Dynamic Detection: May the ML be with you

Ali ÇETİN (YapıKredi Bank, Turkey)

Semih GELİŞLİ (YapıKredi Bank, Turkey)

About Yapı Kredi Bank

We are the 3rd largest private bank in Turkey with total assets worth $ 3.04 billion as of the end of 2021

805
Branches

4,590
ATM's

~15.5k
Employees

12.9m
Credit Cards

9.1m
Digital Customers

84%
Digital Banking Active Customer Penetration

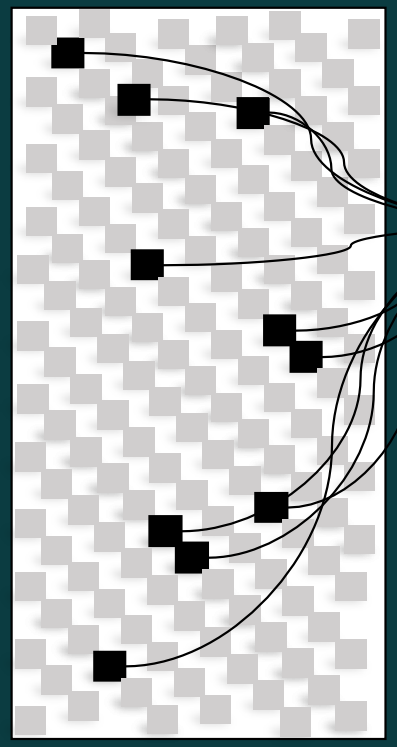
Yapı Kredi's SOC Infrastructure

Our philosophy on Security Data Lake and Artificial Intelligence is to place technology enabling human to make tasks more effective and efficient with output-driven methodology

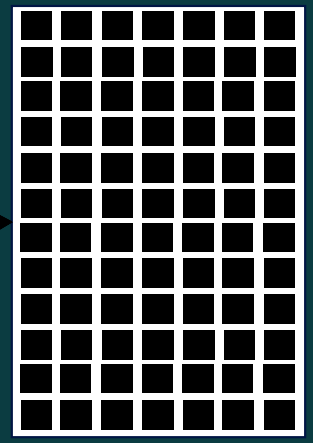
100+
Data Sources

50+
Data Collector

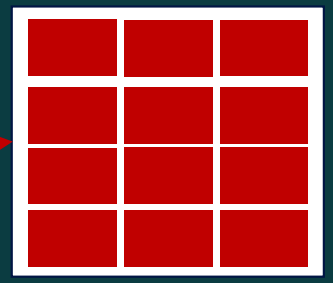
~%45
Successfully Filtering out



~60 Billion / Daily
Security Logs



9.000+
Investigative leads have been analyzed



1.000+
Anomaly events

%95+
SLA Success

100+
Monthly Alert Per Analyst

8 min
Mean time to Detect

*2021 Yearly Statistics

Agenda

What's lurking in your environment?

- What are we looking for and how can we spot abnormal behaviors within our environment?

Non-ML gurus are digging ML

- The methodology of our use-cases and how did we kick off?

Wrestling with Likelihood approaches and ML

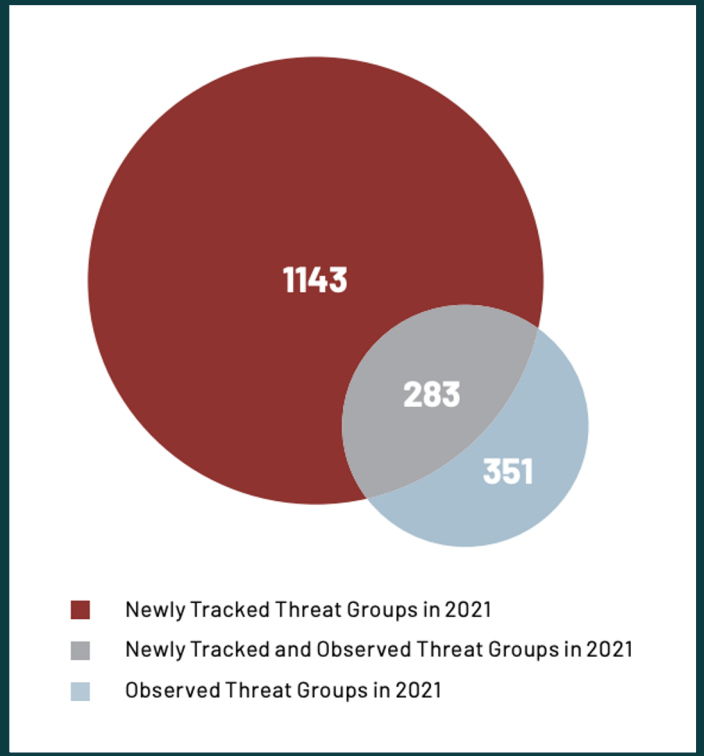
- Statistics and ML for Security related data
- Process Anomaly Detection
- Service User Account Behaviors
- CMDLine Obfuscation Detection

Wrap Up

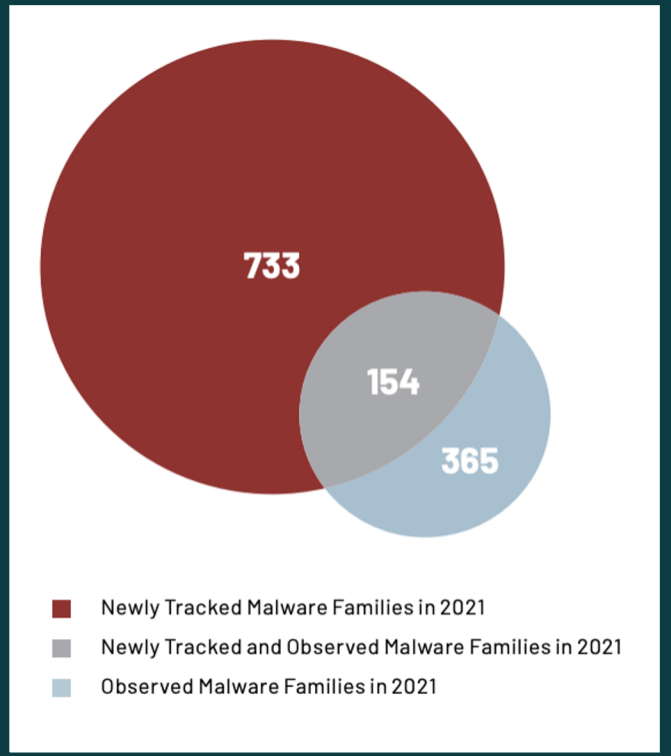
What’s lurking in your environment?

Mandiant M-Trend 2022 Report shows us day by day new APT groups and Malwares have been increasing significantly

APT Groups



Malware

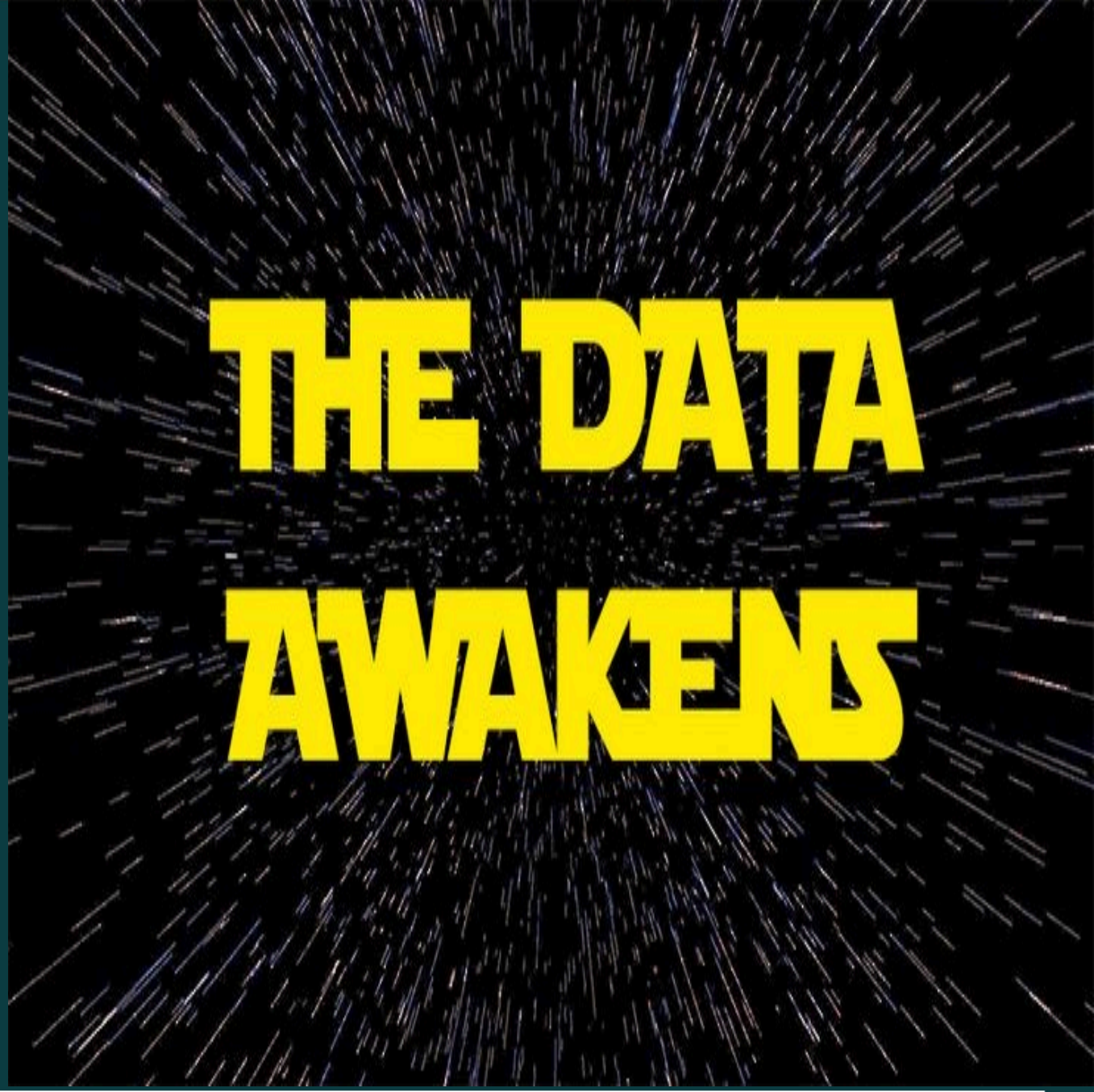


‘A long time ago in a **data** galaxy far, far away…’ – Opening Credits

Data Awaken with Right Data & Quality

Unlock the hidden value of your data

“Data is food for AI, and modern AI systems need not only calories, but also high quality nutrition.” Andrew NG

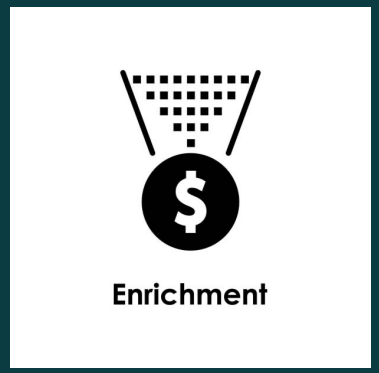


Our Strategy for “Detecting Unknown Unknowns”

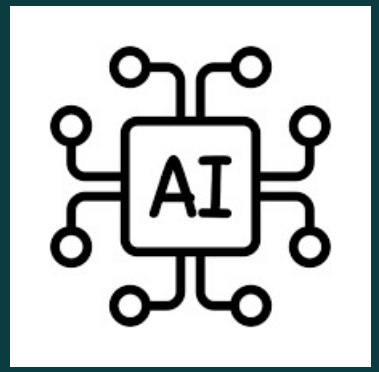
Managing the Generated Log on the Data Sources

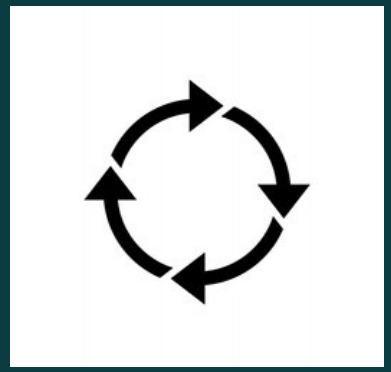
Filtering Out

Data Enrichment



Training Data





Lifecycle of Use case

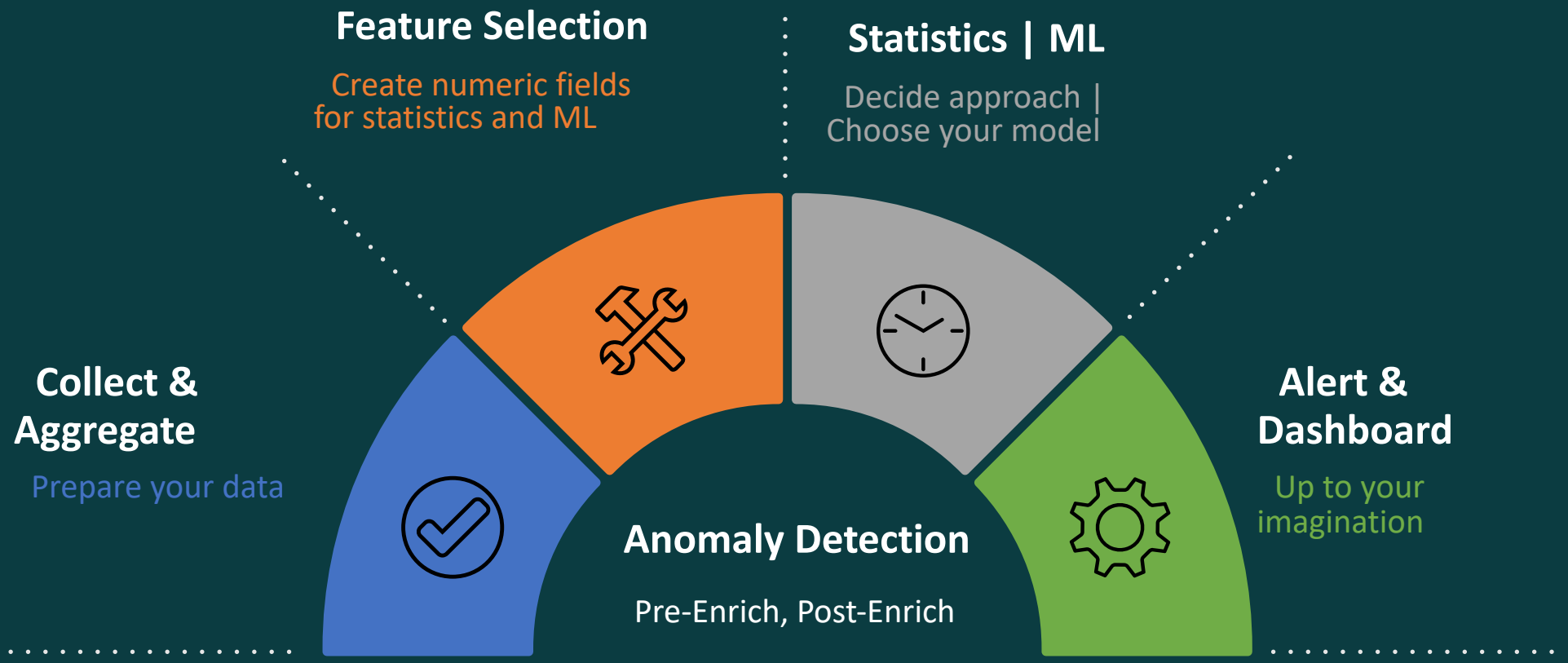
Threat Hunting

Three Steps of Methodology

- The Trigger
- Investigation & Enrichment
- Resolution

Non-ML Gurus are digging Statistics & ML

Statistics & Machine Learning Steps



Use Case 1: Process Anomaly Detection

Analogy

Q: How can we define the model to detect anomalous process creations and first time seen process tree per host & globally?

Likelihood Approach Methodology

- aggregate and count up number of times that process trees seen
- create features; total count global and local, global and local ratio percent, threshold
- likelihood approach
- run every night e.g. over 30 days
- collect everything into summary index
- create post enrichment steps; clustering, process eps, path eps, Echotrail etc.

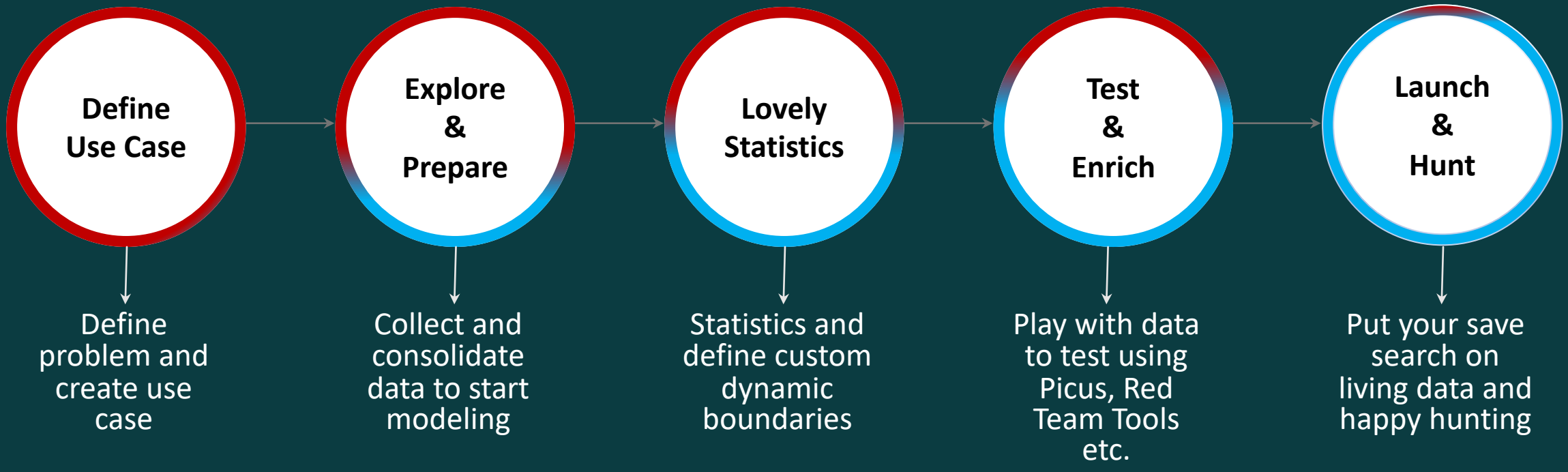
Results

- outliers of same hash running with multiple process tree
- outliers of same process running with multiple hashes
- first time appearance of process tree

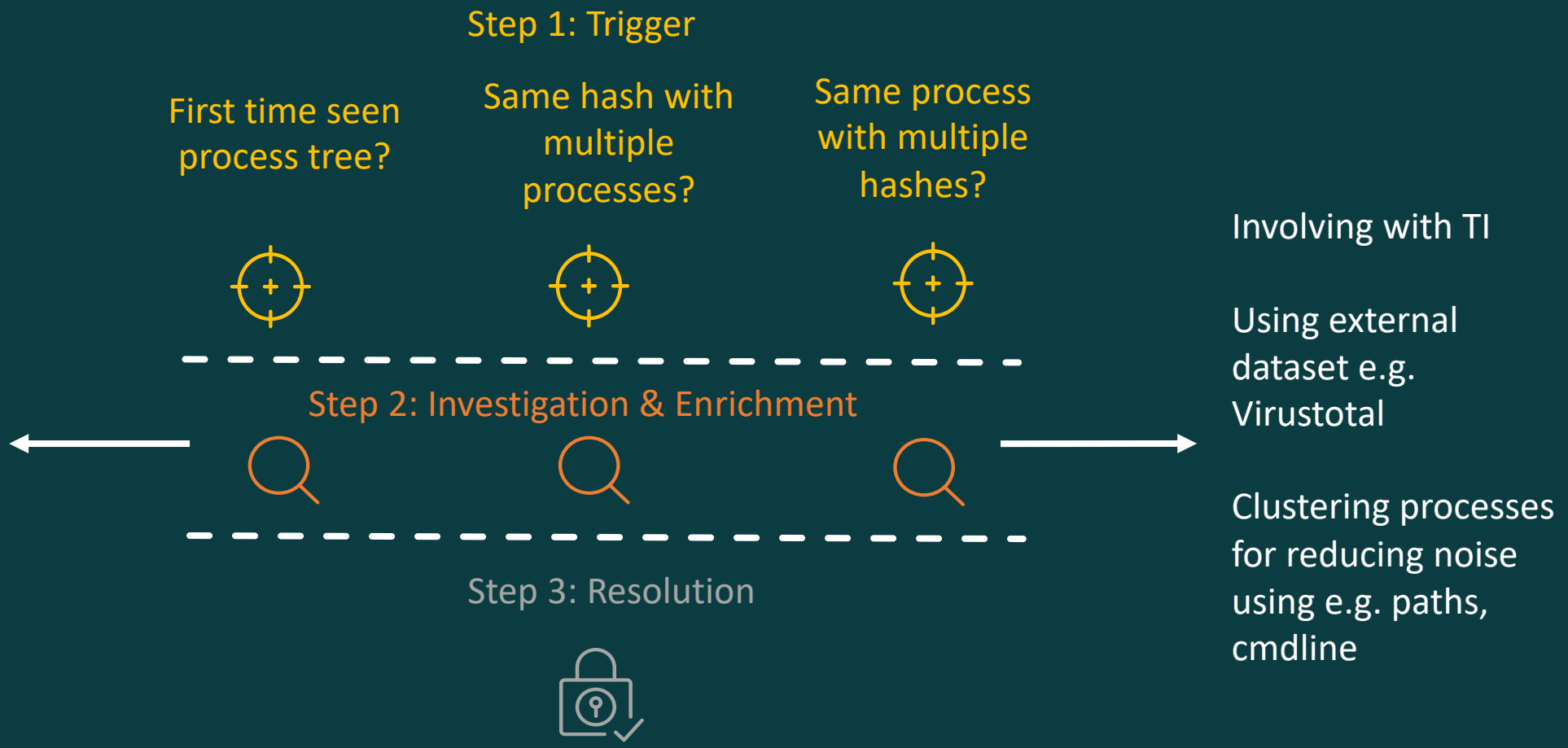
Challenges

- processes with /$TEMP$/ and /$USER$/ directories are making noise
- legit processes can have multiple different hashes and can be running under multiple paths and may still come as noise

Likelihood Approach Steps



Detection Diagram



Statistics & Likelihood Details

```
4  | stats sum(count) as total_hour max(_time) as LastTime by dest
     full_process _time,process_hash
5  | stats sum(total_hour) as total max(_time) as LastTime by full_process
     process_hash dest
6  | eventstats sum(total) as total_by_dest by dest
7  | eventstats sum(total) as total_global
8  | eventstats sum(total) as total_global_process by full_process
9  | eventstats sum(total) as total_global_hash by process_hash
10 | eval ratio_perc_local=(total/total_by_dest)*100
11 | eval ratio_perc_global_process=(total_global_process/total_global)*100
12 | eval threshold=ratio_perc_global_process/4
13 | eval ratio_perc_global_hash=(total_global_hash/total_global)*100
14 | eval recent=relative_time(now(),"-1d@d")
15 | sort +ratio_perc_global_hash
16 | where (ratio_perc_global_hash < threshold  AND LastTime > recent) OR
     (ratio_perc_global_process < ratio_perc_global_hash  AND LastTime >
     recent)
17 | eval signature=if((ratio_perc_global_hash < threshold  AND LastTime >
     recent),"same process with multiple hashes","same hash with different
     processtree")
18 | eval _time=LastTime
19 | table full_process process_hash dest total signature _time
20 | collect index=whatever sourcetype=whatever addtime=true marker
     ="search_name=\"it's not magic, yet here we are\""
```

Creating process tree and aggregate data with | stats

Calculating base feature (total count locally and globally)

Unleash total ratio locally and globally by hash values

Adding likelihood approaches!

Clean table & write outputs

Enrichment Details

```
| eval comb=path_len*dir_count
| eventstats stdev(path_len) as stdev_path_len stdev(parent_len) as
    stdev_parent_len stdev(ut_shannon) as stdev_shannon avg(path_len) as
    avg_path_len avg(parent_len) as avg_parent_len stdev(comb) as stdcomb a
    (comb) as avgcomb by process_name
```

Creating features for cluster

```
|  eval upperBound=avgcomb+stdcomb,lowerBound=avgcomb-stdcomb,upperBoundv2
    =avgcomb+stdcomb*2,lowerBoundv2=avgcomb-stdcomb*2
|  eval cluster=case(comb <= upperBound AND comb >= lowerBound,"1",comb >
    upperBound AND comb < upperBoundv2,"2", comb < lowerBound AND comb >
    lowerBoundv2,"3",(comb > upperBoundv2 OR comb < lowerBoundv2),"-1")
```

Creating custom cluster

```
|  cluster t=0.8 field=process_path_value showcount=true labelonly=t
| lookup process_tracker full_process OUTPUTNEW dest as previous_dest
|  stats values(detected_dest) as detected_dest  dc(previous_dest) as
    count_previous_dest dc(detected_dest) as count_detected_dest values
    (full_process) as full_process by process_name dir_count cluster
    cluster_label cluster_count
```

Double checking with | cluster

```
|  eval first_seen=if(count_previous_dest="0" AND count_detected_dest="1"
    ,"First Seen Globaly","Seen Previously")
|  search first_seen="First Seen Globaly" AND cluster="*"
```

Looking for the first time seen globally

Drilldown Investigation

Enriching output with external data sources

```
| stats count by process_name
| rename process_name as process_info
| lookup echotrail process_info OUTPUT process_eps
| search NOT process_eps="*"
| head 250
| echotrail echotrail_hash="hash" echotrail_eps="process_eps" echotrail_pat
    ="paths"
```

process_info	process_path_value	check_path	paths	process_eps
taskhostw.exe	C:\\Windows\\System32	1	["C:\\Windows\\System32","99.89"] ["C:\\WINDOWS\\System32","0.11"]	86.41
explorer.exe	C:\\Windows	1	["C:\\Windows","67.50"] ["C:\\Windows\\SysWOW64","32.06"]	94.72

Server First Seen

process_id	parent_process_name	process_name
7748	456406.exe	net.exe
9060	456406.exe	net.exe
11716	456406.exe	powershell.exe
2128	456406.exe	powershell.exe

Detailed Process Tree

tree

```
picus.agent.service.exe (11676)
|--- 456406.exe (7104)                    2022-04-25 03:20:04 "C:\ProgramData\Picus Security\Picus Agent\Scenarios\57300-456406\4
`` |--- powershell.exe (11716)             2022-04-25 03:20:10 "powershell.exe" -command "if(Test-Path -Path 'HKLM:SOFTWARE\Classe
[0].Replace('WinRAR.exe', 'Rar.exe') 'a -r %%public%%\Downloads\328054.rar %%public%%\Downloads\test\mimi.log'}else{Write-Host 'YOK'}"
`` |--- net.exe (9060)                    2022-04-25 03:20:09 "net" localgroup administrators guest /delete
`` `` |--- net1.exe (5728)                2022-04-25 03:20:10 C:\Windows\system32\net1 localgroup administrators guest /delete
`` |--- net.exe (7748)                     2022-04-25 03:20:09 "net" localgroup administrators guest /add
`` `` |--- net1.exe (11588)               2022-04-25 03:20:09 C:\Windows\system32\net1 localgroup administrators guest /add
`` |--- taskkill.exe (5712)                2022-04-25 03:20:08 "taskkill.exe" /f /im hh.exe
`` |--- cmd.exe (1380)                     2022-04-25 03:20:08 "cmd.exe" /c del C:\Windows\TEMP\CradleTest.txt
`` |--- hh.exe (10368)                     2022-04-25 03:20:08 "hh.exe" http://10.60.68.11:80/529363\CradleTest.txt
`` |--- cmd.exe (11980)                    2022-04-25 03:20:07 "cmd.exe" /c del "C:\wmi.dll"
`` |--- cmd.exe (2276)                     2022-04-25 03:20:07 "cmd.exe" /c whoami &gt; C:\wmi.dll 2&gt;&amp;1
`` `` |--- whoami.exe (10988)             2022-04-25 03:20:07 whoami
`` |--- cmd.exe (11696)                    2022-04-25 03:20:07 "cmd.exe" /c echo \"1q2w3e4r\" &gt;&gt; pass.txt
`` |--- cmd.exe (5164)                     2022-04-25 03:20:06 "cmd.exe" /c echo \"Password1\" &gt;&gt; pass.txt
`` |--- cmd.exe (9348)                     2022-04-25 03:20:06 "cmd.exe" /c echo \"1q2w3e4r\" &gt;&gt; pass.txt
`` |--- cmd.exe (9300)                     2022-04-25 03:20:06 "cmd.exe" /c echo \"Password1\" &gt;&gt; pass.txt
`` |--- cmd.exe (9368)                     2022-04-25 03:20:06 "cmd.exe" /c del dirlog.txt
`` |--- cmd.exe (6840)                     2022-04-25 03:20:05 "cmd.exe" /c dir &gt; dirlog.txt
```

Use Case 2: Service Account Anomaly Detection

Analogy

Q: How can we define the model to profile each service account login behaviors and detect outliers?

ML Approach Methodology

- aggregate service accounts by LogonType, dest and src
- create features; distinct count of dest, all unique values of dest, distinct count of source, all unique values of source, LogonType
- DensityFunction model on each feature by service account user
- apply your model each day to test last day's data
- collect into index
- generate risk score for priority
- run every night and happy hunting

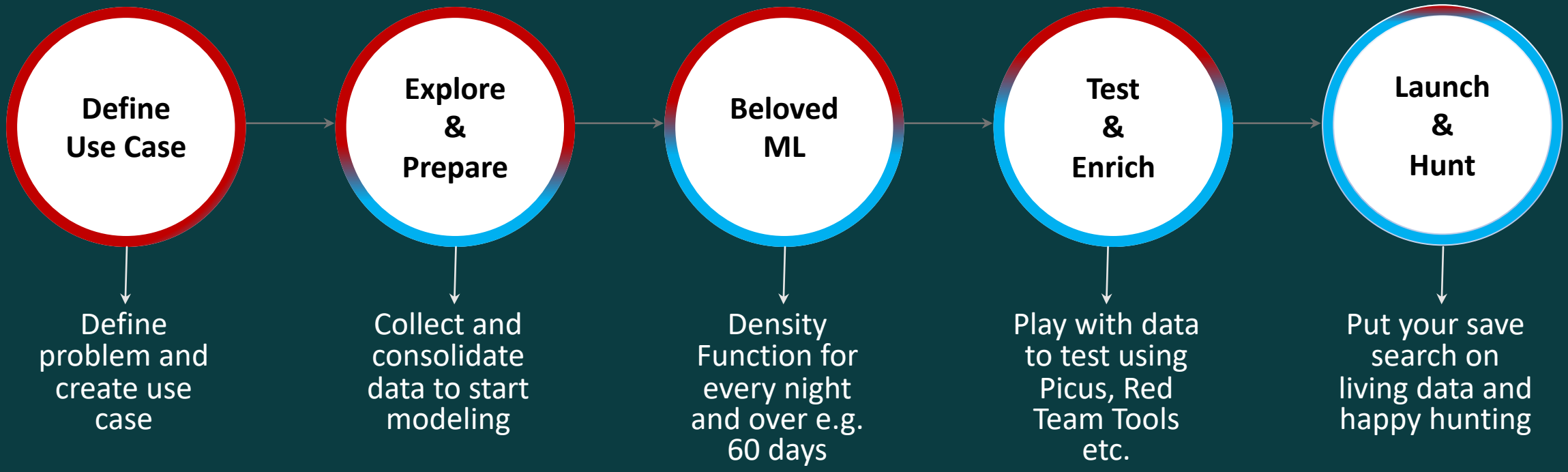
Result

- spots three type of outliers: LogonType anomaly, Destination anomaly and Source Anomaly
- gives you another perspective to look for e.g. Lateral Movement, Credential Theft

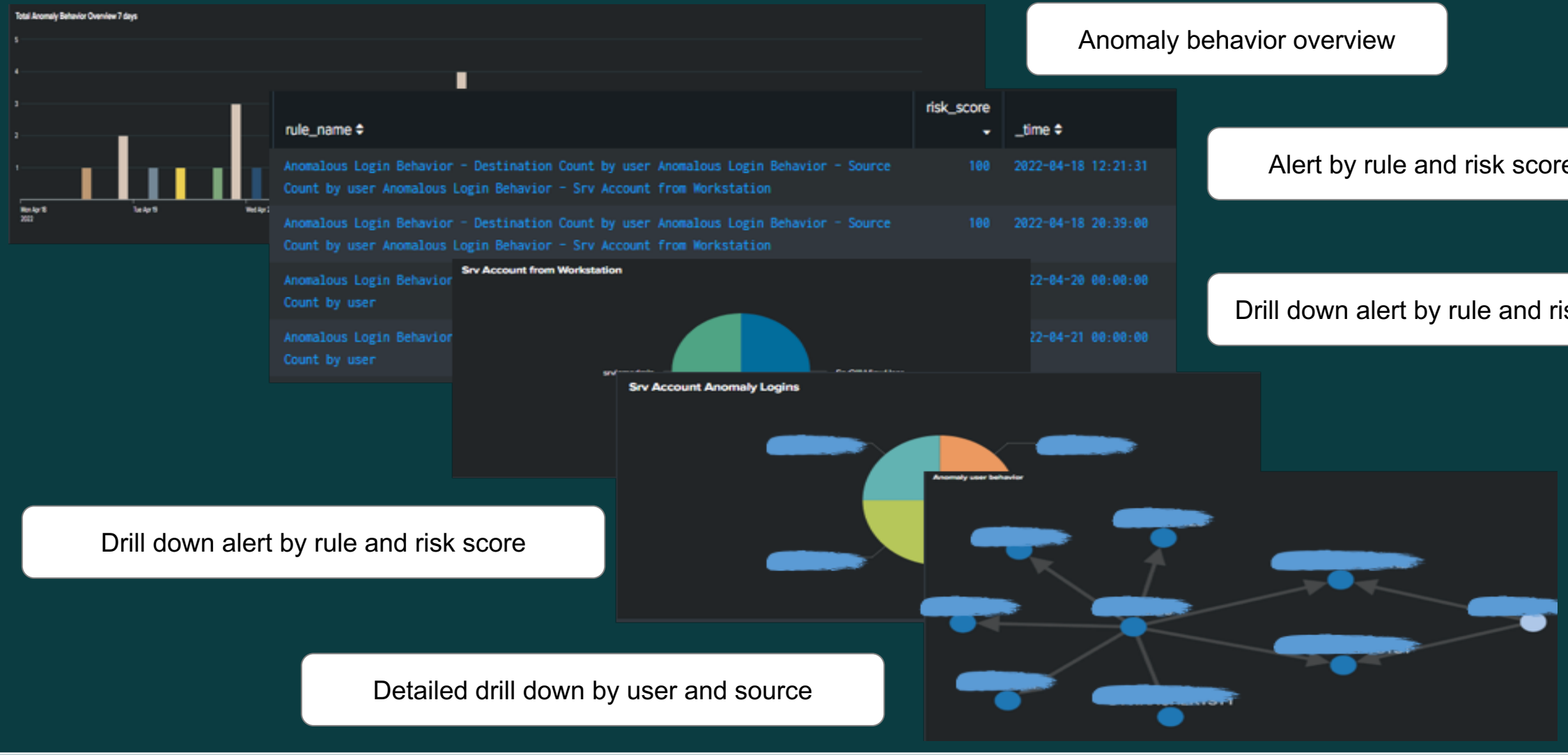
Challenges

- newly added users or servers login sessions can make noise: Enrich data with Active Directory data

Likelihood Approach Steps



Happy Hunting



Anomaly behavior overview

Alert by rule and risk score

Drill down alert by rule and risk score

Drill down alert by rule and risk score

Detailed drill down by user and source

Use Case 3: Obfuscated CMDLine Detection

Analogy

Q: How can we define the model to profile each service account login behaviors and detect outliers?

ML Approach Methodology

- create labeled dataset
- aggregate cmdline with users, host
- create features: proportion of special char, proportion of escape sequences, and TFIDF
- reduce dimension with PCA
- Classification model on each feature by cmdline
- choose best fitted model
- apply your model on living data
- collect into index and add sigma rules

Result

- spots obfuscated CMDLine
- gives another perspective to look for e.g. Execution, Defense Evasion

Challenges

- brand-new obfuscation methods may not be detected
- management of whitelist & blacklist to update model periodically

Feature selection & ML Details

```
| stats count by cmdline
| eval cmdline_len = max(1,len(cmdline))
| rex field=cmdline max_match=0 "(?<spaces>\s)"
| rex field=cmdline max_match=0 "(?<special>[^\w|\s\\\:\.\"]+)"
| rex field=cmdline max_match=0 "(?<upper>[A-Z-\s]+)"
| eval space_ratio=if(isnull(spaces),0.0,mvcount(spaces) / cmdline_len)
| eval special_ratio=if(isnull(special),0.0,mvcount(special) / cmdline_len)
| eval upper_ratio=if(isnull(upper),0.0,mvcount(upper))
```

creating features for classification, more features=higher accuracy

```
| `ut_shannon(cmdline)`
| eval num_obfuscation =
    (mvcount(split(cmdline,"`"))-1)
  + (mvcount(split(cmdline, "^"))-1)
  + (mvcount(split(cmdline, "'"))-1)
  + (mvcount(split(cmdline, "%"))-1)
  + (mvcount(split(cmdline, "*"))-1)
  + (mvcount(split(cmdline, "+"))-1)
  + (mvcount(split(cmdline, "="))-1)
  + (mvcount(split(cmdline, ">"))-1)
```

looking for specific characters that mostly using for obfuscation

- cleaning data
- features
- TFIDF
- PCA
- classification

obfuscated	obfuscated	powershell	""'\""dir c:\windows\system32\calc.exe\""|&(\""iex\"")'|&(\""IEX\""); Write-Host \""<>^^|\&^\"
obfuscated	obfuscated	powershell	""SEt O07sTn ([cHaR[&}3'+'{'+')}2{}0{}2'+'{'+',)'+'& epy}2{+}'+'2'+'{t'+'}2{+'+'}2{ ; '+'DERIAPN}2{+'+'}2{U}2{+}2{'+' ;txt.a'+'l}2'+'{+}2{b led}2{('(
obfuscated	obfuscated	powershell	""Write-Host 'this i
obfuscated	obfuscated	powershell	""Write-Host \""this is a test 1\"
obfuscated	obfuscated	powershell	""dir \""c:\windows\system32\ca*c.exe\"""

class	predicted	orig_text
obfuscated	obfuscated	CmD , , ,/R "", (^set G^T=ch)& (^sEt i^J=^e)& (s^Et ^Ma=o %%^te^mp%%)&& , cAll, S^ET C23z=%%i^J%%%%G^T%%%%^Ma%%%
obfuscated	obfuscated	CmD , ,,,,,,,,/V:ON,,,,,,,,,/C"",,,,,,,,(set ` =h)&&(,(,(set } =wr),),)&&(,(,(set ? =c),),)&&(,(,(set , =P%%"""

Wrap Up

“You will find only what you bring in to your data.” - Yoda

- Likelihood and Statistics are still one of the best weapon in arsenal
- risk score can reduce FP
- right data with high quality model allows you to look at hidden
- features and enrichments are the key
- combine threat hunting steps with Math and signature based rules
- we don’t know if we never try!
- research, engage and build!

Speakers

Semih GELİŞLİ

- Head of Cyber Security
- 12+ years of experience in Cyber Security and Information Security
- in/sgelisli/ – LinkedIn
- @sgelisli – Twitter

Ali ÇETİN

- Senior Cyber Security Engineer
- 4+ years of experience in Cyber Security and Network Security
- in/muhammedalicetin – LinkedIn

Thank you

Go raibh maith agat

Teşekkürler