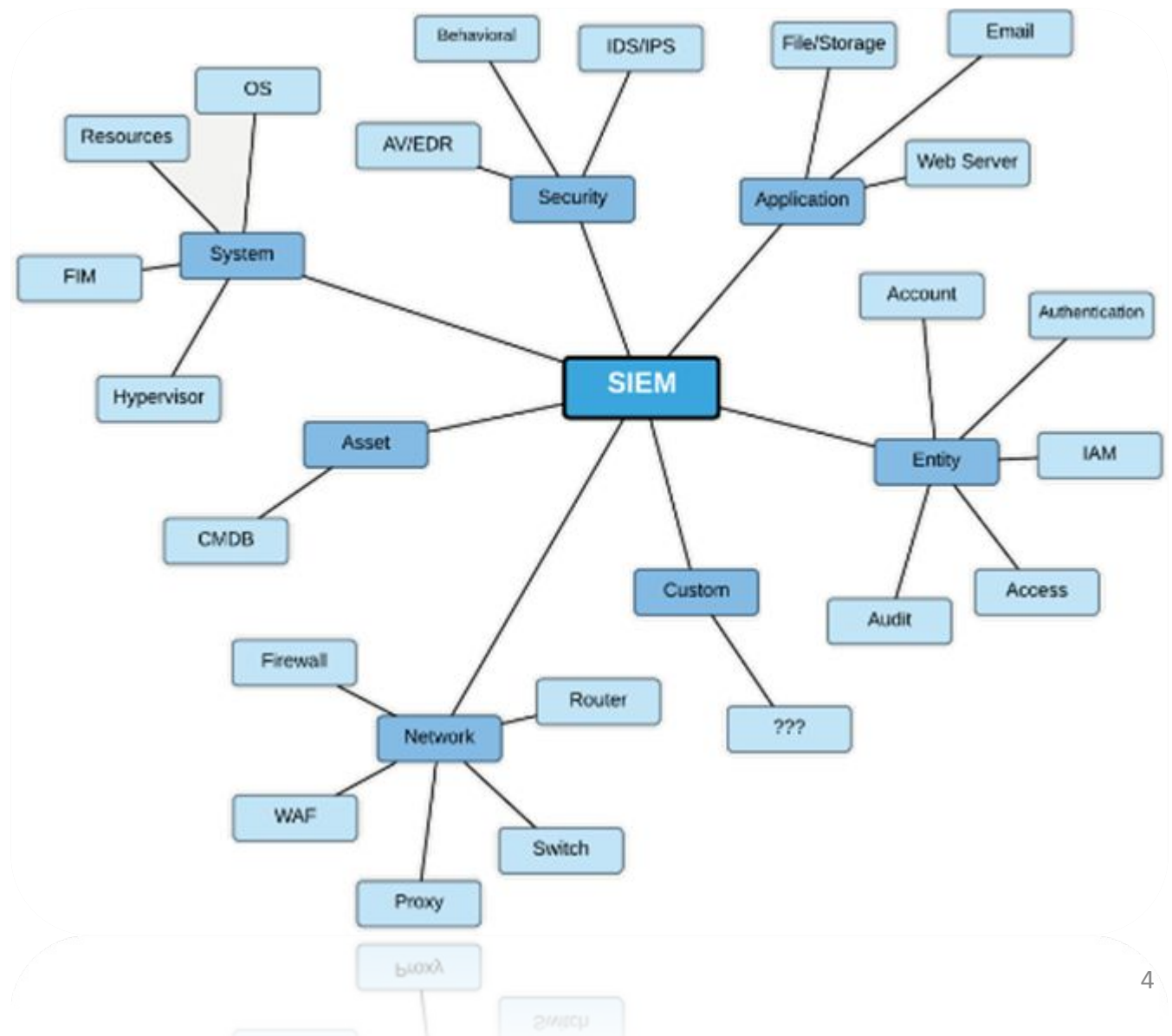
Agenda



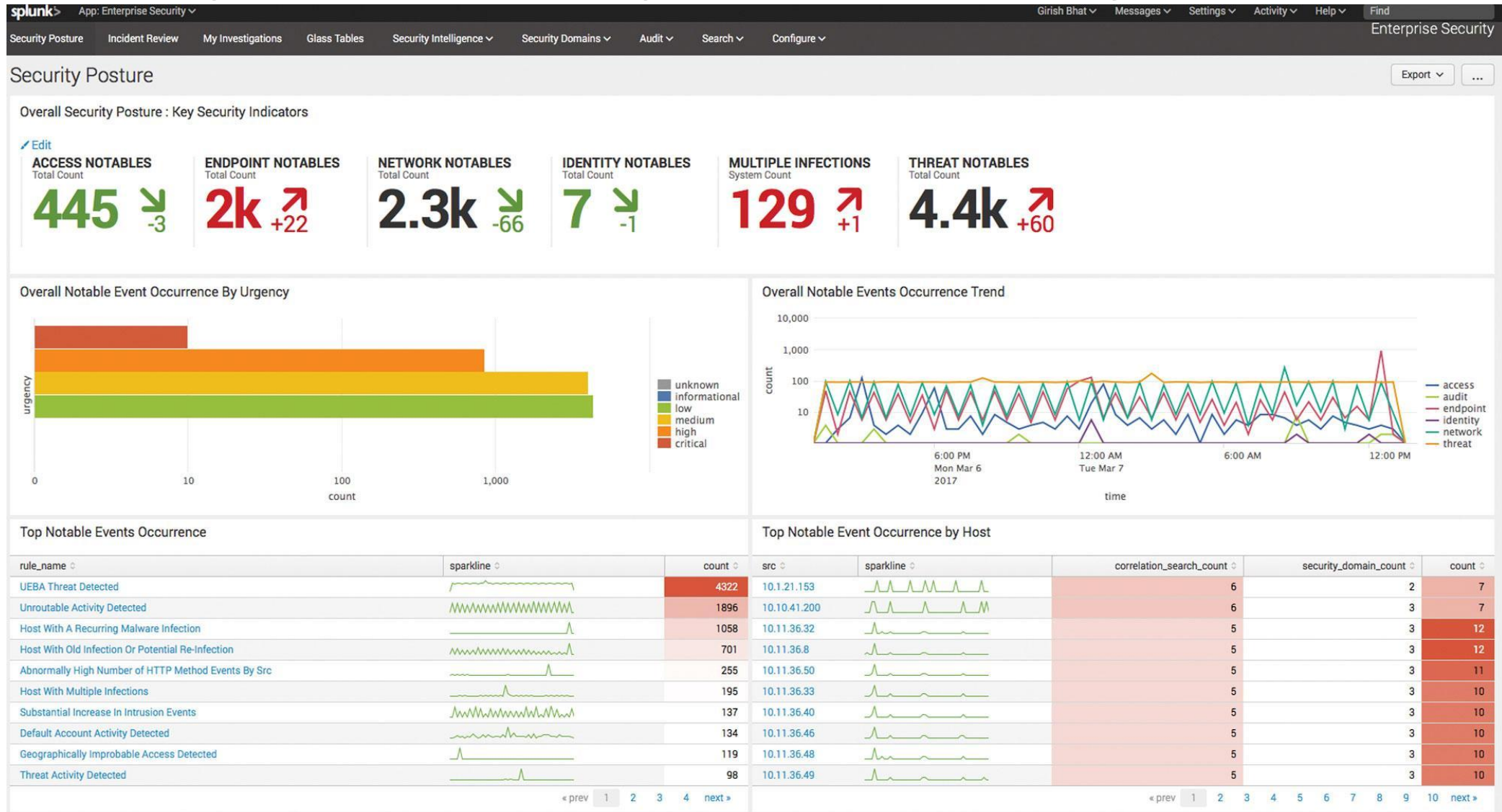
- What is a SIEM?
- Why/how is it used?
- How can you get started?
- Process
- Common problems
- Extra Resources
- Assumptions
 - you probably already have some familiarity w/ Security, SIEM, SOC, IR, data, Splunk

a SIEM has_(logs):

a SIEM does:



Splunk Enterprise Security SIEM



Custom Incident Response Dashboard

SPE Incident Response

Search Indexes and Sourcetypes for Email Data

Email Address

O365

O365

Keyword search supports wildcards and boolean: *, AND, OR, etc

Keyword

User DHCP

WinEvent Log

Maximum currently calculated risk score (out of possible 11)

5

Currently included datasets:

- User has Admin access &/or User is VIP

User attribute search: email address preferred

Email or User ID

Password Last Set:

Account Status:

Has Admin Access:

Is VIP Account:

Risk Score (Max of 11)

2020-07-23

active

yes

no

2

sid

uid

status

created

disabled

pwd_set

type

name

mail

title

phone

mobile

manager

region

country

office

company

department

division

alias

Splunk training

- free courses offered by Splunk
- fundamentals 1 if you're mostly a user/searcher/data person

Training + Certification

Free Courses ^

[Overview](#)

[Free Splunk Fundamentals 1](#)

[Splunk Infrastructure Overview](#)

[Splunk User Behavior Analytics](#)

[SignalFx Fundamentals Series \(eLearning\)](#)



Should you put your data in Splunk?

- Is it machine data with events of interest during an incident?
- Are there events that should be monitored because they indicate something bad could be happening?
- Does your data provide context that could be useful in an investigation?

-
- You have your data in Splunk...now what?!

New Search

| enter search here...






Process



- Find your data
- Clean/normalize your data
- Save “base” searches
- Develop analytics, reports, dashboards, alerts

Finding your data

index=?? sourcetype=??

- Choose something unique from your data source that you can search for in Splunk
- Something you can generate OR something that you **know** (not think) already occurred
 - We will keyword search for the generated locating data “pretty please”
- `index=* sourcetype=* keyword`
 - alternatively, if you know something of the architecture
`| tstats count WHERE index=* by index sourcetype`
- Found your data?
 - Immediately stop using **`index=* sourcetype=*`**
- After we have located our data we can:
 - Clean our data  
 - Build a base search  
 - Develop analytics 
- Getting a **good** base search can take time, frequently a full days’ worth of work at least and often more

Building your SPL*



*Search Processing Language

- Don't start with fancy SPL
- Don't restrict your search with fields at first
- Don't run it over a large time range
- Start with "Verbose Mode"
- Incrementally define your search
- Start with "keyword" searches then build faster indexed "field" searches
- As you narrow the scope of the data you can expand your time window
- "ctrl+\" for nice formatting

Cleaning/normalizing your data



- Iterate removing noise from the data using “| fields - field field field...”
- Normalize remaining fields (and values where appropriate) with CIM (Common Information Model)
 - **src_ip=#.#.#.#**
 - “source_ip” or “source_address” or “src_address” etc
 - **src_mac=aa:bb:cc:00:11:22**
 - not “AA-BB-CC-00-11-22” or “aabbcc001122” etc
- You should end up with a nice list of normalized 10-20 fields with the most important values in your data
- This is a good base search that can be used over and over for various analytics

```
index=* sourcetype=* frosting.com|
```

```
index=[REDACTED] sourcetype=pan:threat frosting.com|
```

```
index=[REDACTED] sourcetype=pan:threat frosting.com  
| fields - date* eventtype host index linecount punct source sourcetype splunk_server
```

```
index=[REDACTED] sourcetype=pan:threat frosting.com  
| fields - date* eventtype host index linecount punct source sourcetype splunk_server  
pcap_id product receive_time repeat_count rule *_number *_session* tunnel* url_* v  
| table _time *
```

```
index=[REDACTED] sourcetype=pan:threat frosting.com  
| fields - date* eventtype host index linecount punct source sourcetype splunk_server*  
vendor* receive_time repeat_count rule *_number tunnel* misc version virtual* vsys*  
src dest signature_id raw_category dest_hostname client* server* dest_name log_sub  
| table _time action type severity cat category threat* signature verdict transport to
```

Gotchas

INTERESTING FIELDS

```
a action 2
a cat 1
a category 59
a dest_class 2
a dest_ip 100+
a dest_location 37
# dest_port 31
a dest_zone 5
a direction 1
a file_name 100+
a http_content_type 81
a http_method 7
a http_referrer 100+
a http_user_agent 100+
a severity 1
a src_class 2
a src_ip 100+
a src_location 22
# src_port 100+
a src_user 100+
a src_zone 18
a threat 1
# threat_id 1
a transport 1
a type 1
a url 100+
a user 100+
```

579 more fields

- Starting small with a keyword makes the job manageable **but is not comprehensive enough to make assumptions about the broader data set**

- initially we get ~25 good fields for further normalization
- We removed ~60 fields full of noise
- Removing our keyword to get a sample of **all** data within our time range is an ugly surprise O_o



The [Admin Guide](#) for your data source can help you identify fields to group different types of events so that you can work on smaller logically similar sets of data one at a time

You need several samples of each type of event so that you not only have representation of the different types but the different data values that can be found in each of those types

Subtype of threat log. Values include the following:

- data—Data pattern matching a Data Filtering profile.
- file—File type matching a File Blocking profile.
- flood—Flood detected via a Zone Protection profile.
- packet—Packet-based attack protection triggered by a
- scan—Scan detected via a Zone Protection profile.
- spyware —Spyware detected via an Anti-Spyware pro
- url—URL filtering log.
- virus—Virus detected via an Antivirus profile.
- vulnerability —Vulnerability exploit detected via a Vuln
- wildfire —A WildFire verdict generated when the firew verdict (malicious, phishing, grayware, or benign, dep Submissions log.
- wildfire-virus—Virus detected via an Antivirus profile.

If you're cleaning, don't worry about your SPL'ing



- Whoa...that search looks terrible!!!
- Too many |fields and too many |table commands!!!
- Don't worry about that right now, you're just cleaning up and organizing our data, you'll clean up and organize your SPL next

```
index=[redacted] sourcetype=pan:threat
| fields - date* eventtype host index linecount punct source sourcetype splunk_server*
          vendor* receive_time repeat_count rule *number tunnel* misc version virtual* vsys*
          src dest signature_id raw_category dest_hostname client* server* dest_name log_source
| table _time type action severity cat category threat transport direction src_user
| eval type=null()
| rename cat as type
| rex field=src_user "(?<src_user>[^\s]*$)"
| rex field=dest_user "(?<dest_user>[^\s]*$)"
| eval user=mvdedup(mvsort(mvappend(mvappend('user','src_user'),'dest_user'))))
| eval src_location=if(match('src_location','\d+\.\d+\.\d+\.\d+[-]\d+\.\d+\.\d+\.\d+'),null(),
| eval src_location=if(match('dest_location','\d+\.\d+\.\d+\.\d+[-]\d+\.\d+\.\d+\.\d+'),null(),
| fields - cat *user
| table _time type action severity category threat transport direction user src_location
```

```
index=[redacted] sourcetype=pan:threat
| eval type=null()
| rename cat as type
| rex field=src_user "(?<src_user>[^\s]*$)"
| rex field=dest_user "(?<dest_user>[^\s]*$)"
| eval user=mvdedup(mvsort(mvappend(mvappend('user','src_user'),'dest_user'))))
| eval src_location=if(match('src_location','\d+\.\d+\.\d+\.\d+[-]\d+\.\d+\.\d+\.\d+'),null(),
| eval src_location=if(match('dest_location','\d+\.\d+\.\d+\.\d+[-]\d+\.\d+\.\d+\.\d+'),null(),
| table _time type action severity category threat transport direction user src_location src_zone src_ip src_port dest_location dest_zone dest_ip dest_port http_method http_referrer url uri_path http_content_type http_user_agent
```


Base search - one more time for the crowd in the back



- Don't start with fancy SPL
- Don't restrict your search with fields at first
- Don't run it over a large time range
- Start with "Verbose Mode"
- Incrementally define your search
- Start with "keyword" searches then build indexed "field" searches
- As you narrow the scope of the data you can expand your time window
- "ctrl+\\" for nice formatting
- Do build your SPL up line by line
- Keywords become field=value pairs
- Less keyword and more field=value means you can search larger time ranges
- Add normalization to well scoped base searches
- Save base searches for all your data sets
- Use base searches to build analytics
- Run finalized analytics in "Fast Mode"

Common problems



- hey Mary, my search isn't working!
 - duplicate tab
 - delete all your lines
 - add lines **ONE** by **ONE**, run your search
 - inspect the output of the fields that aren't doing what you want

- hey Mary, how do I know which fields to use?
 - go back to slide 9-13
 - build slide 6 unless you like doing the same thing over and over

Thanks!!!



- This wasn't really finished, hope it went ok!
- If you're weak on regular expressions pick up "Sams Teach Yourself Regular Expressions in 10 Minutes"
 - you can get by with only reading like half the book and using the quick guide in the back :D