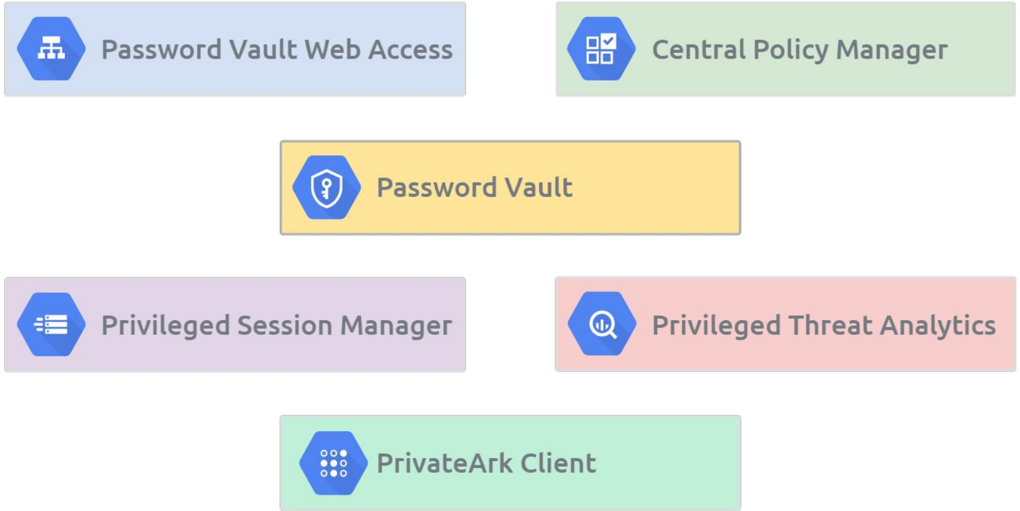
Malicious behavior detection based on CyberArk PAS logs through string matching and genetic neural networks

Presenters: **Ivar Slotboom and Mike Slotboom**, SNE/UvA
Supervisors: **Roel Bierens and Bartosz Czaszynski**, Deloitte

What is CyberArk Privileged Access Security (PAS)?

CyberArk PAS offers:

- Privileged access to hosts via managed sessions
- Password management based on policies



Password Vault Web Access

Central Policy Manager

Password Vault

Privileged Session Manager

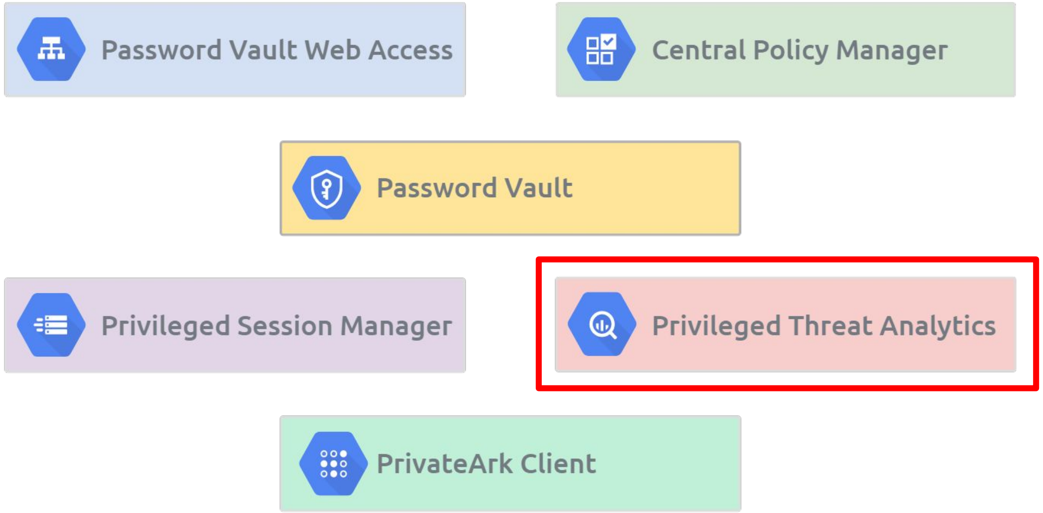
Privileged Threat Analytics

PrivateArk Client

What is the issue?

CyberArk PTA does not have a holistic view of misuse within the entire solution

1. PTA looks at user session only
2. Samples logs to handle load
3. Based on hardcoded triggers
4. Minimal data in output logs


Password Vault Web Access

Central Policy Manager

Password Vault

Privileged Session Manager

Privileged Threat Analytics

PrivateArk Client

Research question

How can one recognize malicious behavior based on the logs from CyberArk PAS in both the present and future?

Sub 1) **Which use cases can be defined for Privileged Access Management to distinguish malicious behavior?**

Sub 2) **How can future incidents be detected by using previously researched behavior from the CyberArk PAS logs?**

Methodology

17 Attack techniques selected from MITRE ATT&CK Enterprise Matrix in privileged sessions (Windows and Linux)

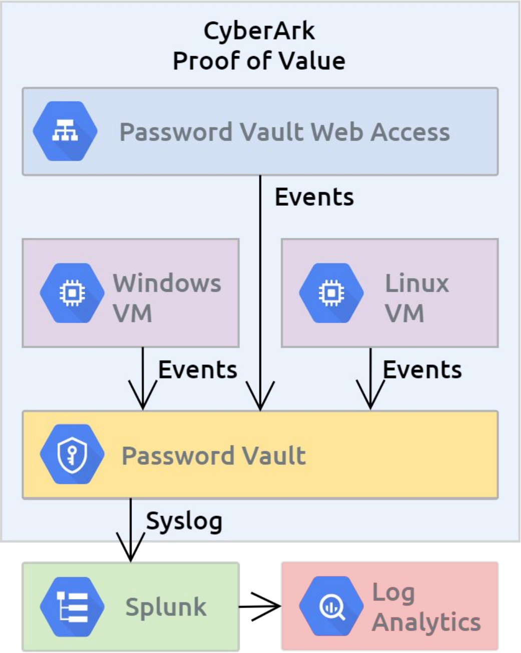
9 Additional techniques defined on CyberArk PAS system (PVWA and Password Vault)

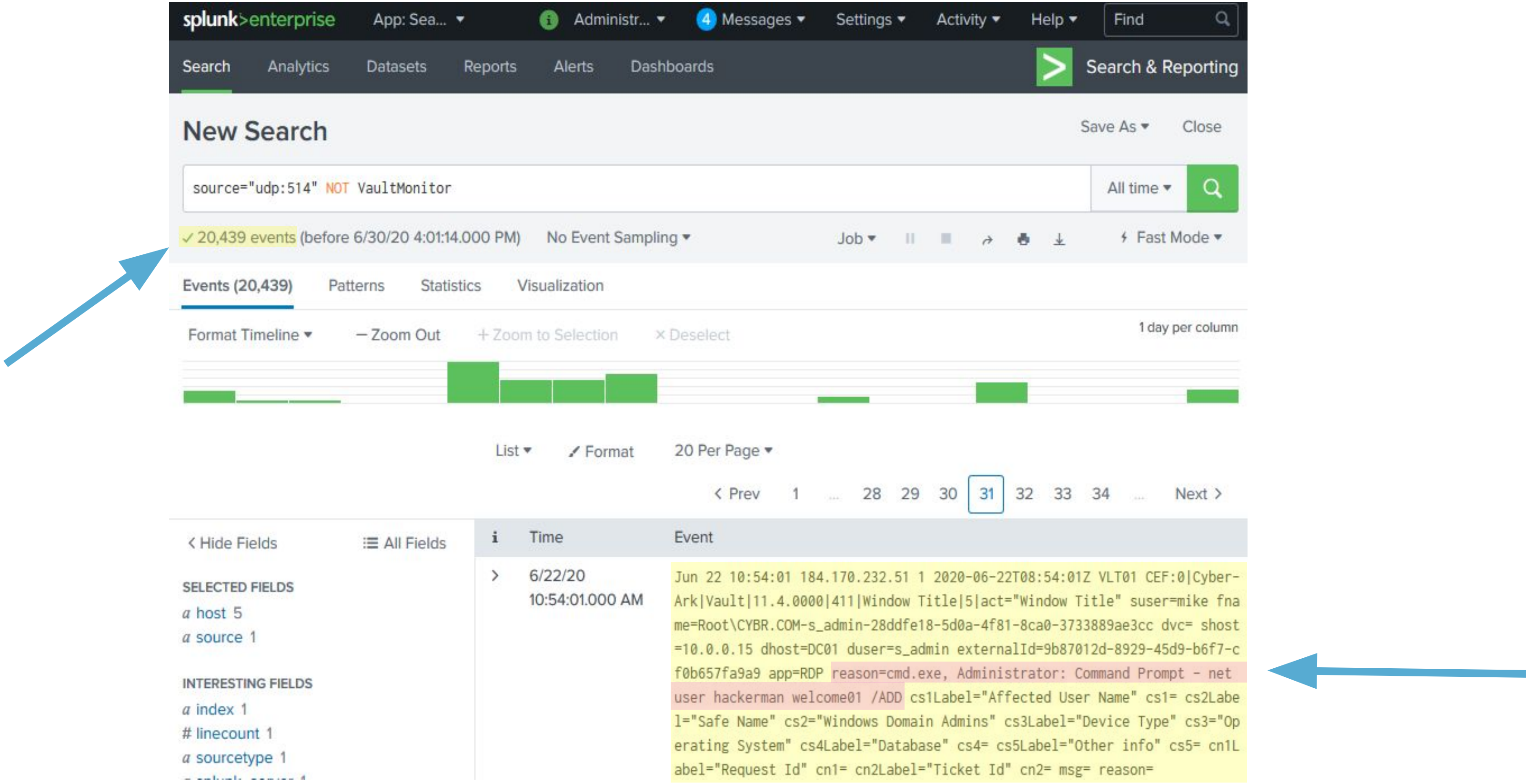
Run attack techniques and normal behavior simulation in test environment (CyberArk PoV) and capture logs

Split logs into normal, suspicious and malicious data sets

Define use cases (i.e. search queries) based on malicious logs

Apply automation in log analytics





```
source="udp:514" NOT VaultMonitor
```

✓ 20,439 events (before 6/30/20 4:01:14.000 PM)

6/22/20 10:54:01.000 AM

```
Jun 22 10:54:01 184.170.232.51 1 2020-06-22T08:54:01Z VLT01 CEF:0|Cyber-
Ark|Vault|11.4.0000|411|Window Title|5|act="Window Title" suser=mike fna
me=Root\CYBR.COM-s_admin-28ddfe18-5d0a-4f81-8ca0-3733889ae3cc dvc= shost
=10.0.0.15 dhost=DC01 duser=s_admin externalId=9b87012d-8929-45d9-b6f7-c
f0b657fa9a9 app=RDP reason=cmd.exe, Administrator: Command Prompt - net
user hackerman welcome01 /ADD cs1Label="Affected User Name" cs1= cs2Labe
l="Safe Name" cs2="Windows Domain Admins" cs3Label="Device Type" cs3="Op
erating System" cs4Label="Database" cs4= cs5Label="Other info" cs5= cn1L
abel="Request Id" cn1= cn2Label="Ticket Id" cn2= msg= reason=
```

A single log entry - sanitized

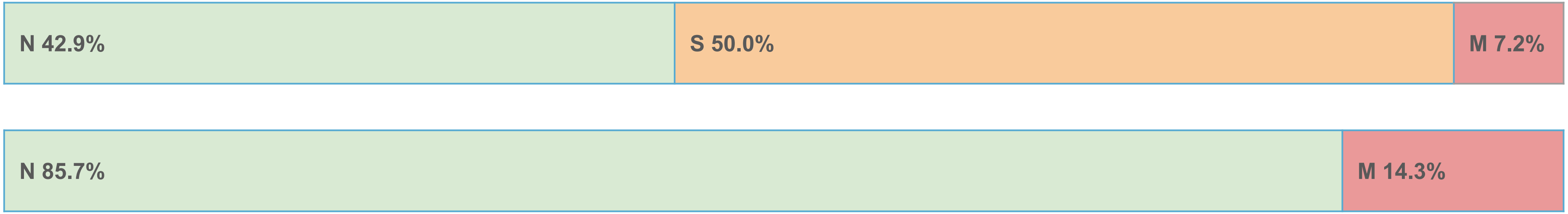
```
Jun  8 10:43:17 184.170.232.50 1 2020-06-08T08:43:17Z VLT01 CEF:0|Cyber-Ark|Vault|11.4.0000|361|Keystroke
logging|5|act="Keystroke logging" suser=Administrator fname=Root\Operating System-UnixSSH-rhel7.cybr.com-root
dvc= shost=10.0.0.15 dhost=rhel7.cybr.com duser=root externalId=8308babe-f4e8-445c-a1a8-4be6c96a61d0
app=SSH reason=sudo EDITOR\/=/usr/bin/nano visudo cs1Label="Affected User Name" cs1= cs2Label="Safe Name"
cs2="Linux Root" cs3Label="Device Type" cs3="Operating System" cs4Label="Database" cs4= cs5Label="Other info"
cs5= cn1Label="Request Id" cn1= cn2Label="Ticket Id" cn2= msg=
```

```
[["month", "Jun"], ["day", "8"], ["time", "10:43:17"], ["ip", "184.170.232.50"], ["unknown", "1"], ["timestamp",
"2020-06-08T08:43:17Z"], ["hostname", "VLT01"], ["format", "CEF:0"], ["platform", "Cyber-Ark"], ["application", "Vault"],
["application_version", "11.4.0000"], ["event_id", "361"], ["event_message", "Keystroke logging"], ["event_level", "5"], ["act",
"Keystroke logging"], ["suser", "Administrator"], ["fname", "Root\\Operating System-UnixSSH-rhel7.cybr.com-root"],
["dvc", ""], ["shost", "10.0.0.15"], ["dhost", "rhel7.cybr.com"], ["duser", "root"], ["externalId",
"8308babe-f4e8-445c-a1a8-4be6c96a61d0"], ["app", "SSH"], ["reason", "sudo EDITOR\\/=/usr/bin/nano visudo"],
["cs1Label", "Affected User Name"], ["cs1", ""], ["cs2Label", "Safe Name"], ["cs2", "Linux Root"], ["cs3Label", "Device Type"],
["cs3", "Operating System"], ["cs4Label", "Database"], ["cs4", ""], ["cs5Label", "Other info"], ["cs5", ""], ["cn1Label", "Request
Id"], ["cn1", ""], ["cn2Label", "Ticket Id"], ["cn2", ""], ["msg", ""]]
```

Unbalanced data set

Data set (i.e. 5300 log entries) consist of:

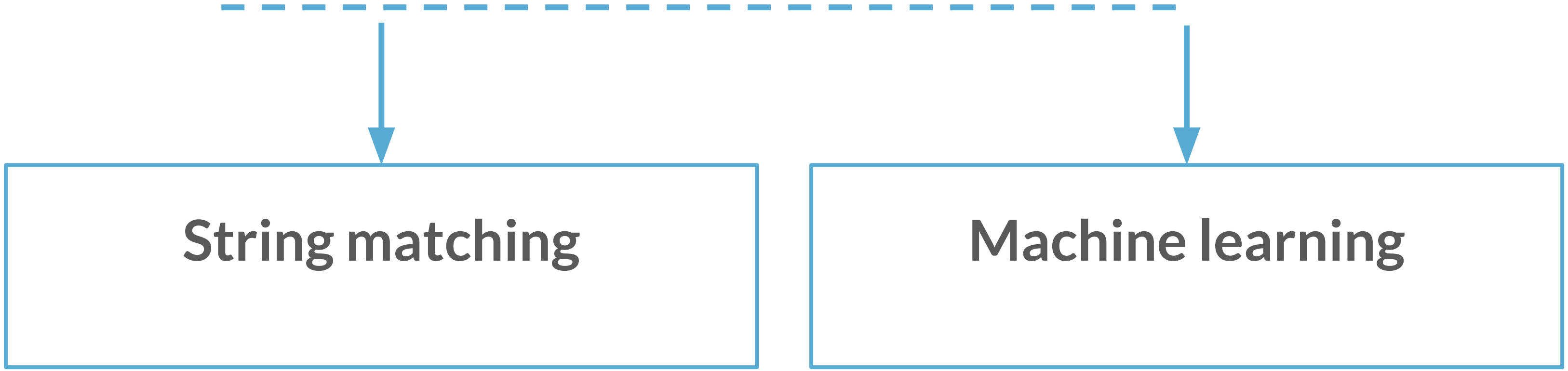
2272 Normal behavior logs ("N")
2648 Suspicious logs ("S")
380 Pure malicious logs ("M")



Hard to classify log as malicious

(We explored)

Two methods to analyze log entries



String matching

Machine learning

String matching

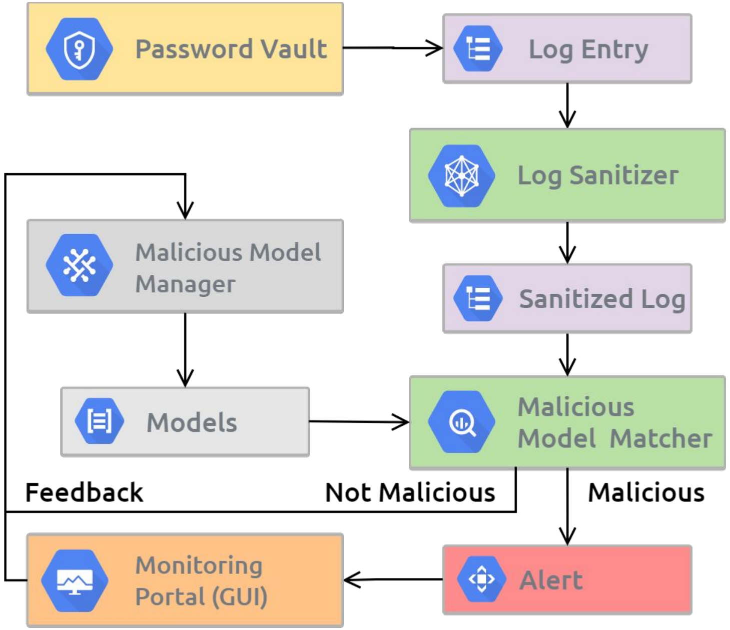
Incoming log entries sanitized and matched with predefined models (i.e. use cases)

Alert raised in portal when log entry is found to be malicious

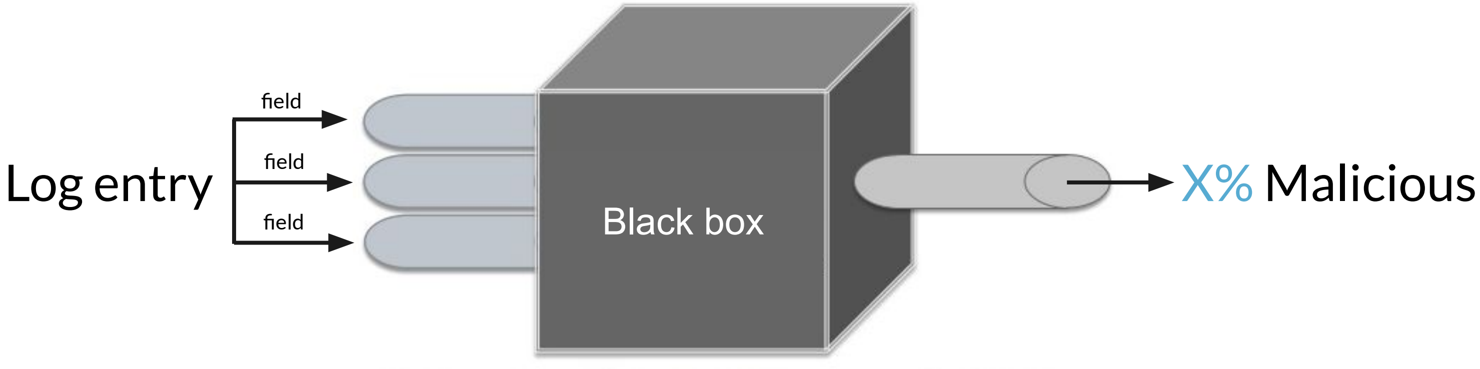
Optional feedback loop to expand models

Portal and Matcher are universal (e.g. Splunk)

Drawbacks: Known models & Human factor



Machine learning

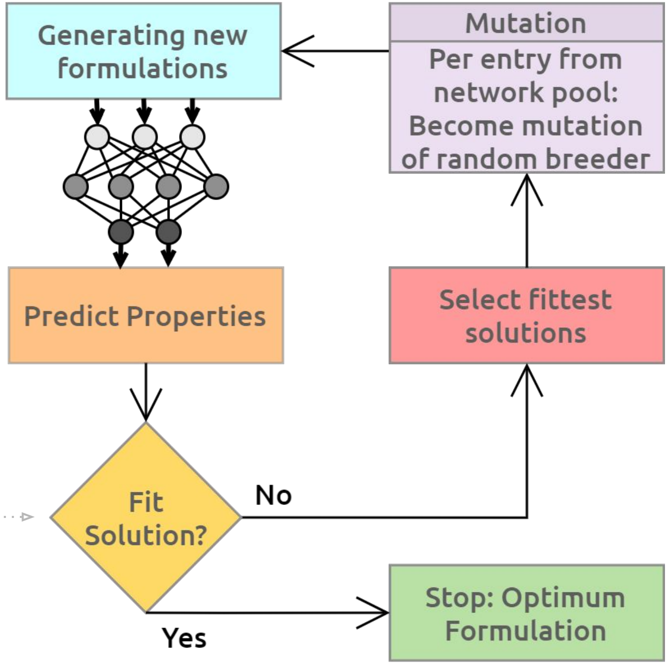


Training process where one teaches a model where no fully satisfactory algorithm is available.

Training a neural network genetically

Pool of 512 networks

E.g. 99% accuracy


Generating new formulations
Predict Properties
Fit Solution?
No
Yes
Select fittest solutions
Mutation
Per entry from network pool: Become mutation of random breeder
Stop: Optimum Formulation

1 to 10 weights get reset for each network

Top 16 of 512 (≈3%) become breeders

Converting a log to neural network inputs

Bag of ~~words~~ phrases, based on frequency.

```
[["month", "Jun"], ["day", "8"], ["time", "10:43:17"], ["ip", "184.170.232.50"], ["unknown", "1"], ["timestamp",
"2020-06-08T08:43:17Z"], ["hostname", "VLT01"], ["format", "CEF:0"], ...]
```

Month	Day	Time	IP	Unk.	Time-stamp	Hostname	Format
104x “Jun”	**72x** “8”	**2x** “10:43:17”	**834x** “184.170.232.50”	**1435x** “1”	**1x** “2020-06-08T08:43:17Z”	**937x** “VLT01”	**1435x** “CEF:0”
23x “Jul”	**68x** “9”	**1x** “9:03:45”	**147x** “184.170.232.49”		**...**	**183x** “VLT02”	
	55x “10”	**1x** “9:03:46”				**...**	
	...	**...**					

Performance indicators

$$Trainingscore = \frac{F1score + Deltascore}{2}$$

F1 score

- Motivates classification correctness regardless of unequal ratios
- Measured using confusion matrix formulas

$$\text{Recall} = \frac{tp}{tp + fn} \qquad \text{Precision} = \frac{tp}{tp + fp}$$

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Delta score

- Motivates output to be precise
- Measured by the error delta
- Own invented solution

```
if $trainingEntry.desiredResult$ == $0.0$ then
    errorTotal += output;
else
    errorTotal += 1.0 - output;
```

$$Deltascore = 1 - \frac{TotalErrorDelta}{TotalErrorDeltaSamples}$$

Model types

Detector

Takes in **any** log, determines whether it's malicious or normal behavior

Single output:

Confidence of the log being malicious

Desired outcome: Either 1 TP or 1 TN

Classifier

Takes in **malicious** logs, determines the type of attack that was performed

Multiple outputs:

One output per attack based on the confidence that it was that attack

Desired outcome: 1 TP and 16 TNs

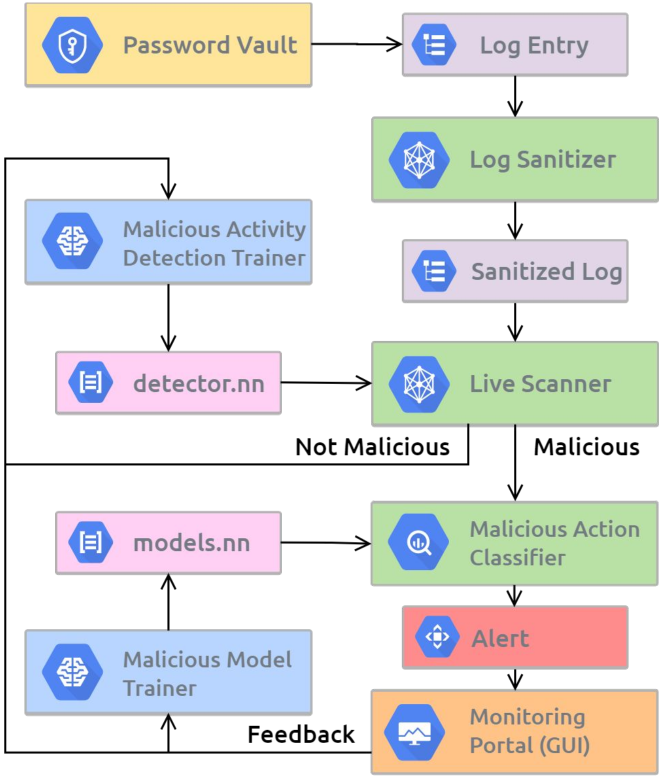
Machine learning framework

Same sanitizing approach as string matching

Machine learning applied in detector and classifier

Live scanner split from Detection Trainer to handle load

Feedback loop to adjust training sets for future incidents



Machine learning performance experiments

Reference setup: 4 hidden layers, 20 nodes per hidden layer, 0.5 classification threshold, “N” data set

Experiment	Title	Values
A	Using different **training sets** (detector only)	Normal Behavior (“N”) Normal Behavior + Suspicious (“N+S”)
B	Using a different number of **hidden layers**	1 2 4 8 12 (detector only) 16 (classifier only)
C	Using a different number of **nodes** per hidden layer	10 20 40
D	Using different classification **thresholds**	0.0, 0.1, …, 0.9, 1.0
E	Using **optimal parameters** from previous experiments to test performance	Depending on first four experiments

Results

Malicious behavior detection

Use cases

✓ 12 of 17 MITRE attack techniques and 6 out of 9 additional attack techniques successfully defined

⚠ 4 Attack techniques were indistinguishable

✗ Remaining 4 attack techniques were not visible in log, which were:

1. Phishing link
2. User circumventing PSM!
3. Capturing client session cookies
4. Deactivating security configuration rules (PTA)!

Machine learning

✓ Capable of handling **large amounts** of log entries

✓ Close to **99%** of the malicious logs can be filtered out successfully (from successful defines)

✓ Able to find anomalies in **any environment**, since no hard coding is required

✓ **Less dependent** on humans, causing less human error

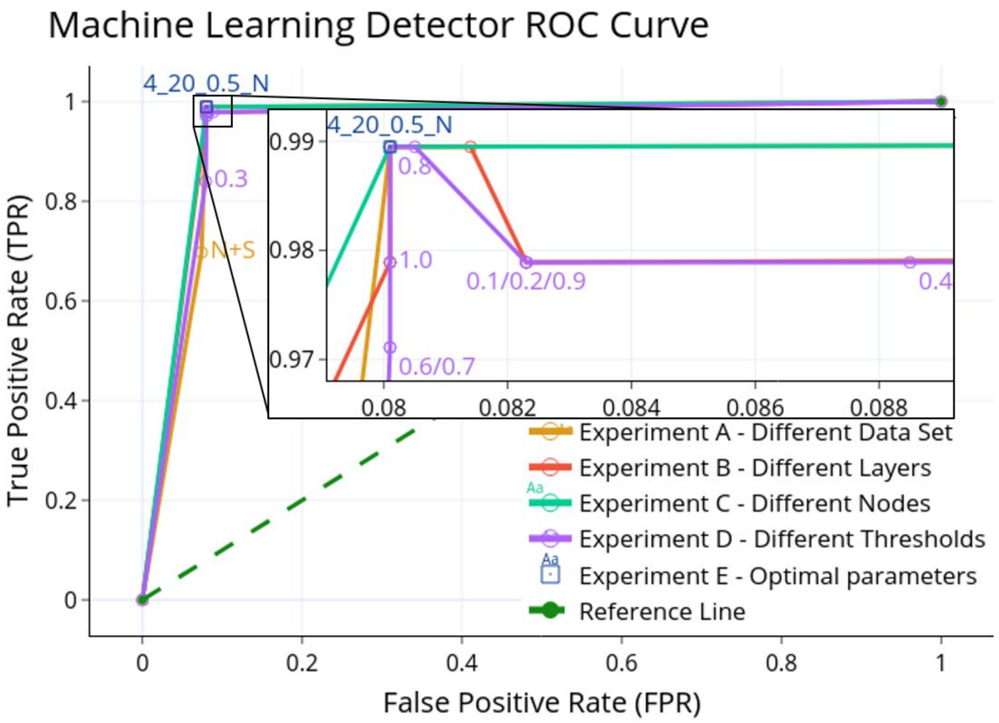
Detector Experiments

How well does it detect malicious logs?

Optimal parameters:

- 4 hidden layers
- 20 nodes per hidden layer
- 0.5 classification threshold
- “N” data set

TP	TN	FP	FN
376	2090	182	4



$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

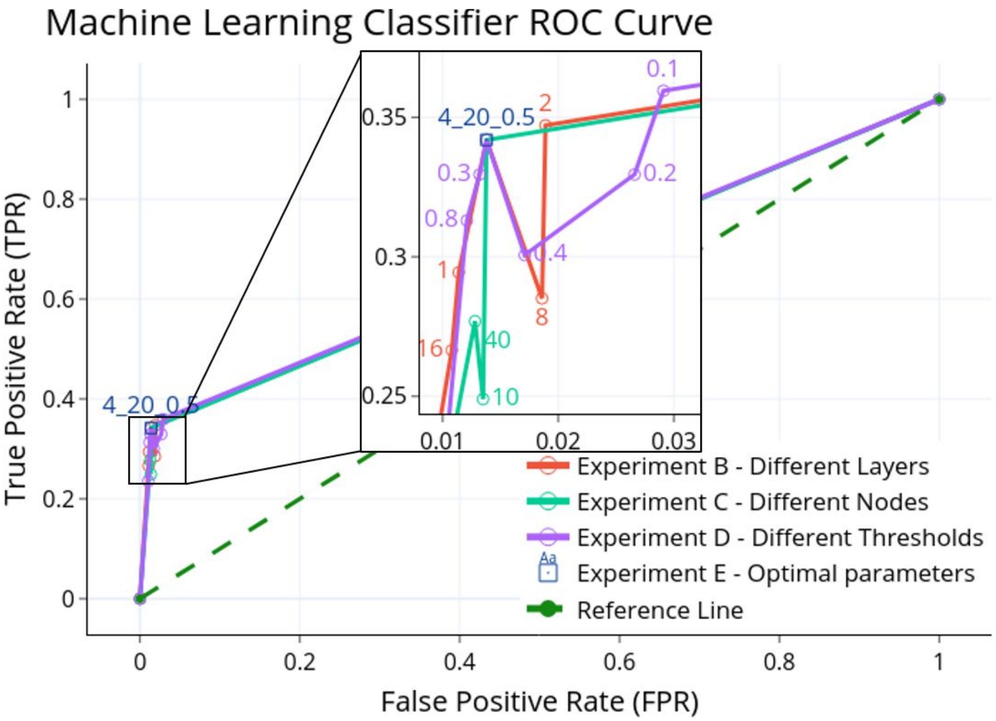
Classifier Experiments

How well does it match a malicious entry with a model?

Optimal parameters:

- 4 hidden layers
- 20 nodes per hidden layer
- 0.5 classification threshold

TP	TN	FP	FN
331	15275	213	637



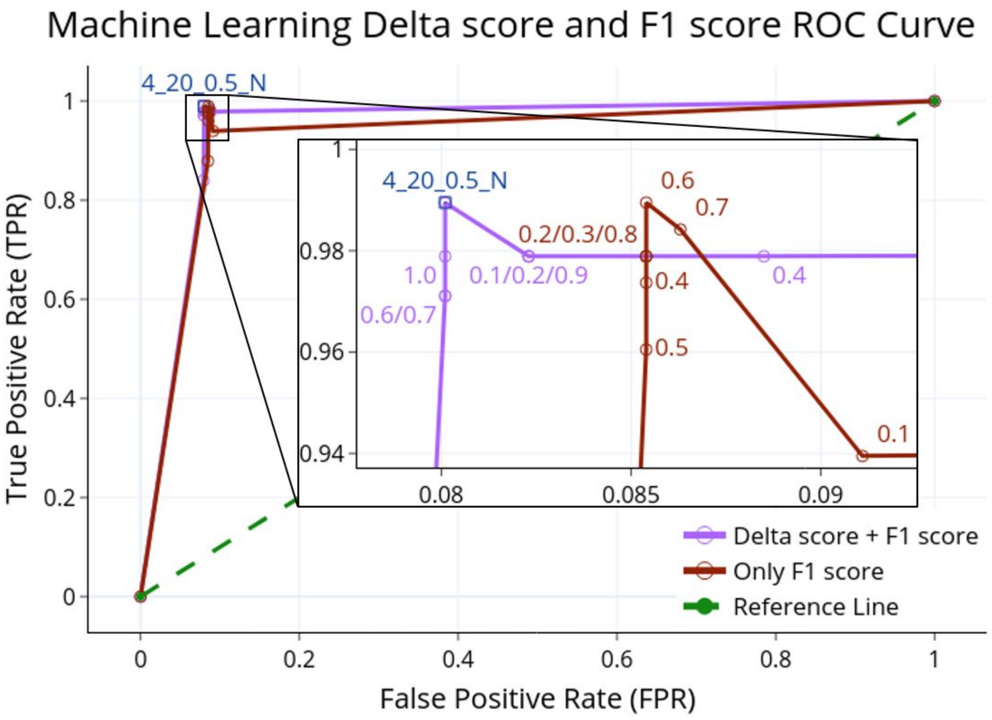
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

Delta score & F1 score

- Separate test to determine value
- Comparison with threshold

Conclusion

- Improved classification results
- Outputs are far more precise (0.60 DS vs 0.92 DS)


Machine Learning Delta score and F1 score ROC Curve

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Conclusion

- Malicious behavior detection in CyberArk PAS

Use cases

- 17+9 Attack techniques performed in test environment
- Logs analysed and 18 use cases defined

Automation

- Two frameworks for log analysis automation: string parsing and machine learning
- Machine learning can be applied with genetic neural networks and bag of words
- Experiments performed for optimal parameters Detector and Classifier
- Addition of Delta score positively influenced the learning process

Future work

Automated pipeline

- Applying frameworks
- Feed forward system

Machine Learning Techniques

- Only genetic neural networks is used with bag of words approach
- Supervised machine learning
- Ability to parse multiple logs compared to a single log for pattern recognition

Extending CyberArk PAS

- In this research, the data was captured using default settings.
- Changing logging, security configuration or applying agents on hosts could be investigated further.

Thank you for your time.

Any questions?

Ivar Slotboom and Mike Slotboom, SNE/UvA

Roel Bierens and Bartosz Czaszynski, Deloitte

Appendix

Related work

Abad et. al (2003)

- Anomaly detection in Intrusion Detection Systems

- Bottom-up approach: Logs → Attacks
- Top-down approach: Attacks → Logs

Meera and Geethakumari (2013)

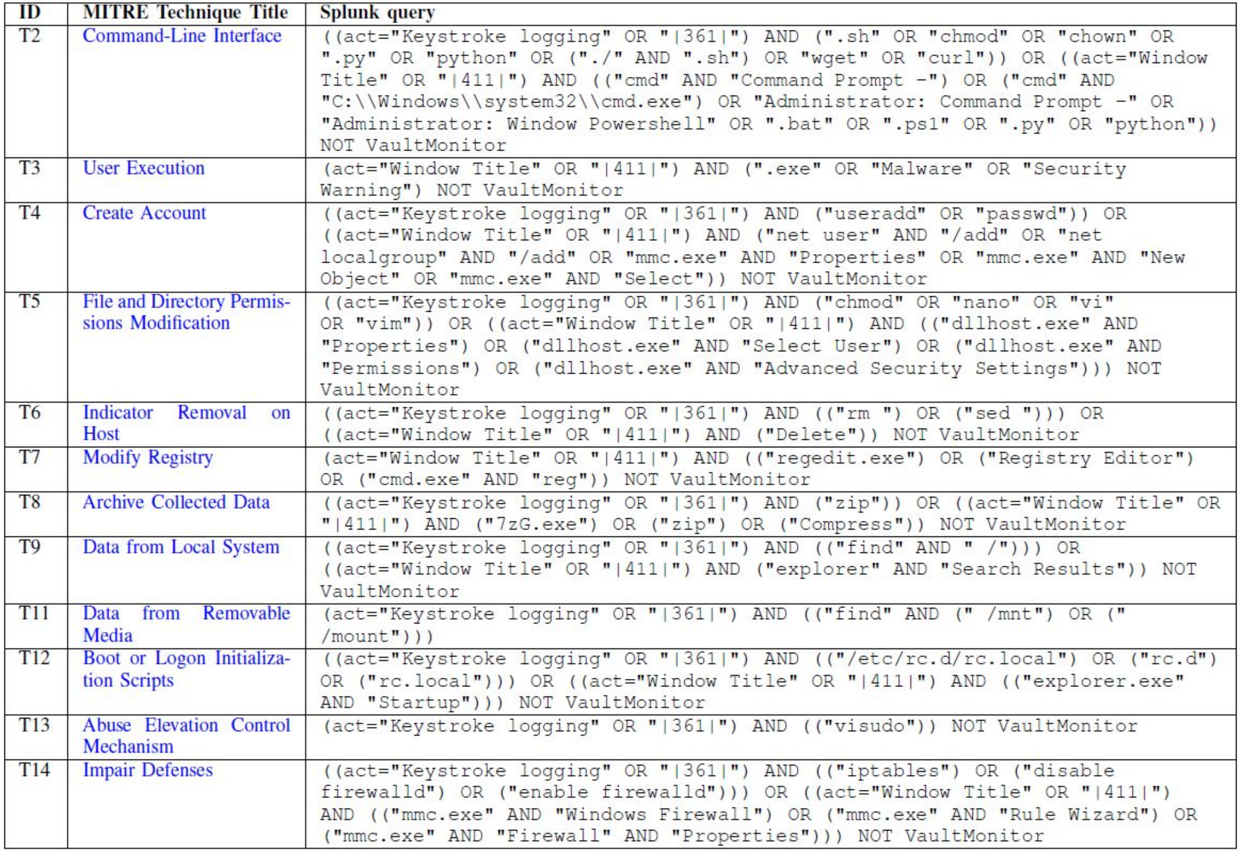
- Cloud API log correlation
- Match and filter on pre-defined atomic conditions

Huizinga (2019)

- OS3 Research
- Analysis of network traffic during a pen test
- Application of supervised machine learning

Use cases

TABLE VIII
SPLUNK QUERIES FROM ATTACK TECHNIQUES



<table>
<tr><th>ID</th><th>MITRE Technique Title</th><th>Splunk query</th></tr>
<tr><td>T2</td><td>Command-Line Interface</td><td>((act="Keystroke logging" OR "|361|") AND (".sh" OR "chmod" OR "chown" OR ".py" OR "python" OR ("./" AND ".sh") OR "wget" OR "curl")) OR ((act="Window Title" OR "|411|") AND (("cmd" AND "Command Prompt -") OR ("cmd" AND "C:\\Windows\\system32\\cmd.exe") OR "Administrator: Command Prompt -" OR "Administrator: Window Powershell" OR ".bat" OR ".ps1" OR ".py" OR "python")) NOT VaultMonitor</td></tr>
<tr><td>T3</td><td>User Execution</td><td>(act="Window Title" OR "|411|") AND (".exe" OR "Malware" OR "Security Warning") NOT VaultMonitor</td></tr>
<tr><td>T4</td><td>Create Account</td><td>((act="Keystroke logging" OR "|361|") AND ("useradd" OR "passwd")) OR ((act="Window Title" OR "|411|") AND ("net user" AND "/add" OR "net localgroup" AND "/add" OR "mmc.exe" AND "Properties" OR "mmc.exe" AND "New Object" OR "mmc.exe" AND "Select")) NOT VaultMonitor</td></tr>
<tr><td>T5</td><td>File and Directory Permissions Modification</td><td>((act="Keystroke logging" OR "|361|") AND ("chmod" OR "nano" OR "vi" OR "vim")) OR ((act="Window Title" OR "|411|") AND (("dllhost.exe" AND "Properties") OR ("dllhost.exe" AND "Select User") OR ("dllhost.exe" AND "Permissions") OR ("dllhost.exe" AND "Advanced Security Settings"))) NOT VaultMonitor</td></tr>
<tr><td>T6</td><td>Indicator Removal on Host</td><td>((act="Keystroke logging" OR "|361|") AND (("rm ") OR ("sed "))) OR ((act="Window Title" OR "|411|") AND ("Delete")) NOT VaultMonitor</td></tr>
<tr><td>T7</td><td>Modify Registry</td><td>(act="Window Title" OR "|411|") AND (("regedit.exe") OR ("Registry Editor") OR ("cmd.exe" AND "reg")) NOT VaultMonitor</td></tr>
<tr><td>T8</td><td>Archive Collected Data</td><td>((act="Keystroke logging" OR "|361|") AND ("zip")) OR ((act="Window Title" OR "|411|") AND ("7zG.exe") OR ("zip") OR ("Compress")) NOT VaultMonitor</td></tr>
<tr><td>T9</td><td>Data from Local System</td><td>((act="Keystroke logging" OR "|361|") AND (("find" AND " /"))) OR ((act="Window Title" OR "|411|") AND ("explorer" AND "Search Results")) NOT VaultMonitor</td></tr>
<tr><td>T11</td><td>Data from Removable Media</td><td>(act="Keystroke logging" OR "|361|") AND (("find" AND (" /mnt") OR (" /mount")))</td></tr>
<tr><td>T12</td><td>Boot or Logon Initialization Scripts</td><td>((act="Keystroke logging" OR "|361|") AND (("/etc/rc.d/rc.local") OR ("rc.d") OR ("rc.local"))) OR ((act="Window Title" OR "|411|") AND (("explorer.exe" AND "Startup"))) NOT VaultMonitor</td></tr>
<tr><td>T13</td><td>Abuse Elevation Control Mechanism</td><td>(act="Keystroke logging" OR "|361|") AND (("visudo")) NOT VaultMonitor</td></tr>
<tr><td>T14</td><td>Impair Defenses</td><td>((act="Keystroke logging" OR "|361|") AND (("iptables") OR ("disable firewalld") OR ("enable firewalld"))) OR ((act="Window Title" OR "|411|") AND (("mmc.exe" AND "Windows Firewall") OR ("mmc.exe" AND "Rule Wizard") OR ("mmc.exe" AND "Firewall" AND "Properties"))) NOT VaultMonitor</td></tr>
</table>

Use cases (cont.)

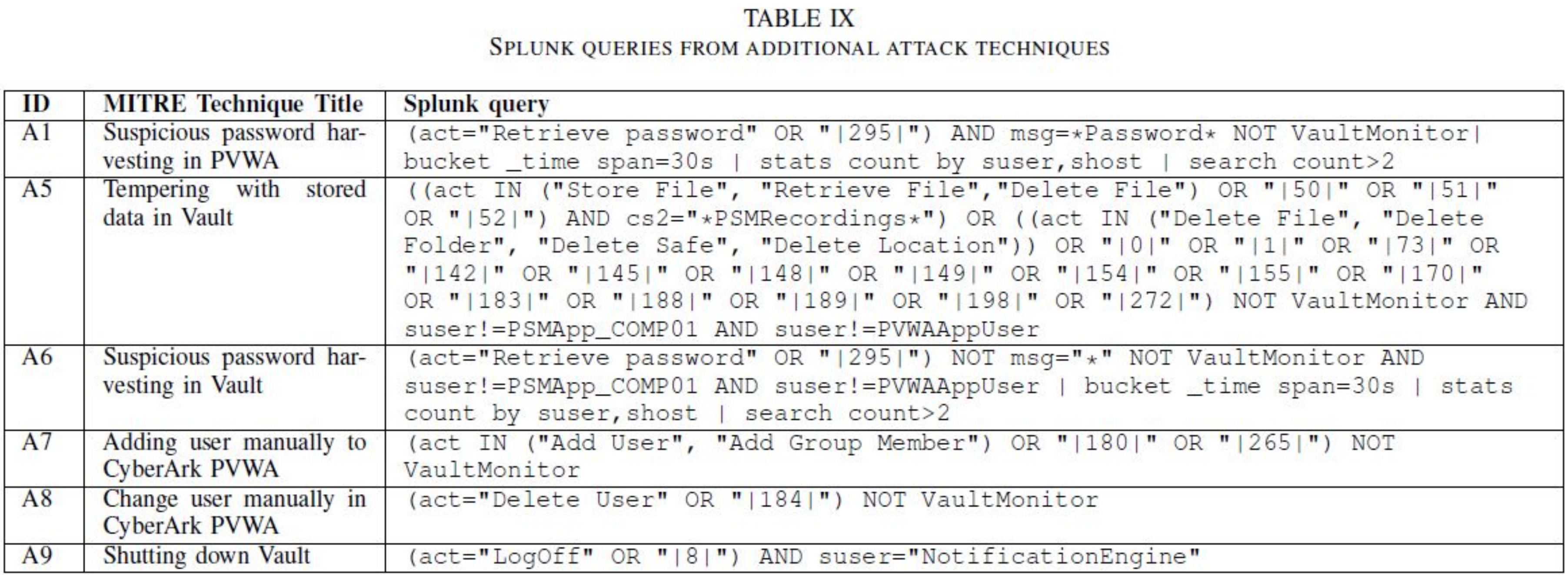


TABLE IX
SPLUNK QUERIES FROM ADDITIONAL ATTACK TECHNIQUES

ID	MITRE Technique Title	Splunk query
A1	Suspicious password harvesting in PVWA	(act="Retrieve password" OR "|295|") AND msg=*Password* NOT VaultMonitor| bucket _time span=30s | stats count by suser,shost | search count>2
A5	Tempering with stored data in Vault	((act IN ("Store File", "Retrieve File","Delete File") OR "|50|" OR "|51|" OR "|52|") AND cs2="*PSMRecordings*") OR ((act IN ("Delete File", "Delete Folder", "Delete Safe", "Delete Location")) OR "|0|" OR "|1|" OR "|73|" OR "|142|" OR "|145|" OR "|148|" OR "|149|" OR "|154|" OR "|155|" OR "|170|" OR "|183|" OR "|188|" OR "|189|" OR "|198|" OR "|272|") NOT VaultMonitor AND suser!=PSMApp_COMP01 AND suser!=PVWAAppUser
A6	Suspicious password harvesting in Vault	(act="Retrieve password" OR "|295|") NOT msg="*" NOT VaultMonitor AND suser!=PSMApp_COMP01 AND suser!=PVWAAppUser | bucket _time span=30s | stats count by suser,shost | search count>2
A7	Adding user manually to CyberArk PVWA	(act IN ("Add User", "Add Group Member") OR "|180|" OR "|265|") NOT VaultMonitor
A8	Change user manually in CyberArk PVWA	(act="Delete User" OR "|184|") NOT VaultMonitor
A9	Shutting down Vault	(act="LogOff" OR "|8|") AND suser="NotificationEngine"