BeyondTrust

Password Safe
Cache User Guide

Powered By PowerBroker

TC:5/14/2019

Table of Contents

Overview **3**
Requirements: Roles and Settings **4**
Requestor & Requestor/Approver Roles 4
ISA Role 4
Access Policy 4
Managed Account Settings 4
Supported Operating Systems 5
Supported APIs 5
Installation Specifications **6**
Windows 6
Linux 6
Configuration **7**
Usage: cfg [options] 8
Examples 8
Advanced Settings **10**
Windows 10
Linux 10

Overview

Password Cache is a lightweight proxy for the Password Safe API, providing high performance throughput for password requests.

Running as a specified Password Safe user, Password Cache makes "View Password"-type requests to Password Safe for all Managed Accounts (for which the user has Requestor or Requestor/Approver roles defined) via the Password Safe API, caching the returned system/account details, request details, and credentials in an encrypted state.

API calls to the Password Cache serve the locally cached data. Requests are refreshed every five minutes or sooner if a request is due to expire before that time.

If communication with the server is lost, the last known good credentials are served from the local cache, even if the associated request has expired.

Requirements: Roles and Settings

Requestor & Requestor/Approver Roles

The Password Safeuser running the Password Cache must have at least one Managed Account Smart Rule configured with the Requestor or Requestor/Approver role.

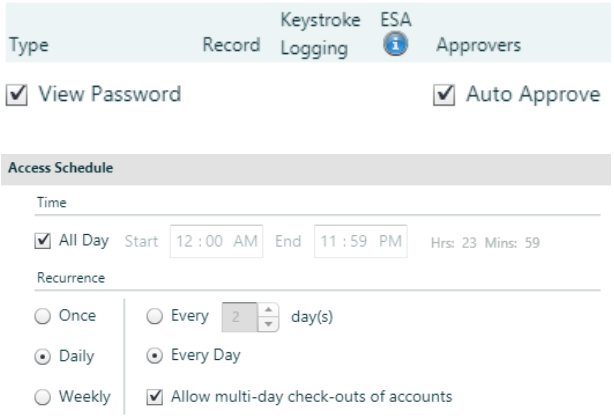
ISA Role

The Password Cachedoes not currently support ISA-based password requests; therefore, it's important to ensure the user running the cache does not have the ISA role defined for any Managed Account Smart Rules.

Access Policy

Auto Approval

The Managed Account Smart Rule configured with the Requestor or Requestor/Approver roles must have an Access Policy assigned that has **View Password** access set to **Auto Approve**.



Daily Recurrence - Multi-day Checkouts

If the Access Policy is configured for **Daily** recurrence, ensure **Allow multi day-check-outs of accounts** is enabled.

Managed Account Settings

Enable for API Access

Ensure this option is enabled for Managed Accounts that will be cached.

Default Release Duration

The Default Release Duration is used to determine how long account credentials are cached before being renewed.

Concurrent Requests

If the Managed Accounts configured to be cached will also be used by other Password Safe users at the same time, concurrent requests should be set to zero (0 - unlimited) or a value greater than one. Requests performed by the Password Cache count as a request.

Supported Operating Systems

- Windows Server 2012 R2 and above releases
- RHEL/Centos 64 bit 6.8 and above releases

Supported APIs

- POST Auth/SignAppIn
- POST Auth/Signout
- GET Requests
- POST Requests
- POST Aliases/{aliasId}/Requests
- GET Credentials/{requestId}
- GET Aliases/{aliasId}/Credentials/{requestId}
- GET ManagedAccounts
- GET ManagedAccounts?systemName={systemName}&accountName={accountName}
- GET Aliases

For details on each method, please see the BeyondInsight and Password Safe API Guide.

Installation Specifications

Windows

- **Installer**: msiexec.exe /i PSPCA-<version>.msi
- **Location**: C:\Program Files (x86)\BeyondTrust\Password Cache\pspca
- **Service Control**: sc stop pspca, sc start pspca

Linux

- **Installer**: rpm -i PSPCA-<version>.x86_64.rpm
- **Location**: /opt/pbps/pspc
- **Service Control**: systemctl stop pspca, systemctl start pspca

Configuration

To configure the cache, call **Password Cache** with the **cfg** options **pspca cfg <args>**.

```
# /opt/pbps/pspca cfg
```

Config:

- Log File (log_file): /var/opt/pbps/log/pspca.log
- Log Level (log_level): INFO
- Password Safe:
    - Host (host): pbps_bi.example.com
    - API RunAS (username): psreq
    - API Key (key): ****************************
- REST API Server: Listen Address (address): 0.0.0.0:443

Client API (Password Cache connections to Password Safe):

- Certificate Validation (password_safe_verify): disabled
- Ciphers List:
    - ECDH+AESGCM:ECDH+CHACHA20:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
- SSLv2: disabled
- SSLv3: disabled
- TLSv1: disabled
- TLSv1.1: enabled
- TLSv1.2: enabled

REST Server (API Client connections to Password Cache):

- Certificate (cache_certificate): bi_client.example.com
    - Issuer: ca.company.com
    - Fingerprint: 96 47 18 4a db 25 d8 42 84 c4 ad e3 08 58 1f 1f ba 9a bc 91
- Certificate Validation (cache_client_verify): disabled
- Ciphers List:
    - ECDH+AESGCM:ECDH+CHACHA20:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
- SSLv2: disabled
- SSLv3: disabled
- TLSv1: disabled

- TLSv1.1: enabled
- TLSv1.2: enabled

Usage: cfg [options]

-L --log_file=<arg>	Log File name
-l --log_level=<arg>	Logging level (error, warning, info, debug, trace)
-h --host=<arg>	Password Safe host[:port]
-a --address=<arg>	Password Cache Listen Address[:port]
-u --username=<arg>	<username> Password Safe API requestor username
-k --key=<arg>	<key> Password Safe API Key
-c --client_certificate=<arg>	Password Safe Client certificate file (pem)
-V --password_safe_verify=<arg>	Password Safe certificate validation 0=no server validation 1=server validation required
-C --client_clear	Clears the Password Safe client certificate
-T --password_safe_ca=<arg>	Trusted Password Safe CA certificate file(s) (pem)
-s --cache_certificate=<arg>	Password Cache server certificate file (pem)
-v --cache_client_verify=<arg>	Password Cache client certificate validation 0=no client validation 1=client validation required
-t --cache_client_trusted_ca=<arg>	Password Cache trusted client CA certificate file(s) (pem)
-p --pem=<arg>	PEM encoded private key for Password Safe or Cache certificate
-P --pem_passwd=<arg>	PEM private key passphrase
--export=<arg>	Export the Password Cache configuration
--import=<arg>	Import the Password Cache configuration
--export_db=<arg>	Export the Password Cache data
--import_db=<arg>	Import the Password Cache data
--passwd=<arg>	Password to be used to encrypt/decrypt the exported Password Cache configuration
-? --help	Display this usage message

Examples

Configure the target Password Safe server that the Password Cache will communicate with:

```
# /opt/pbps/pspca cfg -u psreq -k 638AA550-37C4-7126-A9C1-22186D5A40A0 -h pbps_bi.example.com
```

To validate the Password Safe Server Certificate, define a Trusted CA and require validation:

```
# /opt/pbps/pspca cfg -T password_safe_ca.pem -V 1
```

To connect the Password Cache to the Password Safe REST API using the Client Certificate:

```
# /opt/pbps/pspca cfg -c client_cert.pem -p client_key.pem -P <pem_password>
```

To change the local configuration for logging and the listen port of the Password Cache:

```
# /opt/pbps/pspca cfg -L /var/log/pspca.log -l warning -a 0.0.0.0:8443
```

To provide custom settings for the Server Certificate used by the Password Cache REST interface:

Note: *By default this interface uses a self signed certificate generated by the Password Cache. This can be modified to provide custom settings for the Certificate.*

```
# /opt/pbps/pspca cfg -s server_cert.pem -p server_key.pem -P <pem_password>
```

To require client certificates to be provided to the Password Cache REST interface using a defined Trusted Client CA and require validation:

```
# /opt/pbps/pspca cfg -t client_ca.pem -v 1
```

To export the Password Cache configuration for recovery and/or replicating the Cache:

```
# /opt/pbps/pspca cfg --export=cache_config.cfg --export_db=cache_data.cfg --passwd <secret>
```

To import the Password Cache configuration for recovery and/or replicating the Cache:

```
# /opt/pbps/pspca cfg --import=cache_config.cfg --import_db=cache_data.cfg --passwd <secret>
```

Advanced Settings

The following advanced settings can be configured outside the configuration tool:

- **LogFile**: Location of log file. (default **/var/opt/pbps/log/pspca.log**)
- **runuser**: The unprivileged user that will be used to run the cache service on Linux. (default **nobody**)
- **rotation_policy**: Set to **1** if **Allow API Rotation Override** is enabled in Password Safe's access policy.
- **http_rest**: Define custom settings for the HTTP REST interface.
- **listen_port**: The port the cache will use to listen for incoming API calls. (default **443**)
- **listen_host**: The interface the cache will use to listen for incoming API calls. (default **0.0.0.0**)

Windows

Advanced settings are stored in the registry:

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BeyondTrust\PBPS\pspca_cfg]

- "LogFile"="C:\Program Files (x86)\BeyondTrust\Password Cache\logs\pspca.log"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BeyondTrust\PBPS\pspca_cfg\http_rest]

- "listen_port"=dword:000001bb
- "listen_host"="0.0.0.0"

Linux

The advanced options are stored in a JSON format in **/etc/opt/pbps/pspca.conf**.

```
{
"LogFile": "/var/opt/pbps/log/pspca.log", "runuser": "nobody",
"http_rest": { "listen_port": 443,
"listen_host": "0.0.0.0"
}
```