



CyberArk VAP (Very Attacked People)

Proofpoint Targeted Attack Protection (TAP) Integration

Business Development

Name of Company: Proofpoint

Website: www.Proofpoint.com

Name of Product: TAP – Targeted Attack Protection

Date: September 2020



Intent

The Proofpoint Targeted Attack Protection (TAP) Integration uses the CyberArk SCIM Discovery module to poll the Proofpoint solution in order to identify users that have been exposed to attacks by clicking on a phishing link, as well as identifying users that are very attacked people as determined by Proofpoint's proprietary Attack Index. The Attack Index quantifies a person's risk based on Actor Type, Targeting Type, Threat Type and Attack Volume. With this information, such users can be added and removed from groups identified in the SCIM discovery configurations. As users attempt to access vaulted credentials through the PVWA, the Conditional Access Integration with Proofpoint can be used to confirm whether the user should be allowed to be exposed to the credentials requested. Other scenarios and use cases can be implemented using the resulting CyberArk groups being synchronized and/or updated from information available from Proofpoint.

CyberArk VAP (Very Attacked People)

Using Proofpoint People API (available here: https://help.proofpoint.com/Threat_Insight_Dashboard/API_Documentation/People_API), through CyberArk SCIM Discovery configuration a scheduled job can be defined to pull the list of VAP (Very Attacked People) emails that correspond to CyberArk Users and either synchronize a CyberArk group with the results, or add/remove them to/from a CyberArk group.

The CyberArk groups could be used to monitor more closely by defining the group to be recorded by PSM, or avoiding exposure via OPM (by disabling shell access to such groups for example) via CyberArk configuration.

Implementation

Prerequisites

- CyberArk SCIM Server version 1.2.0+
- Proofpoint API access

SCIM Discovery Module Configuration

The **DiscoveryConfig.yml** used for the Discovery module are located in the **SCIM Config** safe in the CyberArk Vault. This YAML formatted file defines the work for the entire Discovery module. Below is an explanation of the parameters for the previously described actions/workflows.

CyberArk VAP (Very Attacked People)

discoveryJobs – Defines the parameters for the integration

Parameter	Description/Value	
name	Name given to the Discovery job – “Proofpoint Very Attacked People”	
className	"com.cyberark.discovery.scheduler.ProofpointDiscovery"	
endpoint	"VeryAttackedPeople"	
connection Account	Vaulted account object used to authenticate to Proofpoint API	
	safe	Safe where the account object will be stored
	folder	Folder where the account object resides
	object	Object Name of the account object
schedule	Parameters to define how often Proofpoint VAP information will be retrieved and processed	
	simple	Interval in seconds
	cron	Schedule in cron unix format. Please see the following link for some examples and explanation: http://www.quartz-scheduler.org/documentation/quartz-2.2.2/tutorials/crontrigger
options	Configurations for interacting with the capability	
	proxyURL	String value for Proxy support
	pageSize	The number of entries on a single page of returned users from the Proofpoint API call
	syncToGroup	Group that users will be placed in that have been identified as Very Attacked People by Proofpoint. Same group used in the ticketing integration configuration
	addToGroups	Tree label for list of groups users are added to that have been identified by the Very Attacked People logic
	-“TestGroup1”	example vault group
	-“FallenAdmins”	example vault group

	removeFromGroups	Tree label for list of groups users are removed from that have been identified by the Very Attacked People logic
	-“powerusers”	example vault group
	-“Vault Admins”	example vault group

Java KeyStore and Truststore Commands

This is an **example** of taking a p12 certificate file and storing it in the Java KeyStore as a jks and then converting it to a PKCS12. These instructions are helpful to setup HTTPS for the SCIM Server.

```
C:\Program Files (x86)\Java\jre1.8.0_xxx\bin>keytool.exe -importkeystore -  
destkeystore <MY_KEYSTORE>.jks -srckeystore <MY_FILE>.pfx/p12 -srcstoretype  
PKCS12 -deststoretype jks -deststorepass <PASSWORD> -srcalias <ALIAS> -destalias  
<NEW_ALIAS>
```

```
keytool.exe -importkeystore -srckeystore <MY_KEYSTORE>.jks -destkeystore  
<MY_KEYSTORE>.p12 -srcstoretype JKS -deststoretype PKCS12 -deststorepass  
<PASSWORD>
```

This is an example of storing a Root certificate and storing it in the **cacerts** truststore. You will be prompted for a password for this command and the OOB truststore password is “changeit”

```
C:\Program Files (x86)\Java\jre1.8.0_xxx\bin>keytool.exe -import -  
trustcacerts -file C:\Certificates\RootCA.crt -alias CYBERARK -keystore cacerts
```