KillerAppz®

Joint solution brief for

CyberArk®

Jack DeSive, Integration Developer & Analyst

Last Modified: Mar 21st, 2019

Solution Summary

The CyberArk® integration with KillerAppz® enables for better coordination, faster and safer operation for accessing and maintaining privileged account management using KillerAppz orchestration playbooks. The integration provides engineers & analysts all the key information and prescriptive instructions they need to create, update, and delete password retrieval for a customer's privileged accounts. Adding ability for this automation helps the development team, test teams and security engineers to take a proactive approach in better defining their IAM practice, rather than a reactive approach to their changing enterprise.

Integration Overview	
KillerAppz®	Provisioned with CyberArk Connector
CyberArk® Requirements	Central Credential Provider v10.5, SCIM Server v1.1.1
CyberArk® Solution	Privileged Account Management, Password Retrieval
CyberArk Use Case	Privileged Account create, update, delete and password retrieval using secure API's
CyberArk Applications	PAM
Uses Custom Application	No
Requires On-Demand License	No

Solution Overview

Solution Benefits

Out-of-the-box integration allows the customer to integrate with CyberArk, and securely create integrations, maintain and update privileged account management platform using one solution.

KillerAppz + CyberArk Privileged Account Management (PAM) integration: Users can use their one or multiple ITSM software (Jira, ServiceNow, BMC Remedy, etc.) to make requests for either creation of a new privileged account with the appropriate description of why a new PAM account is required based on the request and once approved by the appropriate teams this request via KillerAppz is passed to CyberArk automatically by the invocation of the CyberArk API. This orchestrates the privileged account creation under CyberArk. If an account already exists, the ticket under the ITSM software is then updated with the response returned by the CyberArk API. Additionally, if there is a need to update an existing privileged account similar workflow happens and the ticket gets updated. Integration covers creation, updating, and deletion of the PAM accounts. The same connector can be tailored to address other use cases that an organization may already have with minimal configuration changes.

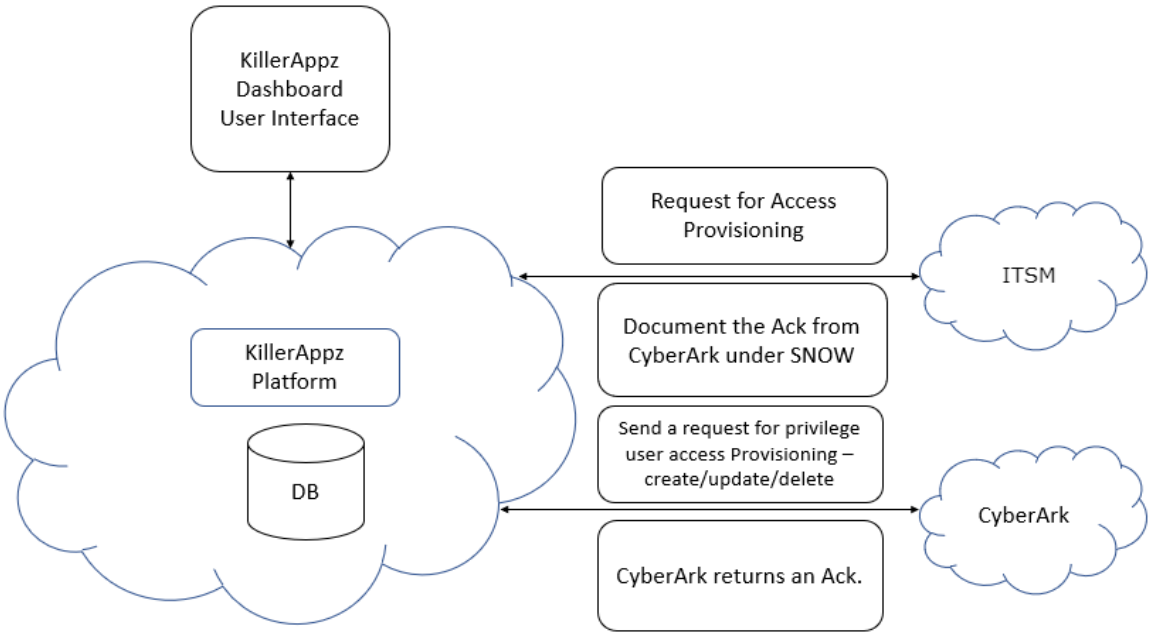


Figure – 01: KillerAppz + CyberArk

KillerAppz + CyberArk Password Retrieval: Customers using the KillerAppz platform would be able to utilize connectors specific to automation testing example – Jenkins. In order to securely execute automated scripts or jobs, users for compliance purposes may not store the privileged account password in free text rather would retrieve it for the session during the script execution and log the purpose and time when the password was retrieved for this purpose, KillerAppz platform simply makes the call to the CyberArk API and does then pass the PAM account details.

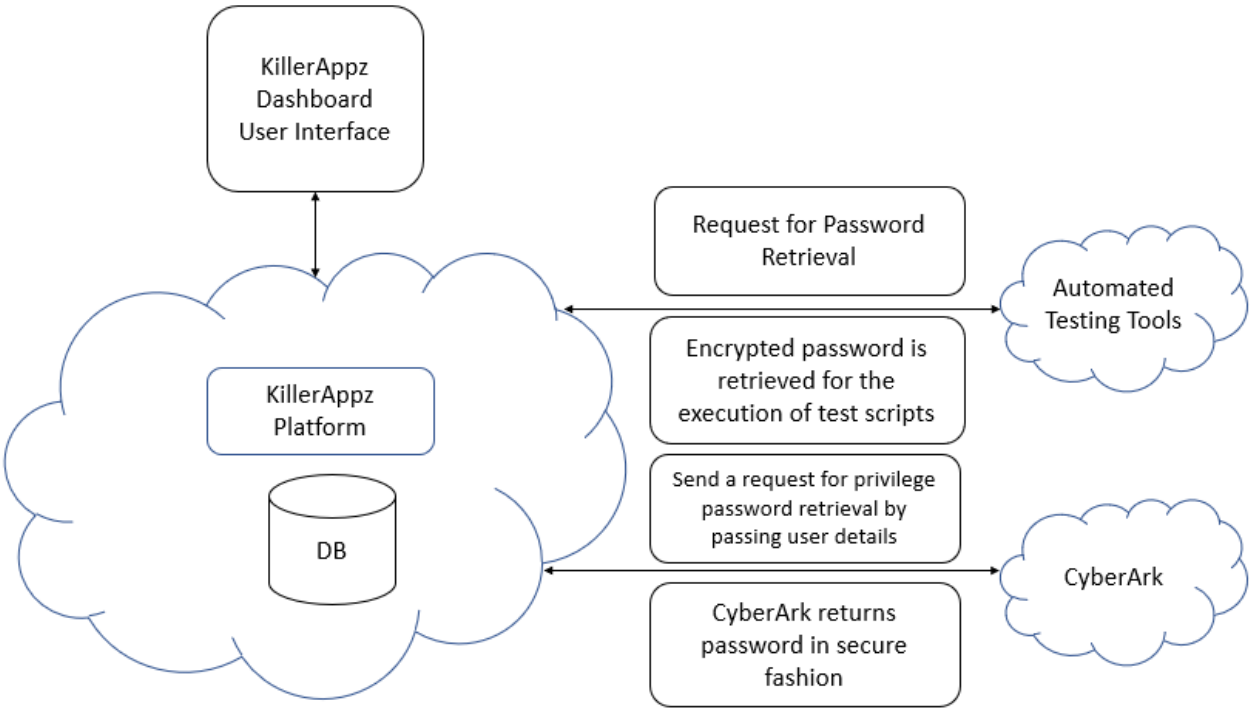


Figure 02: KillerAppz + CyberArk Password Retrieval Use Case

API Details:

Verify the credentials to establish a secure connection with the CyberArk Connector:

POST http://<server>/user/test_connection

Comments (0)

http://<server>/user/test_connection

Check Connector Connection

Headers

Content-Type	application/json

Body raw (application/json)

```
{
    "orgId": 1,
    "username": "client-user",
    "password": "*************",
    "url": "http://services-uscentral.skytap.com:12571",
    "refreshRate": 30000,
    "appId": "SOFTwarfare",
    "safe": "test",
    "folderName": "root"
}
```

Example Request http://<server>/user/test_connection

```
curl --location --request POST "http://%3Cserver%3E/user/test_connectio
n" \
  --header "Content-Type: application/json" \
  --data "{
    \"orgId\": 1,
    \"username\": \"client-user\",
    \"password\": \"*************\",
    \"url\": \"http://services-uscentral.skytap.com:12571\",
    \"refreshRate\": 30000,
    \"appId\": \"SOFTwarfare\"
```

User Configuration API:

POST http://<server>/user/config

Comments (0)

http://<server>/user/config

User Configuration for Connector

Headers

Content-Type	application/json

Body raw (application/json)

```
{
    "orgId": 1,
    "username": "client-user",
    "password": "*************",
    "url": "http://services-uscentral.skytap.com:12571",
    "refreshRate": 30000,
    "appId": "SOFTwarfare",
    "safe": "test",
    "folderName": "root"
}
```

Example Request http://<server>/user/config

```
curl --location --request POST "http://%3Cserver%3E/user/config" \
  --header "Content-Type: application/json" \
  --data "{
    \"orgId\": 1,
    \"username\": \"client-user\",
    \"password\": \"*************\",
    \"url\": \"http://services-uscentral.skytap.com:12571\",
    \"refreshRate\": 30000,
    \"appId\": \"SOFTwarfare\",
    \"safe\": \"test\"
```

Check User if already exists in the CyberArk Safe:

GET http://<server>/user/<id>/check_user

Comments (0)

http://<server>/user/<id>/check_user

Check user existance before creation

Example Request http://<server>/user/<id>/check_user

```
curl --location --request GET "http://%3Cserver%3E/user/%3Cid%3E/check_user"
```

Create a new PAM user using CyberArk API:

POST http://<server>/ticket/new

Comments (0)

http://<server>/ticket/new

Create a New PAM user in CyberArk

Headers

Content-Type	application/json

Body raw (application/json)

```
{
    "orgId": 1,
    "username": "Demo-user-05",
    "password": "*************",
    "requestBy": "CT"
}
```

Example Request http://<server>/ticket/new

```
curl --location --request POST "http://%3Cserver%3E/ticket/new" \
  --header "Content-Type: application/json" \
  --data "{
    \"orgId\": 1,
    \"username\": \"Demo-user-05\",
    \"password\": \"*************\",
    \"requestBy\": \"CT\"
}"
```

List of Request for CyberArk and ServiceNow (ITSM system used) Integration:

GET http://<server>/ticket/list/<orgId>/<page>

Comments (0)

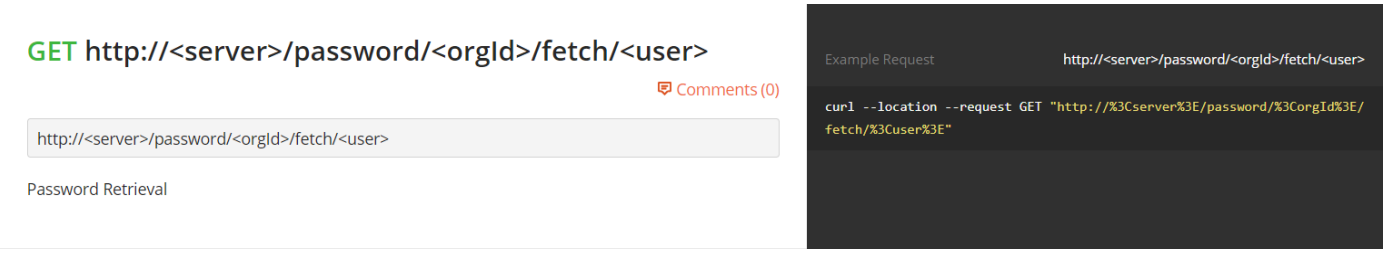
http://<server>/ticket/list/<orgId>/<page>

List of Requests through Integration

Example Request http://<server>/ticket/list/<orgId>/<page>

```
curl --location --request GET "http://%3Cserver%3E/ticket/list/%3CorgId%3E/%3Cpage%3E"
```

Password Retrieval API:



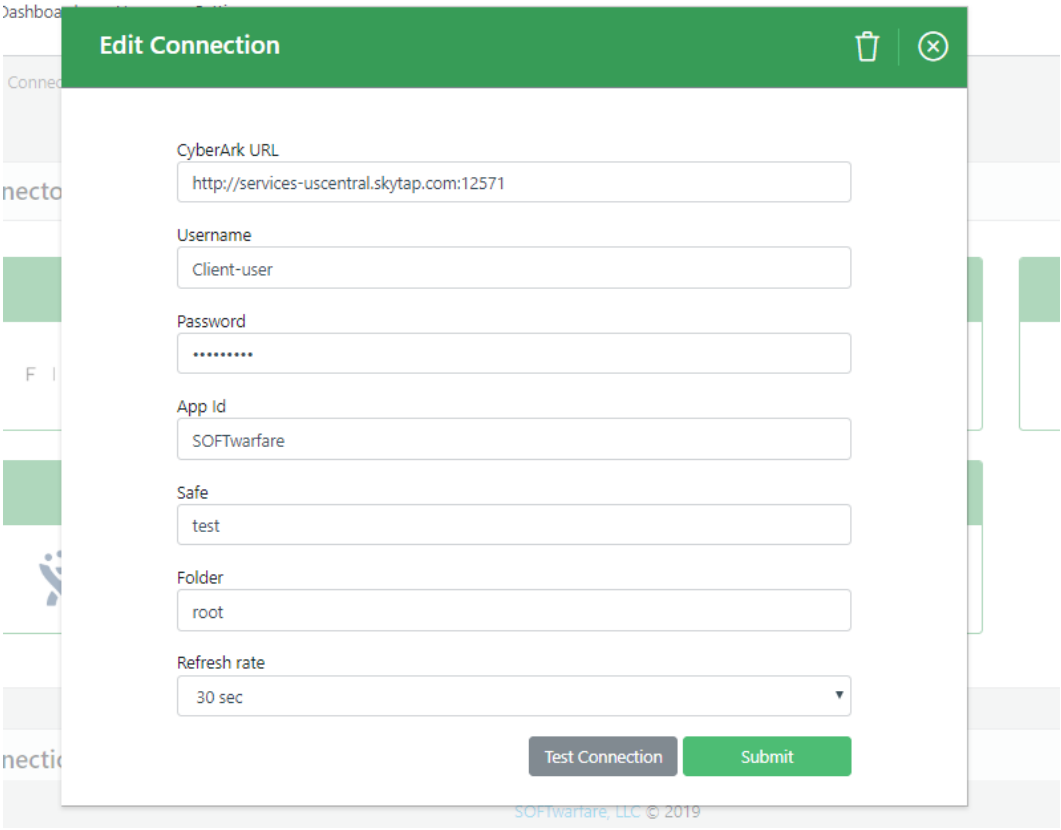
KillerAppz Configuration

To configure KillerAppz to integrate with CyberArk, perform the following steps below:

1. Deploy KillerAppz 1.x version and above.
2. Go to **Connections > Add New Connection**
3. Locate the **CyberArk** logo
4. Click on the CyberArk button to **Add instance** to create and configure a new integration. You should configure the following settings:

CyberArk Server URL	**The server where the integration resides.**
Username/Password	**Credentials for the CyberArk Instance**
App ID	**Customer Application ID – Unique Instance Identifier**
Safe	**Safe within the CyberArk Instance**
Folder	**Folder that stores all the accounts with in the CyberArk Safe**
Refresh Rate	**Refresh rate defines the interval in which KillerAppz must synch with the CyberArk Instance. Refresh rate could be either set to 30 sec, 1 min, 10 minutes, 30 minutes or 60 minutes.**

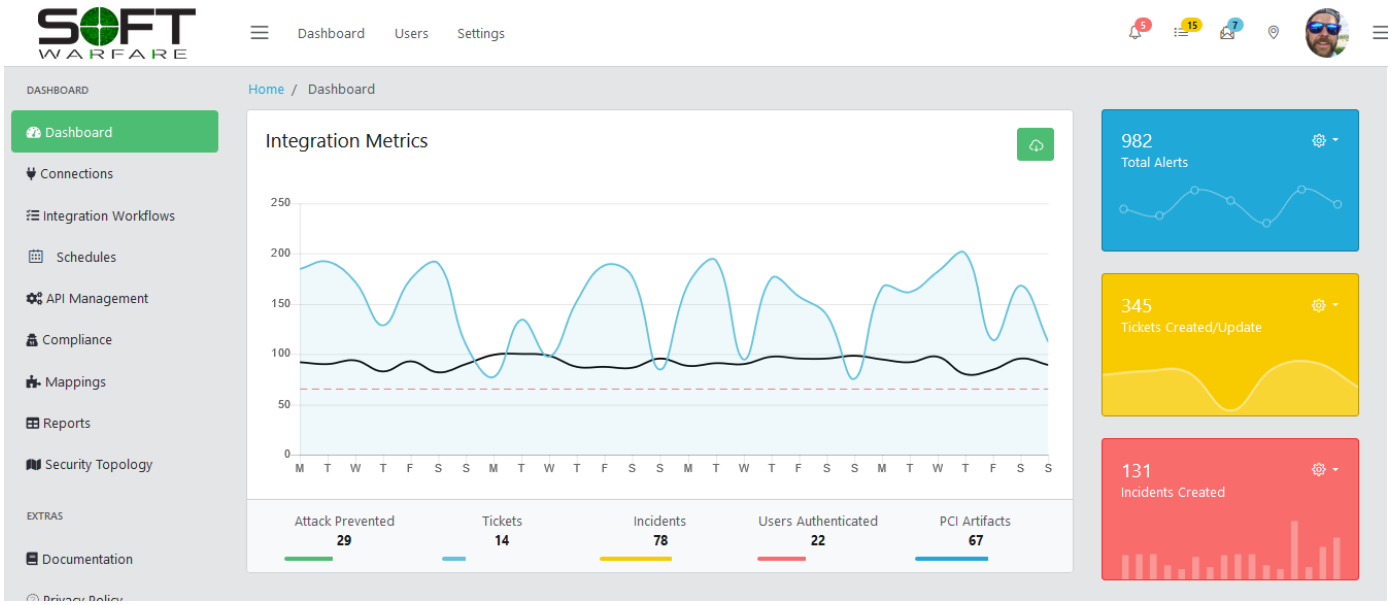
Press the Submit button to establish the connection.



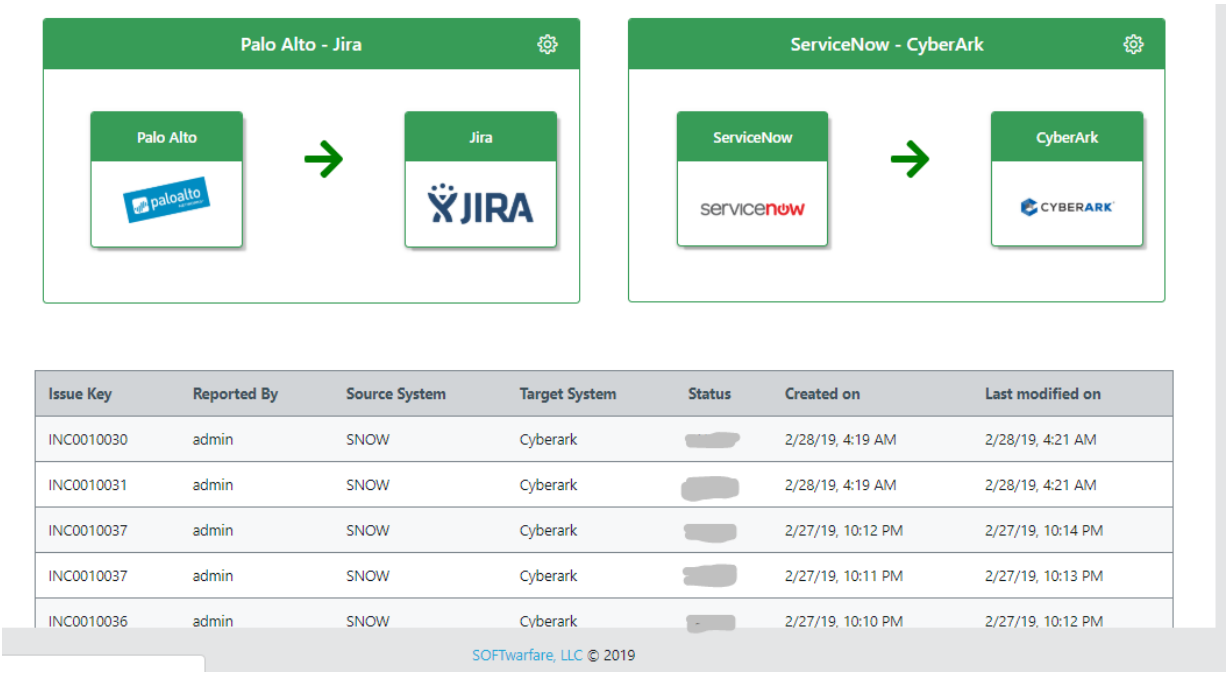
Connection

5. If you are experiencing issues with the service configuration, please contact SOFTwarfare support at support@softwarfare.com

Dashboard:



System Orchestration:



Certification Environment for Privileged Account Management (PAM) and Password Retrieval use case using CyberArk

Date Tested: March 01st, 2019

Product Name	Version Information	Operating System
KillerAppz 1.0	1.0	Docker containers on Linux
CyberArk		Windows 2016 – Cloud