



CyberArk Privileged Access Manager

Microsoft Sentinel SIEM Integration

Business Development

Important Notice

Copyright © 1999-2019 CyberArk Software Ltd. All rights reserved.

This document contains information and ideas, which are proprietary to CyberArk Software Ltd. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of CyberArk Software Ltd.

Azure Sentinel

Collect data at cloud scale across all users, devices, applications, and infrastructure, both on-premises and in multiple clouds.

Detect previously uncovered threats and minimize false positives using analytics and unparalleled threat intelligence from Microsoft.

Investigate threats with AI and hunt suspicious activities at scale, tapping into decades of cybersecurity work at Microsoft.

Respond to incidents rapidly with built-in orchestration and automation of common tasks.

CyberArk Data Connector

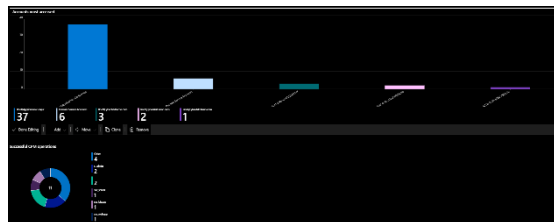
The CyberArk Privileged Access Security connector allows you to connect your Digital Vault Syslog events with Azure Sentinel allowing you to view workbooks, create alerts, create dashboards, trigger actions and manipulate data to provide a comprehensive security overview of your privileged accounts managed by CyberArk.

CyberArk Workbooks

Gain extensive insight and extensive monitoring into CyberArk's Privilege Access Security by analyzing Syslogs generated by the Digital Vault.

Workbooks provide a comprehensive view into all actions run against your Digital Vault to identify where your resources should be directed in order to maintain an efficient access security posture.

Effortlessly monitor what privileged accounts are being managed and accessed, who is accessing those privileged accounts, and track what commands are being run during monitored sessions.



Pre-Installation

This integration workflow requires that you have a valid CyberArk Digital Vault, a Syslog staging server, and an Azure Sentinel workspace. The following guidelines outline what will be needed to fully implement this integration.

Prerequisites

From CyberArk MarketPlace

- Ensure the following files are stored on the Vault server in
<install drive>\Program Files(x86)\PrivateArk\Server\Syslog
- **RFC5424Changes.xml** – This is a translator file that takes each Syslog message from the Vault and pushes it through this translator to build an RFC5424 compliant header to place at the beginning of the message

```

1 #include <string.h>
2
3 int main() {
4     char username[100];
5
6     // Get the user's name
7     printf("Enter your name: ");
8     fgets(username, sizeof(username), stdin);
9
10    // Remove the trailing newline character
11    username[strcspn(username, "\n")] = 0;
12
13    // Print the user's name
14    printf("Hello, %s!\n", username);
15
16    // Get the user's age
17    printf("Enter your age: ");
18    fgets(username, sizeof(username), stdin);
19
20    // Remove the trailing newline character
21    username[strcspn(username, "\n")] = 0;
22
23    // Print the user's age
24    printf("You are %s years old.\n", username);
25
26    // Get the user's email address
27    printf("Enter your email address: ");
28    fgets(username, sizeof(username), stdin);
29
30    // Remove the trailing newline character
31    username[strcspn(username, "\n")] = 0;
32
33    // Print the user's email address
34    printf("Your email address is %s.\n", username);
35
36    // Get the user's phone number
37    printf("Enter your phone number: ");
38    fgets(username, sizeof(username), stdin);
39
40    // Remove the trailing newline character
41    username[strcspn(username, "\n")] = 0;
42
43    // Print the user's phone number
44    printf("Your phone number is %s.\n", username);
45
46    // Get the user's address
47    printf("Enter your address: ");
48    fgets(username, sizeof(username), stdin);
49
50    // Remove the trailing newline character
51    username[strcspn(username, "\n")] = 0;
52
53    // Print the user's address
54    printf("Your address is %s.\n", username);
55
56    // Get the user's password
57    printf("Enter your password: ");
58    fgets(username, sizeof(username), stdin);
59
60    // Remove the trailing newline character
61    username[strcspn(username, "\n")] = 0;
62
63    // Print the user's password
64    printf("Your password is %s.\n", username);
65
66    // Get the user's security question
67    printf("Enter your security question: ");
68    fgets(username, sizeof(username), stdin);
69
70    // Remove the trailing newline character
71    username[strcspn(username, "\n")] = 0;
72
73    // Print the user's security question
74    printf("Your security question is %s.\n", username);
75
76    // Get the user's security answer
77    printf("Enter your security answer: ");
78    fgets(username, sizeof(username), stdin);
79
80    // Remove the trailing newline character
81    username[strcspn(username, "\n")] = 0;
82
83    // Print the user's security answer
84    printf("Your security answer is %s.\n", username);
85
86    // Get the user's account type
87    printf("Enter your account type: ");
88    fgets(username, sizeof(username), stdin);
89
90    // Remove the trailing newline character
91    username[strcspn(username, "\n")] = 0;
92
93    // Print the user's account type
94    printf("Your account type is %s.\n", username);
95
96    // Get the user's account status
97    printf("Enter your account status: ");
98    fgets(username, sizeof(username), stdin);
99
100   // Remove the trailing newline character
101   username[strcspn(username, "\n")] = 0;
102
103   // Print the user's account status
104   printf("Your account status is %s.\n", username);
105
106   // Get the user's account creation date
107   printf("Enter your account creation date: ");
108   fgets(username, sizeof(username), stdin);
109
110   // Remove the trailing newline character
111   username[strcspn(username, "\n")] = 0;
112
113   // Print the user's account creation date
114   printf("Your account creation date is %s.\n", username);
115
116   // Get the user's account expiration date
117   printf("Enter your account expiration date: ");
118   fgets(username, sizeof(username), stdin);
119
120   // Remove the trailing newline character
121   username[strcspn(username, "\n")] = 0;
122
123   // Print the user's account expiration date
124   printf("Your account expiration date is %s.\n", username);
125
126   // Get the user's account last login date
127   printf("Enter your account last login date: ");
128   fgets(username, sizeof(username), stdin);
129
130   // Remove the trailing newline character
131   username[strcspn(username, "\n")] = 0;
132
133   // Print the user's account last login date
134   printf("Your account last login date is %s.\n", username);
135
136   // Get the user's account last login IP address
137   printf("Enter your account last login IP address: ");
138   fgets(username, sizeof(username), stdin);
139
140   // Remove the trailing newline character
141   username[strcspn(username, "\n")] = 0;
142
143   // Print the user's account last login IP address
144   printf("Your account last login IP address is %s.\n", username);
145
146   // Get the user's account last login user agent
147   printf("Enter your account last login user agent: ");
148   fgets(username, sizeof(username), stdin);
149
150   // Remove the trailing newline character
151   username[strcspn(username, "\n")] = 0;
152
153   // Print the user's account last login user agent
154   printf("Your account last login user agent is %s.\n", username);
155
156   // Get the user's account last login device
157   printf("Enter your account last login device: ");
158   fgets(username, sizeof(username), stdin);
159
160   // Remove the trailing newline character
161   username[strcspn(username, "\n")] = 0;
162
163   // Print the user's account last login device
164   printf("Your account last login device is %s.\n", username);
165
166   // Get the user's account last login location
167   printf("Enter your account last login location: ");
168   fgets(username, sizeof(username), stdin);
169
170   // Remove the trailing newline character
171   username[strcspn(username, "\n")] = 0;
172
173   // Print the user's account last login location
174   printf("Your account last login location is %s.\n", username);
175
176   // Get the user's account last login referrer
177   printf("Enter your account last login referrer: ");
178   fgets(username, sizeof(username), stdin);
179
180   // Remove the trailing newline character
181   username[strcspn(username, "\n")] = 0;
182
183   // Print the user's account last login referrer
184   printf("Your account last login referrer is %s.\n", username);
185
186   // Get the user's account last login referrer host
187   printf("Enter your account last login referrer host: ");
188   fgets(username, sizeof(username), stdin);
189
190   // Remove the trailing newline character
191   username[strcspn(username, "\n")] = 0;
192
193   // Print the user's account last login referrer host
194   printf("Your account last login referrer host is %s.\n", username);
195
196   // Get the user's account last login referrer path
197   printf("Enter your account last login referrer path: ");
198   fgets(username, sizeof(username), stdin);
199
200   // Remove the trailing newline character
201   username[strcspn(username, "\n")] = 0;
202
203   // Print the user's account last login referrer path
204   printf("Your account last login referrer path is %s.\n", username);
205
206   // Get the user's account last login referrer query string
207   printf("Enter your account last login referrer query string: ");
208   fgets(username, sizeof(username), stdin);
209
210   // Remove the trailing newline character
211   username[strcspn(username, "\n")] = 0;
212
213   // Print the user's account last login referrer query string
214   printf("Your account last login referrer query string is %s.\n", username);
215
216   // Get the user's account last login referrer fragment identifier
217   printf("Enter your account last login referrer fragment identifier: ");
218   fgets(username, sizeof(username), stdin);
219
220   // Remove the trailing newline character
221   username[strcspn(username, "\n")] = 0;
222
223   // Print the user's account last login referrer fragment identifier
224   printf("Your account last login referrer fragment identifier is %s.\n", username);
225
226   // Get the user's account last login referrer status code
227   printf("Enter your account last login referrer status code: ");
228   fgets(username, sizeof(username), stdin);
229
230   // Remove the trailing newline character
231   username[strcspn(username, "\n")] = 0;
232
233   // Print the user's account last login referrer status code
234   printf("Your account last login referrer status code is %s.\n", username);
235
236   // Get the user's account last login referrer status message
237   printf("Enter your account last login referrer status message: ");
238   fgets(username, sizeof(username), stdin);
239
240   // Remove the trailing newline character
241   username[strcspn(username, "\n")] = 0;
242
243   // Print the user's account last login referrer status message
244   printf("Your account last login referrer status message is %s.\n", username);
245
246   // Get the user's account last login referrer status reason phrase
247   printf("Enter your account last login referrer status reason phrase: ");
248   fgets(username, sizeof(username), stdin);
249
250   // Remove the trailing newline character
251   username[strcspn(username, "\n")] = 0;
252
253   // Print the user's account last login referrer status reason phrase
254   printf("Your account last login referrer status reason phrase is %s.\n", username);
255
256   // Get the user's account last login referrer status text
257   printf("Enter your account last login referrer status text: ");
258   fgets(username, sizeof(username), stdin);
259
260   // Remove the trailing newline character
261   username[strcspn(username, "\n")] = 0;
262
263   // Print the user's account last login referrer status text
264   printf("Your account last login referrer status text is %s.\n", username);
265
266   // Get the user's account last login referrer status code and reason phrase
267   printf("Enter your account last login referrer status code and reason phrase: ");
268   fgets(username, sizeof(username), stdin);
269
270   // Remove the trailing newline character
271   username[strcspn(username, "\n")] = 0;
272
273   // Print the user's account last login referrer status code and reason phrase
274   printf("Your account last login referrer status code and reason phrase is %s.\n", username);
275
276   // Get the user's account last login referrer status code and status message
277   printf("Enter your account last login referrer status code and status message: ");
278   fgets(username, sizeof(username), stdin);
279
280   // Remove the trailing newline character
281   username[strcspn(username, "\n")] = 0;
282
283   // Print the user's account last login referrer status code and status message
284   printf("Your account last login referrer status code and status message is %s.\n", username);
285
286   // Get the user's account last login referrer status code and status reason phrase
287   printf("Enter your account last login referrer status code and status reason phrase: ");
288   fgets(username, sizeof(username), stdin);
289
290   // Remove the trailing newline character
291   username[strcspn(username, "\n")] = 0;
292
293   // Print the user's account last login referrer status code and status reason phrase
294   printf("Your account last login referrer status code and status reason phrase is %s.\n", username);
295
296   // Get the user's account last login referrer status code and status text
297   printf("Enter your account last login referrer status code and status text: ");
298   fgets(username, sizeof(username), stdin);
299
300   // Remove the trailing newline character
301   username[strcspn(username, "\n")] = 0;
302
303   // Print the user's account last login referrer status code and status text
304   printf("Your account last login referrer status code and status text is %s.\n", username);
305
306   // Get the user's account last login referrer status code and status code and reason phrase
307   printf("Enter your account last login referrer status code and status code and reason phrase: ");
308   fgets(username, sizeof(username), stdin);
309
310   // Remove the trailing newline character
311   username[strcspn(username, "\n")] = 0;
312
313   // Print the user's account last login referrer status code and status code and reason phrase
314   printf("Your account last login referrer status code and status code and reason phrase is %s.\n", username);
315
316   // Get the user's account last login referrer status code and status code and status message
317   printf("Enter your account last login referrer status code and status code and status message: ");
318   fgets(username, sizeof(username), stdin);
319
320   // Remove the trailing newline character
321   username[strcspn(username, "\n")] = 0;
322
323   // Print the user's account last login referrer status code and status code and status message
324   printf("Your account last login referrer status code and status code and status message is %s.\n", username);
325
326   // Get the user's account last login referrer status code and status code and status reason phrase
327   printf("Enter your account last login referrer status code and status code and status reason phrase: ");
328   fgets(username, sizeof(username), stdin);
329
330   // Remove the trailing newline character
331   username[strcspn(username, "\n")] = 0;
332
333   // Print the user's account last login referrer status code and status code and status reason phrase
334   printf("Your account last login referrer status code and status code and status reason phrase is %s.\n", username);
335
336   // Get the user's account last login referrer status code and status code and status text
337   printf("Enter your account last login referrer status code and status code and status text: ");
338   fgets(username,
```

NOTE: There is already an RFC5424Changes.xml on the Vault server, there are minor modifications to this translator to more accurately conform to RFC5424 header guidelines and will affect the outcome of the outgoing messages.

- **Sentinel.xsl** – This is a translator file that will take the xml data produced by the Vault Syslog capability and parse it into a CEF standard message to be sent to your syslog staging server

Setup Sentinel in the Azure Portal

Connect logs to Azure Sentinel workspace and log analytics to view and query CyberArk logs using the CyberArk Syslog Data connector

<https://docs.microsoft.com/azure/sentinel/quickstart-onboard>

Syslog forwarding server (syslog-ng, rsyslog)

This integration requires that messages sent from the Vault get received at a Syslog staging server. Once you have made a decision on either syslog-ng or rsyslog follow the instructions in the link below to install the Azure Monitoring Agent to get the messages from the staging server to Sentinel.

Refer to the link below for more detailed information:

<https://learn.microsoft.com/en-us/azure/azure-monitor/agents/agents-overview>

Installation

Setting Up the Vault

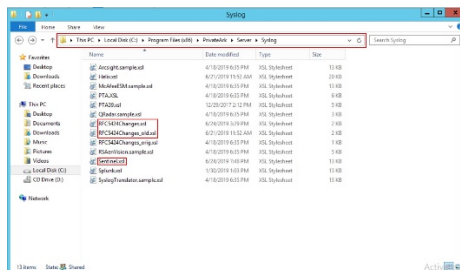
Update the translator files

- Go to the Vault server and navigate to
<install folder>\Program Files(x86)\PrivateArk\Server\Syslog

- Rename the **RFC5424Changes.xsl** that is currently in place

NOTE: If you modify the RFC5424Changes translator, this will modify the header for **ALL** messages coming from the Vault. If you are sending Syslog to multiple locations this will modify the header for those messages too.

- Place the two translator files in the Syslog folder



Modify the dbparm.ini

- Navigate to **\Program Files(x86)\PrivateArk\Server\Conf** -> edit the **dbparm.ini**
- Modify the dbparm to reflect the following values for the SYSLOG section:

```
[SYSLOG]
UseLegacySyslogFormat=No
SyslogTranslatorFile=Syslog\Sentinel.xsl
SyslogServerPort=<port to the Syslog staging server>
SyslogServerIP=<IP for the Syslog staging server>
SyslogServerProtocol=UDP or TCP
SyslogMessageCodeFilter=0-999
```

```
DebugLevel=SYSLOG(2)
[SYSLOG]
UseLegacySyslogFormat=No
SyslogTranslatorFile=Syslog\Sentinel.xsl
SyslogServerPort=514
SyslogServerIP=10.0.0.3
SyslogServerProtocol=TCP
SyslogMessageCodeFilter=0-999
```

NOTE: if you are sending logs to Sentinel as an additional location, use the pipe “|” character to separate the values, for more information on configuring Syslog on the Vault, refer to the following link:

<https://docs.cyberark.com/Product-Doc/OnlineHelp/PAS/Latest/en/Content/PASIMP/Integrating-with-SIEM-Applications.htm?Highlight=syslog>

To have the Vault pick up the changes made to the dbparm.ini, you will need to restart the **PrivateArk Server** service

Configuring your Linux Machine

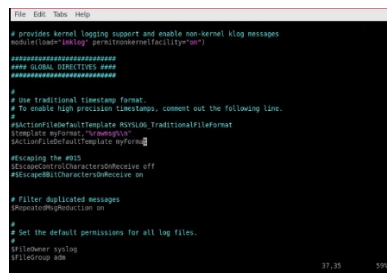
Configuring Linux Syslog Staging Server

1. Create a Linux machine that will receive the CyberArk Syslog messages for the Azure Sentinel Agent to collect
2. Run the configuration scripts that can be found in the CyberArk Data Connectors page in Azure Sentinel

<https://learn.microsoft.com/en-us/azure/sentinel/connect-cef-syslog-ama?tabs=portal>

3. The script will download and install the Log Analytics agent and configure your machine Syslog daemon depending on your Syslog capturing capability (syslog-ng, rsyslog)
4. Note that the CyberArk messages will come in through the “kernel” facility. If using rsyslog, the following syntax can be placed in the `/etc/rsyslog.conf` file:

```
$template myFormat,"%rawmsg%\n"
$ActionFileDefaultTemplate myFormat
```



5. Once the agent picks up on the incoming data, CEF messages will be sent to Log Analytics, below is a link on how to validate connectivity

<https://learn.microsoft.com/en-us/azure/sentinel/connect-cef-syslog-ama?tabs=portal>

6. Depending on your installation of the Syslog staging server, follow the directions in the link provided to install the agent on your staging server:
<https://learn.microsoft.com/en-us/azure/azure-monitor/agents/agents-overview>

Syslog Header Info

Below is how your RFC5424 Header should be coming into the log files and forwarded to Log Analytics.

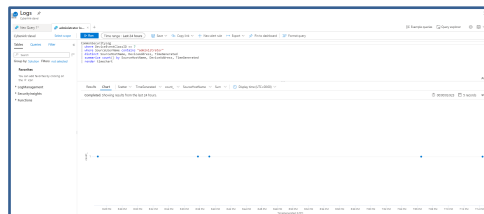
```
<%Priority%>1 yyyy-mm-ddThh:mm:ssZ %Hostname% Cyber-Ark
%ProcessID% %MSGID% -
```

Below is an example of a failed CPM Verification

```
<7>1 2019-06-25T21:32:21Z VAULT Cyber-Ark 38 38 -
```

Data

Once you have data showing up in your Log Analytics view, you can start to query data to build a picture to display in Dashboards and Workbooks.



Workbooks

CEF standard custom label functionality has limitations and to solve those, here are the field mappings done for CyberArk data connector. Refer to the table below for further context.

Old Label	Sentinel Label	xsl KeyName
Safe Name	DestinationUserPrivileges	dpriv
Device Type	FileType	fileType
Affected User Name	SourceUserPrivileges	spriv
Database	DeviceExternalID	deviceExternalId
Other info	destinationProcessName	dproc
Request Id	FileID	fileId
Ticket Id	OldFileID	oldFileId

The workbooks provided by the Azure Sentinel connector are simply examples to get you started. Your enterprise's security view will dictate what fields need to be depicted in your workbooks and Sentinel's ease of use allows for dynamic views of your Vault activity.

To access the workbooks created for this integration, navigate to **Azure Sentinel**, select **Workbooks**, select **CyberArk**.

Troubleshooting

CyberArk

- Ensure that your dbparm configurations for Syslog reflect the correct information for the Syslog staging server.
- Take note of the **SyslogServerProtocol** and ensure that the staging server syslog configs are setup to receive either TCP or UDP depending on what you have set for this parameter.
- If want to know what is being sent from the Vault, change the **DebugLevel** in the dbparm to **SYSLOG(2)**. This will print an output of what is being sent according to the translator file and Syslog settings in the dbparm

Syslog Agent

- If you are seeing kernel messages populate the **/var/log/kern.log** file this should lead to messages populating in Log Analytics within 15-20 minutes. If you are not seeing kernel messages populating on the staging server, you will need to troubleshoot your syslog configurations on the Vault Server
- Ensure that you have messages incoming from the Vault server by running the following command on the Syslog staging server:

tcpdump -A -ni any port <port you set up in the dbparm> -vv

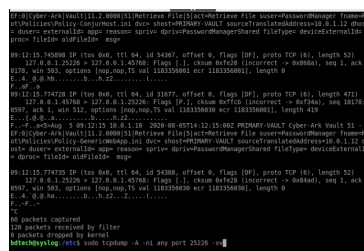
Example

Messages from the Vault:

tcpdump -A -ni any port 514 -vv

Messages being processed by Sentinel agent:

tcpdump -A -ni any port 25226 -vv



- If you have made any modifications to configuration files on the syslog server then you may need to restart the service.

Example:

service rsyslog restart

service syslog-ng restart

- Ensure that what you have set in your syslog config matches what you have set in the dbparm regarding protocol (TCP vs UDP)

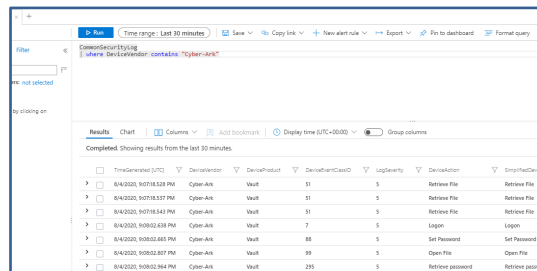
Azure Sentinel

- Under **Azure Sentinel workspace -> Advanced settings -> Connected Sources -> Linux Servers** you should see that you have a connected resource. This may take up to 15-20 minutes for the changes to be reflected.
- Once you are receiving messages in the Log Analytics workspace and you are receiving results using the following query:

CommonSecurityLog

| where DeviceVendor contains "Cyber-Ark"

Now you are ready to begin configuring parsers and implementing workbooks



The screenshot shows the Azure Sentinel query results interface. The query is 'CommonSecurityLog | where DeviceVendor contains "Cyber-Ark"'. The results are displayed in a table with columns: TimeGenerated [UTC], DeviceVendor, DeviceProduct, DeviceVendorClassID, LogSeverity, DeviceAction, and SimpleDevice. The table shows several rows of data, including log entries for 'Vault' and 'Login' actions.

TimeGenerated [UTC]	DeviceVendor	DeviceProduct	DeviceVendorClassID	LogSeverity	DeviceAction	SimpleDevice
6/4/2020, 10:07:18.528 PM	Cyber-Ark	Vault	51	5	Retrieve File	Retrieve File
6/4/2020, 10:07:18.537 PM	Cyber-Ark	Vault	51	5	Retrieve File	Retrieve File
6/4/2020, 10:07:18.541 PM	Cyber-Ark	Vault	51	5	Retrieve File	Retrieve File
6/4/2020, 10:08:02.838 PM	Cyber-Ark	Vault	7	5	Login	Login
6/4/2020, 10:08:02.845 PM	Cyber-Ark	Vault	88	5	Set Password	Set Password
6/4/2020, 10:08:02.857 PM	Cyber-Ark	Vault	88	5	Open File	Open File
6/4/2020, 10:08:02.864 PM	Cyber-Ark	Vault	255	5	Retrieve password	Retrieve password

- If you are not able to see information in your workbook tiles, ensure that there is data for the time you have selected. If this still provides no data, select the log analytics button and ensure that your subscription and workspace is getting properly assigned. If it is not contact Azure Support.

Contacts

CyberArk Business Development

Business Contact	Name	Yael Nishry
	Email	Yael.Nishry@cyberark.com
Engineering Contact	Name	James Stutes
	Email	james.stutes@cyberark.com