Lessons Learned From Building A Threat Detection Program

Chris Ogden
Principal Threat Detection Engineer | Sony

Drew Guarino
Senior Threat Detection Engineer | Sony

splunk> .conf19

Lessons Learned From Building A Threat Detection Program

Chris Ogden
Principal Threat Detection Engineer | Sony

Drew Guarino
Senior Threat Detection Engineer | Sony

splunk> .conf19

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. 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, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Legal Disclaimer

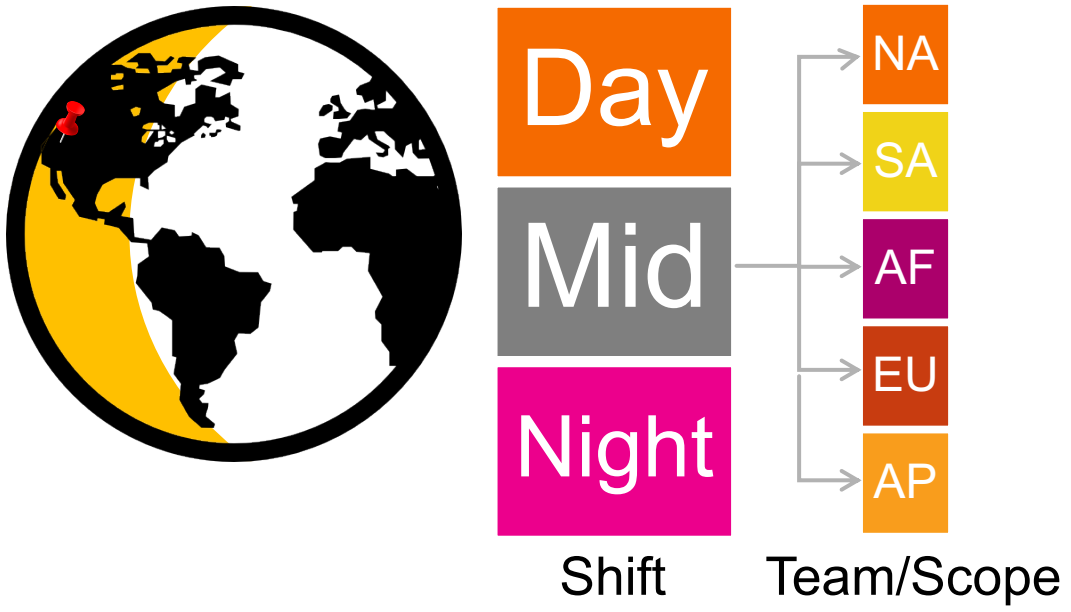
Our presentation and remarks are in a personal capacity representing our own views, opinions, and experiences. Our statements do not reflect the views, positions, or activities of any Sony Group company.

Introduction

Organizational Impact of Centralized Monitoring

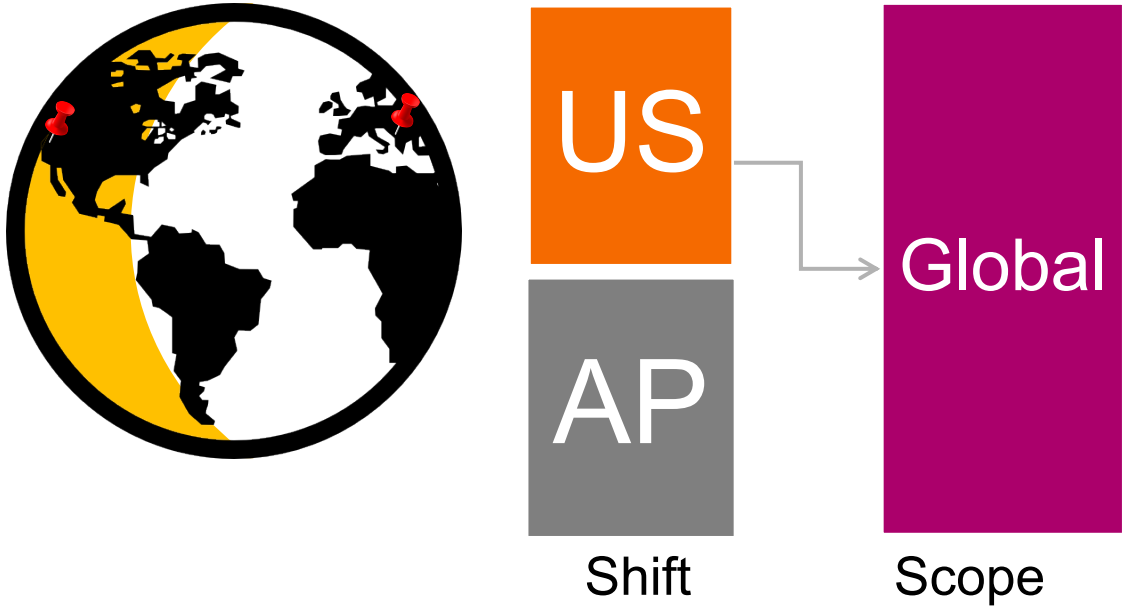
De-centralized Model

- Multiple teams
- Regional scope
- Shift-Work
- Vendor dependent



Centralized Model

- Fewer teams
- Global scope
- “Follow The Sun”
- Custom Detections



Overview

Lesson Learned #1: APT – Admin Persistent Threat

Lesson Learned #2: Scalable Maintenance & Macro Usage

Lesson Learned #3: Working With Timestamps

Lesson Learned #4: Standardizing Fields

Lesson Learned #5: Leveraging ES Frameworks

Lesson Learned #6: Data Hygiene

#1: Admin Persistent Threat

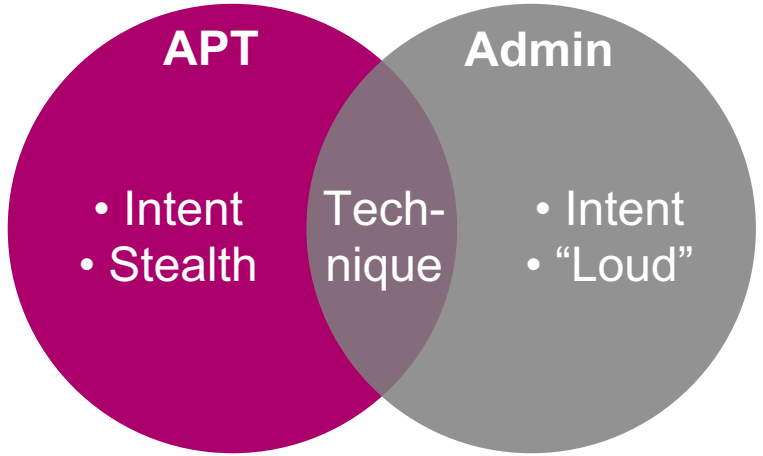
Distinguishing between admins & threat actors

Considerations for a successful Threat Detection Program

Environment Configuration

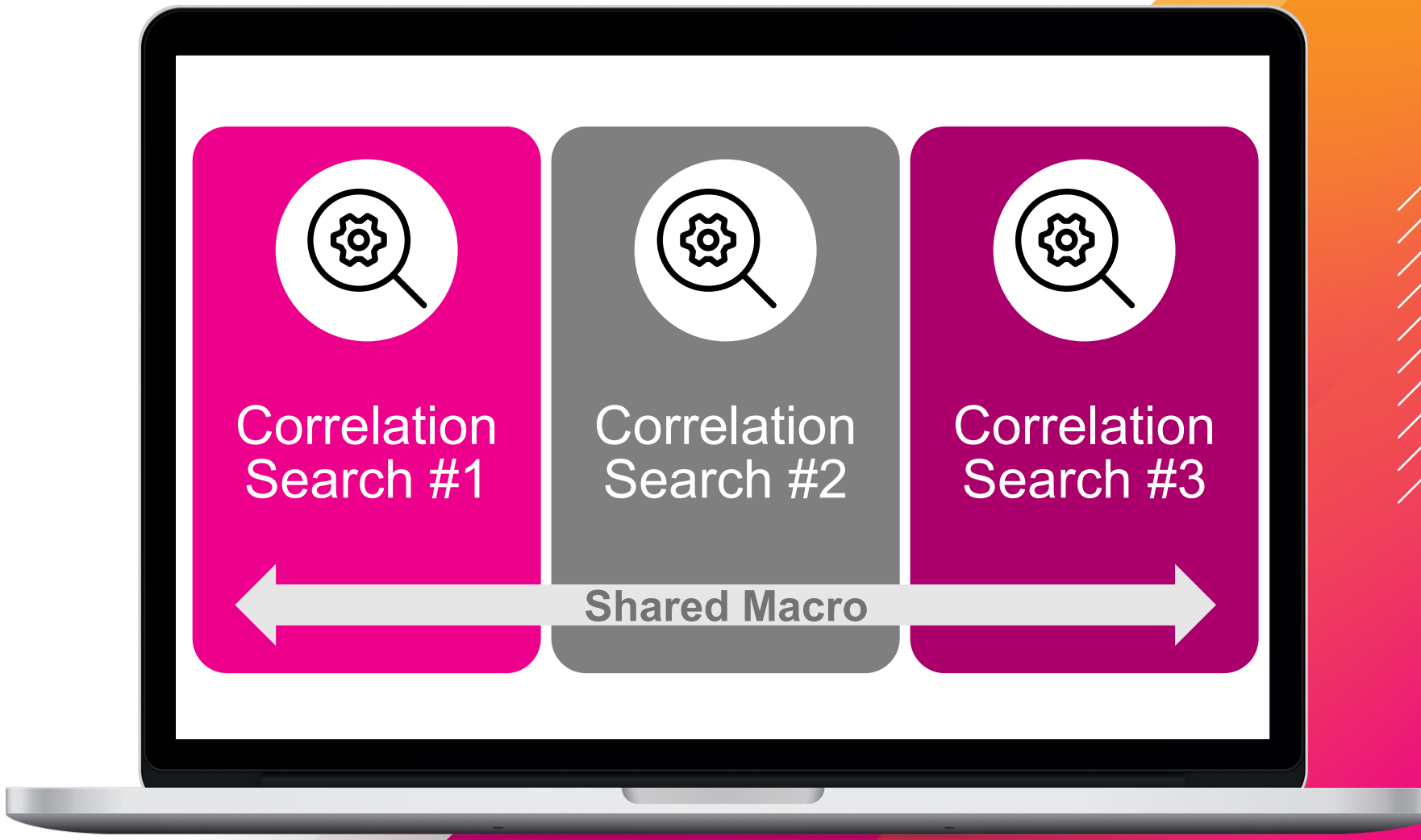
Noise Levels

User Base



#2: Scalable Maintenance & Macro Usage

Consistency, Scalability, User-Friendly



#2: Scalable Maintenance & Macro Usage

Consistency, Scalability, User-Friendly

All notables now have 3 new fields:

- action
- signature
- user

Advanced search » Search macros » add_aws_context

Definition * Enter the string the search macro expands to when it is referenced in another search. If arguments are included, enclose them in dollar signs. For example: $arg1$

```
| eval action = case(
    lower(blocked)="true","Blocked",
    lower(blocked)="false","Allowed",
    isnull(blocked),"Unknown"),

    signature = coalesce(signature,'detail.type',"N/A"),

    user = case(
    isnull(user) AND 'userIdentity.type'="IAMUser",'userIdentity.userName',
`comment("Normal user")`
    isnull(user) AND
'userIdentity.type'="AssumedRole",mvindex(split('userIdentity.arn',"/"),-1),
`comment("Temporary Credentials - User can be derived from arn")`
    isnull(user) AND 'userIdentity.type'="Root" AND NOT 'userIdentity.userName' = "Root"
AND NOT 'userIdentity.userName'="",'userIdentity.userName',    `comment("Root (Alias)
Credentials")`
isnull(user) AND 'userIdentity.type'="Root" AND
isnull('responseElements.accessKey.userName'),mvindex(split('userIdentity.arn',":"),-1),
`comment("This is usually 'root', even in the ARN like above. But I extract it, rather
than use a static string just in case it's ever somehow not.")`
    isnull(user) AND 'userIdentity.type'="Root" AND
isnotnull('responseElements.accessKey.userName'),'responseElements.accessKey.userName',
`comment("There is an obscure exception to the rule above.")`
    isnotnull(user) AND NOT user="", user, `comment("This field has been populated
elsewhere already.")`
    1=1,"N/A"
    ),
```

#2: Scalable Maintenance & Macro Usage

Consistency, Scalability, User-Friendly

Fieldnames (Before Macro)

Field
detail.resource.instanceDetails.networkInterfaces{}.networkInterfaceId
detail.resource.instanceDetails.networkInterfaces{}.privateDnsName
detail.resource.instanceDetails.networkInterfaces{}.privateIpAddress
detail.resource.instanceDetails.networkInterfaces{}.privateIpAddresses{}.privateDnsName
detail.resource.instanceDetails.networkInterfaces{}.privateIpAddresses{}.privateIpAddress
detail.resource.instanceDetails.networkInterfaces{}.publicDnsName
detail.resource.instanceDetails.networkInterfaces{}.publicIp
detail.resource.instanceDetails.networkInterfaces{}.securityGroups{}.groupId
detail.resource.instanceDetails.networkInterfaces{}.securityGroups{}.groupName
detail.resource.instanceDetails.networkInterfaces{}.subnetId
detail.resource.instanceDetails.networkInterfaces{}.vpcId
detail.resource.instanceDetails.platform
detail.resource.instanceDetails.productCodes{}.productCodeId
detail.resource.instanceDetails.productCodes{}.productCodeType
detail.resource.instanceDetails.tags{}.key
detail.resource.instanceDetails.tags{}.value
detail.resource.resourceType

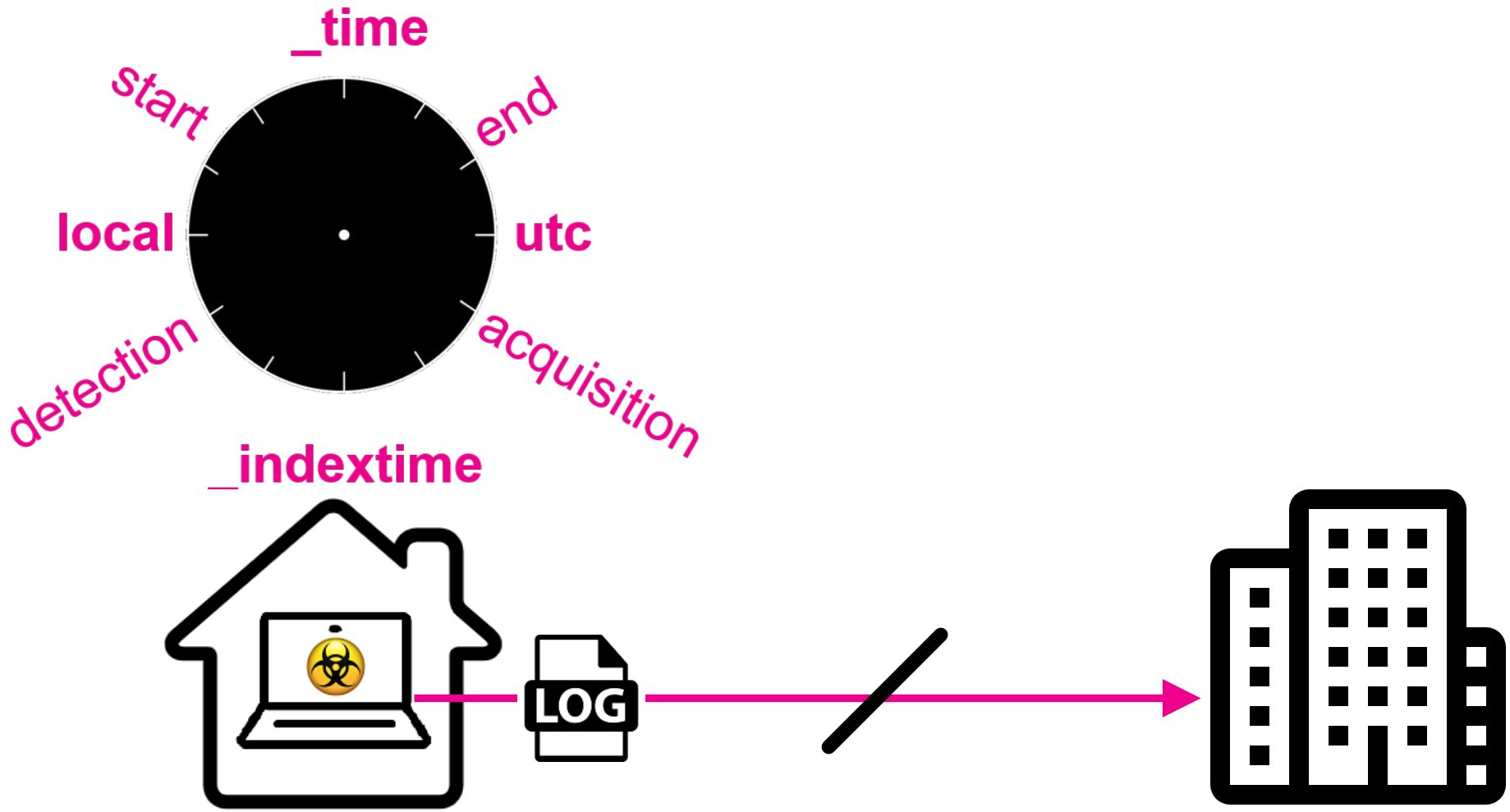
Fieldnames (After Macro)

Field
host
source
sourcetype
account
action
app
blocked
category
cidrIp
dest
dest_port
detail.description
detail.title
detail.type
index
instanceId
src
src_port
user
vpcId

splunk> .conf19

#3: Working with Timestamps

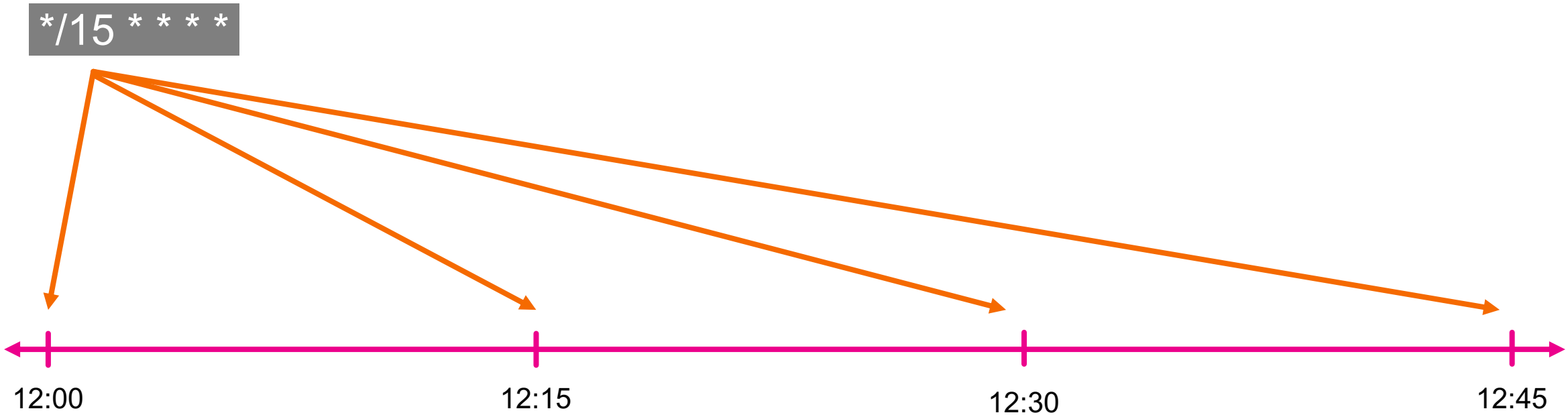
Trust but verify. Beware of delayed events



#3: Working with Timestamps

Gotchas when using _time

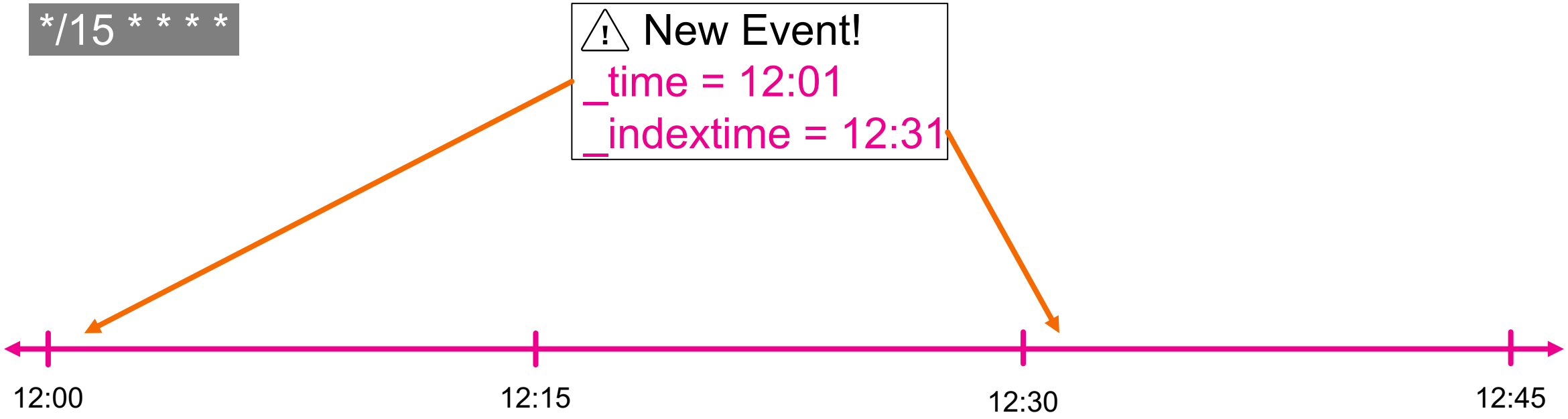
index=AV category=malware action=allow



#3: Working with Timestamps

Gotchas when using _time

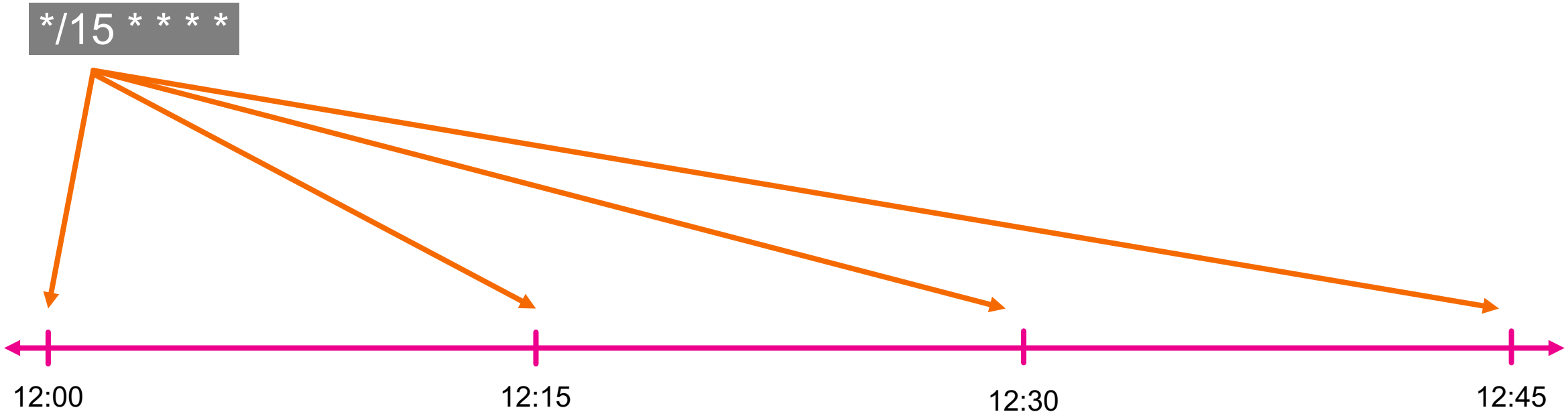
index=AV category=malware action=allow



#3: Working with Timestamps

Gotchas when using _time

_index_earliest=-15m@m index=AV category=malware action=allow



#3: Working with Timestamps

Gotchas when using _indextime

_index_earliest=-15m@m ... <your search>

```
*/15 * * * *
```

#3: Working with Timestamps

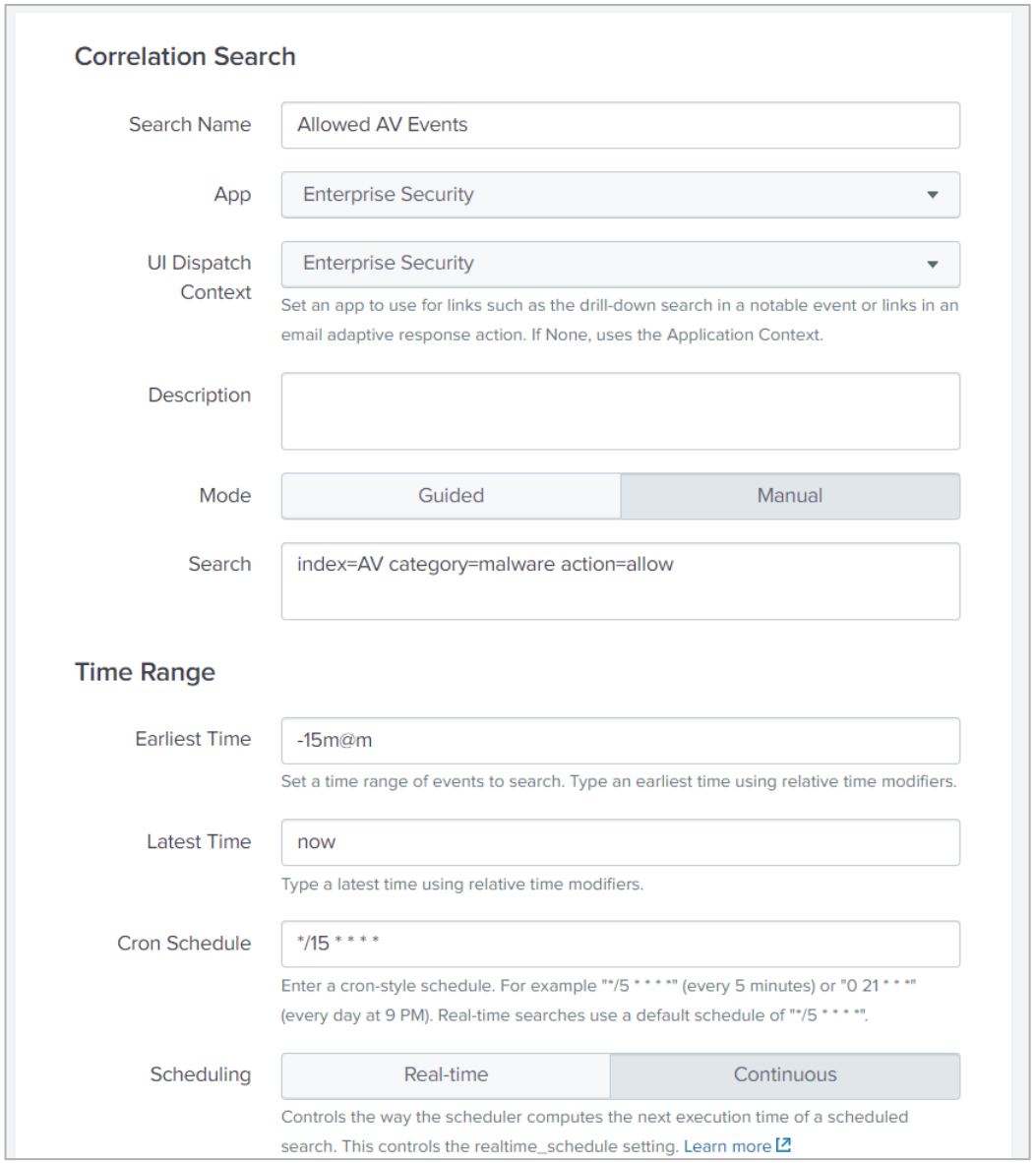
Gotchas when using _indextime

_index_earliest=-15m@m ... <your search>

```
*/15 * * * *
```

#3: Working with Timestamps

Gotchas when using _time



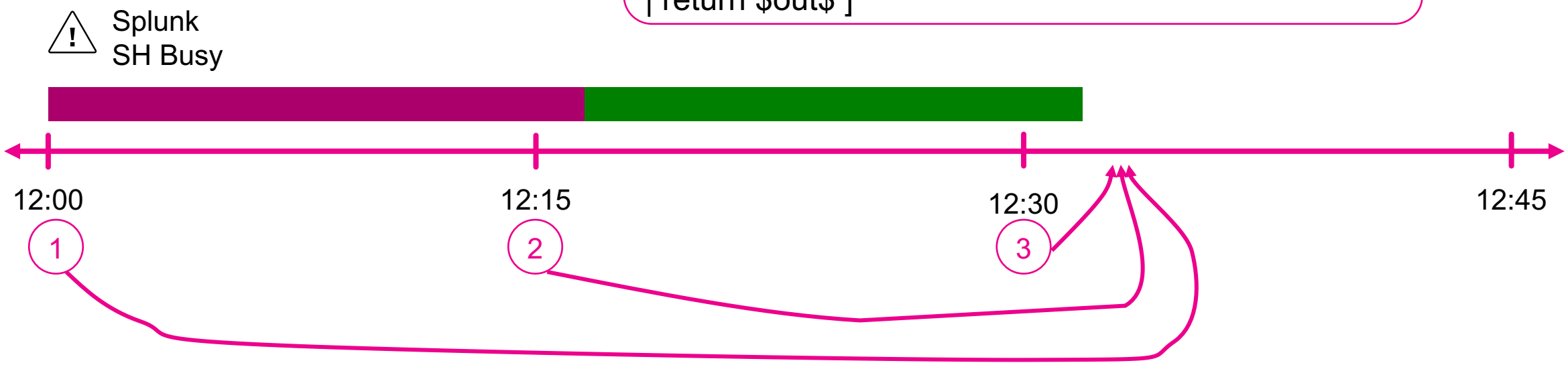
#3: Working with Timestamps

Gotchas when using _indextime

_index_earliest=`set_earliest(-15m@m)` ... <your search>

```
*/15 * * * *
```

```
[ | makeresults
| addinfo
| eval out=relative_time(info_max_time, "$time$")
| return $out$ ]
```



#4: Standardizing Fields

Setting Up Your Detection Program For Success

Why?

- Find what you are looking for in index=notable
- Integrate your alerts with ES frameworks
- Writing macros is easier
- Generate metrics
- All the same reasons you would use CIM (Common Information Model)

#4: Standardizing Fields

Setting Up Your Detection Program For Success

Decide on a naming strategy

Identify key pieces of information you always want to have

- Split it up by broad categories - users, computers, files, emails, cloud objects, etc.

Document it

Adhere to it

#4: Standardizing Fields

Setting Up Your Detection Program For Success

Object Type	Field	Required?	Note
All	signature	Yes, if applicable	Only required if it is a pass-thru alert
All	bunit	No	Business unit name
All	index	Yes	
All	sourcetype	Yes	
All	dvc	Yes	Created by `add_reqd_fields()` macro. Generic product name that is the source of the data (e.g. "Windows Event Logs").
All	special_time	Yes	Created by `add_reqd_fields()` macro
All	orig_raw	Yes	Created by `add_reqd_fields()` macro
System	src / dest	Yes	There should always be a src and/or dest field, it will be a copy of one of the other fields for devices (e.g. src_ip or src_host)
System	src_ip / dest_ip	Yes, if applicable	IPv4 and IPv6 are both acceptable
System	src_host / dest_host	Yes, if applicable	Host name or FQDN acceptable
System	src_mac / dest_mac	Yes, if applicable	If you have a MAC address, please call the `format_mac` macro to ensure formatting is consistent
System	md5 / sha1 / sha256	Yes, if applicable	
User	user	Yes	AD username such as "smithj" or "DOMAIN\smithj". If there is only a src OR dest user available, then use this field for either. If there is **both** a source and target/destination user available, then this should represent the **destination** user.
User	src_user	Yes, if applicable	AD username such as "smithj" or "DOMAIN\smithj". If there is both a source and target/destination user available, then this field should represent the source user.
User	email	Yes, if applicable	If you have an email address, please call the `format_email()` macro to ensure formatting is consistent

#5: Leveraging ES Frameworks

Build out frameworks **early**!

Assets & Identities

- Make ES your single pane of glass

Type	✓	Field	Value
Event	☐	user ▾	JSmith
	☐	user_buint ▾	CISO
	☐	user_category ▾	Security Analyst
	☐	user_email ▾	john.smith@example.com
	☐	user_identity ▾	jsmith
			john.smith@example.com

#5: Leveraging ES Frameworks

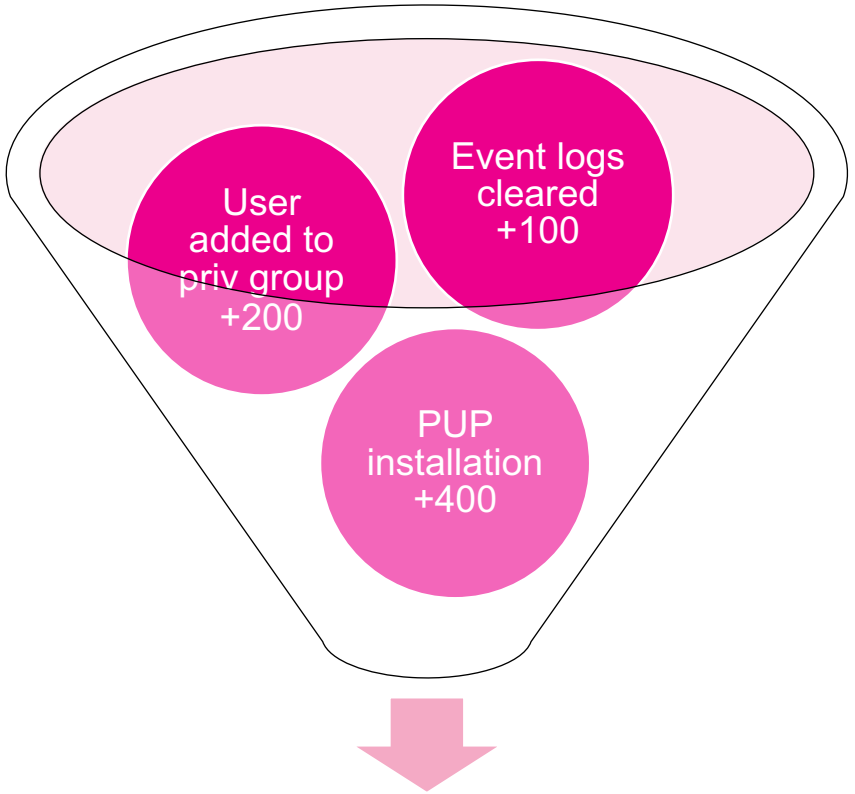
Build out frameworks **early**!

Assets & Identities

- Make ES your single pane of glass

Risk

- Get more value out of your detection content



Risk Score for Object = 700

#5: Leveraging ES Frameworks

Build out frameworks **early**!

Assets & Identities

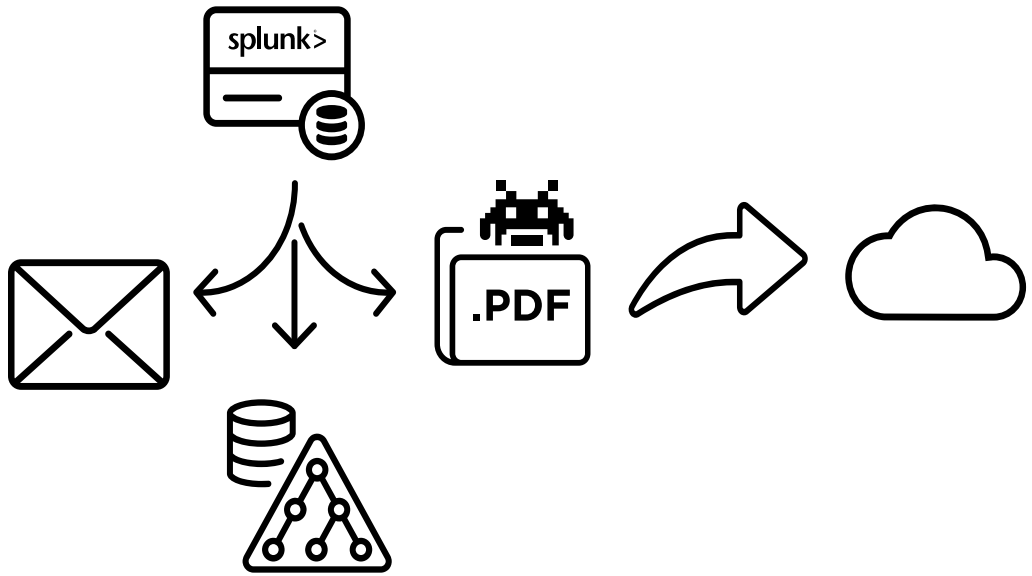
- Make ES your single pane of glass

Risk

- Get more value out of your detection content

Adaptive Response

- Creating / updating tickets
- Pulling in extra data
- External API actions



#6: Data Hygiene

Catching Things That Get Missed

I onboarded all my data into Splunk, now what?

Data model acceleration failing

- Search heads overtaxed

Data feeds stopping

- Network firewall changes

Field extractions or values changing

- Product version upgrade leads to change in log format

#6: Data Hygiene

Catching Things That Get Missed

At small scale you can make an alert for each sourcetype

- Use fieldsummary command to look for fields not being extracted
- Write a regex to match expected values of critical fields

Large scale environments require automated auditing of correlation searches

- Manual or automated Red Team testing
  - Free tools: Atomic Red Team, Endgame Red Team Automation, etc.
- Automate the whole process with tools such as: Verodin, SafeBreach, AttackIQ

Key Takeaways

1. Dedicate adequate attention towards maintenance cycles
2. Leverage macros & frameworks
3. Verify your timestamps & data feeds are working as expected

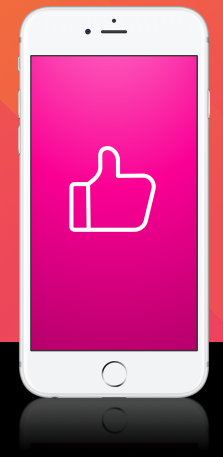
.conf19

splunk>

Thank You!

Go to the .conf19 mobile app to

RATE THIS SESSION



Q&A

Chris Ogden
Principal Threat Detection Engineer | Sony

Drew Guarino
Senior Threat Detection Engineer | Sony

splunk> .conf19