****

Privileged Access Management (PAM)

Onboarding Automation

December 2020

**TABLE OF CONTENTS**

[1 Version Control 3](#_Toc59216216)

[1.1 Revision History 3](#_Toc59216217)

[1.2 Review History 3](#_Toc59216218)

[1.3 Approval History 3](#_Toc59216219)

[1.4 References 4](#_Toc59216220)

[1.5 Terms 4](#_Toc59216221)

[2 Introduction 4](#_Toc59216222)

[2.1 Executive Summary 4](#_Toc59216223)

[2.2 Scope 4](#_Toc59216224)

[2.3 Intended Audience 4](#_Toc59216225)

[3 Utility 5](#_Toc59216226)

[3.1 Deployment server 5](#_Toc59216227)

[3.2 Utility location 5](#_Toc59216228)

[3.3 Utility files 5](#_Toc59216229)

[3.3.1 Configuration file – app.cfg 5](#_Toc59216230)

[3.3.2 Shell file – psrun.sh 7](#_Toc59216231)

[3.3.3 Input file - newservers.csv 12](#_Toc59216232)

[3.4 Utility execution 13](#_Toc59216233)

[3.5 Logging/Audit 16](#_Toc59216234)

# Version Control

## Revision History

This section represents the change history of the document. All revisions of the document must be tracked by identifying a new version number, the date it was modified, the person making the change, and the reason for the change.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.0 | 12/17/2020 | KPMG | Initial draft |
|  |  |  |  |

## Review History

|  |  |  |
| --- | --- | --- |
| **Version** | **Reviewed By** |  |
| 1.0 |  |  |

## Approval History

The signing of this document acknowledges each individual’s concurrence with the validity and accuracy of the information contained in this document.

|  |  |  |
| --- | --- | --- |
| **Version** | **Approved By** | **Date** |
| 1.0 | N/A |  |

## References

|  |  |
| --- | --- |
| **Document** | **Author** |
| BeyondInsight and Password Safe API Guide 6.10 | BeyondTrust |

## Terms

|  |  |
| --- | --- |
| **Term** | **Description** |
| Privileged Access Management (PAM) Administrator | Personnel is responsible for the administration and support of the BeyondTrust PAM deployment. |
| BeyondTrust | BeyondTrust Privileged Management is an enterprise-class, gold-standard privileged management solution that helps security and IT organizations achieve compliance, control privileged access, and prevent and contain breaches. |
| Database | A database is a repository used to store information or data. |
| Linux / Unix | Linux / Unix is the operating system used. |
| Password Safe | Password Safe is an automated password and privileged session management solution offering secure access control, auditing, alerting, and recording for any privileged account. |

# Introduction

## Executive Summary

Royal Caribbean Limited (RCL) is leveraging the BeyondTrust’s Privileged Access Management (PAM) solution to secure the access to privileged accounts throughout their global enterprise. Access and/or usage of a privileged account is based on the principle of least-privilege in which it can be functionally enabled through secure proxy remote access, automated password management, and a robust logging and monitoring capability.  
Detailed within this document is the onboarding automation process, utilizing the native APIs provided by BeyondTrust.

## Scope

This document is a guide on the automation of new servers onboarding into Beyond Insight and PasswordSafe through a utility.

## Intended Audience

This document is intended for the PAM stakeholders responsible for the operational support for the BeyondTrust Privileged Access Management Suite solution.

# Utility

## Deployment server

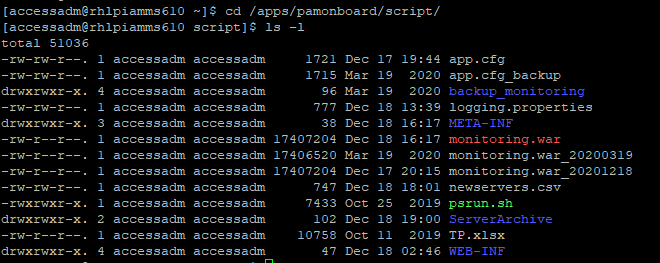
The script is run from the IAM/PAM monitoring servers. These are Linux based OS’s and there are 2 servers per environment ie. 2 servers in Stage and 2 in Prod.

Following are the server details:

|  |  |  |
| --- | --- | --- |
|  | **Server 1** | **Server 2** |
| **Stage** | Rhlsiamms610.na.rccl.com (10.18.42.238) | Rhlsiamms611.na.rccl.com (10.18.42.239) |
| **Production** | Rhlpiamms610.na.rccl.com (10.18.42.236) | Rhlpiamms611.na.rccl.com (10.18.42.237) |

## Utility location

The automation utility is a java based utility that is packaged as part of a .war file and run on the command line from the monitoring server. The ‘monitoring.war’ file is the name of the file that has the necessary java classes. The files are stored in the /apps/pamonboard/script’ folder.   
  
The image below shows the file location.



## Utility files

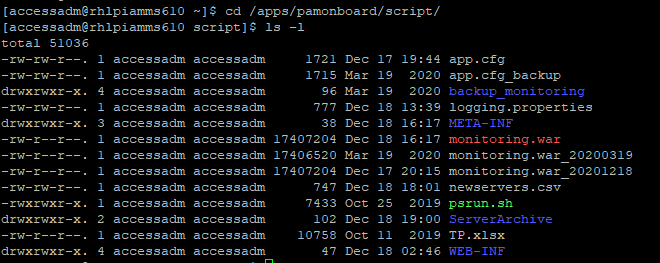
In order for the onboarding utility to run, it requires the following files :

1. **App.cfg**
2. **Psrun.sh**
3. **Newservers.csv**

### 3.3.1 Configuration file – app.cfg

Essential configuration details are stored in a configuration file in the same directory that the war file is deployed. The configuration file stores the details as following :

* API username
* Encrypted apikey
* Asset smart rules
* Managed account smart rule
* Functional account names
* Password policies
* DSS key rules
* Default accounts names
* Password policy id per platform

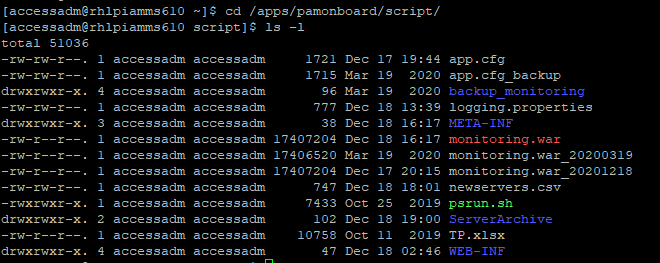


Below is the app.cfg file :

|  |
| --- |
| #system key, change it in the beginning. If change later, need re-encrypt all encrypted values.  system.key=change it before 1st time use the system  PasswordSafeHost=pamapi.rccl.com  PasswordSafeUser=api\_pam  PasswordSafeApiKey = *<hidden>*  monitoring.script.registerNewServer.smartRule.Windows=PS\_Windows  monitoring.script.registerNewServer.smartRule.Linux=PS\_Linux  monitoring.script.registerNewServer.smartRule.AIX=PS\_UNIX  monitoring.script.registerNewServer.smartRule.Oracle=PS\_Oracle  monitoring.script.registerNewServer.smartRule.MS SQL Server=PS\_MSSQL  FunctionalAccount.WINDOWS=pam\_fa\_win-P  FunctionalAccount.LINUX=pam\_fa\_linux-P  FunctionalAccount.AIX=pam\_fa\_aix-P  FunctionalAccount.ORACLE=pam\_fa\_oracle  FunctionalAccount.MS SQL=pam\_fa\_win-P  DefaultAccountsDB.WINDOWS=OSTEAM  DefaultAccountsDB.LINUX=root  DefaultAccountsDB.AIX=root  DefaultAccountsDB.ORACLE=sys  DefaultAccountsDB.MS SQL=SA  PasswordRule.WINDOWS=PAM\_WinPassword\_Policy  PasswordRule.LINUX=PAM\_LinuxPassword\_Policy  PasswordRule.AIX=PAM\_AIXPassword\_Policy  PasswordRule.ORACLE=PAM\_OraclePassword\_Policy  PasswordRule.MS SQL=PAM\_WinPassword\_Policy  DSSKeyRule.LINUX=PAM\_LinuxDSS\_Policy  DefaultAccounts.Windows.accounts=OSTEAM  DefaultAccounts.Linux.accounts=root  DefaultAccounts.AIX.accounts=root  DefaultAccounts.Windows.smartrule=PS\_Win\_MA  DefaultAccounts.Linux.smartrule=PS\_Linux\_MA  DefaultAccounts.AIX.smartrule=PS\_Unix\_MA  onboarding.platformGroups=PS\_Windows,PS\_Linux,PS\_Unix  onboarding.AIX=4  onboarding.Linux=2  onboarding.Windows=1 |

### 3.3.2 Shell file – psrun.sh

The server onboarding details commands are stored in shell script – psrun.sh. The shell script, psrun.sh has various commands that can be run to perform various actions. For the onboarding action, the task name is ‘batch\_reg\_new\_server’.

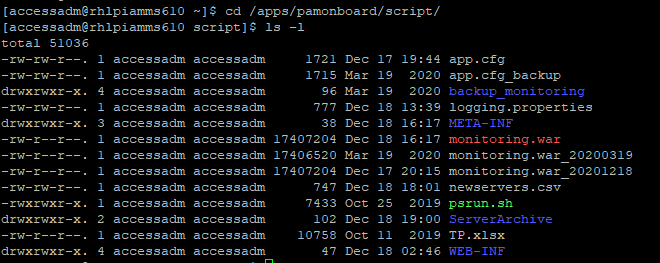


Below is the psrun.sh file (*the task is highlighted*):

|  |
| --- |
| #!/bin/bash  CLASSPATH=/apps/pamonboard/script/WEB-INF/lib/activation-1.1.jar:/apps/pamonboard/script/WEB-INF/lib/commons-compress-1.18.jar:/apps/pamonboard/script/WEB-INF/lib/gson-2.8.5.jar:/apps/pamonboard/script/WEB-INF/lib/poi-ooxml-4.1.0.jar:/apps/pamonboard/script/WEB-INF/lib/automation-1.0-SNAPSHOT.jar:/apps/pamonboard/script/WEB-INF/lib/commons-csv-1.7.jar:/apps/pamonboard/script/WEB-INF/lib/javax.mail-1.6.2.jar:/apps/pamonboard/script/WEB-INF/lib/poi-ooxml-schemas-4.1.0.jar:/apps/pamonboard/script/WEB-INF/lib/commons-codec-1.12.jar:/apps/pamonboard/script/WEB-INF/lib/commons-math3-3.6.1.jar:/apps/pamonboard/script/WEB-INF/lib/jsch-0.1.55.jar:/apps/pamonboard/script/WEB-INF/lib/xmlbeans-3.1.0.jar:/apps/pamonboard/script/WEB-INF/lib/commons-collections4-4.3.jar:/apps/pamonboard/script/WEB-INF/lib/curvesapi-1.06.jar:/apps/pamonboard/script/WEB-INF/lib/poi-4.1.0.jar  JAVA\_HOME=/apps/pamonboard/jdk1.8.0\_221  if [ $1 = 'test' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.Test $2  elif [ $1 == 'enc' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.kpmg.rccl.pam.EncryptionUtils $2  elif [ $1 == 'platform\_service\_vault' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskPlatformServiceAccountVault $2  elif [ $1 == 'platform\_service\_vault\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskPlatformServiceAccountVaultCleanup $2  elif [ $1 == 'platform\_shared\_vault' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskPlatformSharedAccountVault $2  elif [ $1 == 'platform\_shared\_vault\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskPlatformSharedAccountVaultCleanup $2  elif [ $1 == 'app\_service\_vault' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppServiceAccountVault $2  elif [ $1 == 'app\_service\_vault\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppServiceAccountVaultCleanup $2  elif [ $1 == 'app\_shared\_vault' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppSharedAccountVault $2  elif [ $1 == 'app\_shared\_vault\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppSharedAccountVaultCleanup $2  elif [ $1 == 'app\_vendor\_vault' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppVendorAccountVault $2  elif [ $1 == 'app\_vendor\_vault\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskAppVendorAccountVaultCleanup $2  elif [ $1 == 'tag\_windows' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskTagWindows $2  elif [ $1 == 'tag\_linux' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskTagLinux $2  elif [ $1 == 'tag\_aix' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskTagAIX $2  elif [ $1 == 'append\_attribute\_for\_assets' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.UtilAppendAttribute $2 $3 $4  elif [ $1 == 'set\_attribute\_for\_assets' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.UtilSetAttribute $2 $3 $4  elif [ $1 == 'register\_platform' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreatePlatformAttribute  elif [ $1 == 'register\_platform\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreatePlatformAttributeCleanup  elif [ $1 == 'register\_acct\_type' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreateAccountTypeAttribute  elif [ $1 == 'register\_acct\_type\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreateAccountTypeAttributeCleanup  elif [ $1 == 'register\_application' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreateApplicationAttribute  elif [ $1 == 'register\_application\_cleanup' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskCreateApplicationAttributeCleanup  elif [ $1 == 'batch\_reg\_new\_server' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskBatchNewServerRegistration $2  elif [ $1 == 'manage\_msh' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TaskOnboardMSH $2 $3  elif [ $1 == 'temp\_platform\_named' ]  then  $JAVA\_HOME/bin/java -cp $CLASSPATH -Djava.util.logging.config.file=/apps/pamonboard/script/logging.properties -DCFG\_FILE=/apps/pamonboard/script/app.cfg com.rccl.pam.onboarding.TempPlatformNamed $2  fi |

### 3.3.3 Input file - newservers.csv

The csv file is the input file that stores the asset information. This file gets populated when the infrastructure team registers the asset by clicking on a provided webservice link and the information is picked up by the monitoring service that runs in the backend. The file lists the platform, IP, hostname, onboarded flag and scanned database. The assets will be appended to the file as they are registered. To prevent the same file being run multiple times, rename the file and move it to the ‘ServerArchive’ folder.



\*Note – The *ServerArchive* folder contains older .csv files that have assets that have been onboarded.

These fields are required headers in the file : **platform,ip,name,onboarded,database\_scaned**

**For eg**. Windows, 10.118.20.18, winsqlscl02.na.rccl.com, ,

Before onboarding, the onboarded flag will be blank, but after the asset is onboarded through the utility, the onboarded flag is populated with ‘Yes’.

Below is a sample newservers.csv file.

|  |
| --- |
| platform,ip,name,onboarded,database\_scaned  Windows,10.118.20.18,winsqlscl02.na.rccl.com,,  Windows,10.118.20.19,WINSQLsCL03.na.rccl.com,Yes,  Windows,10.118.20.20,stgqpasa02.na.rccl.com,Yes, |

## Utility execution

The onboarding utility automates the onboarding of an asset into BeyondInsight and subsequently manages the asset in Password Safe along adding and managing the default platform account.

The utility is executed by running the command below from the */apps/pamonboard/script* folder :

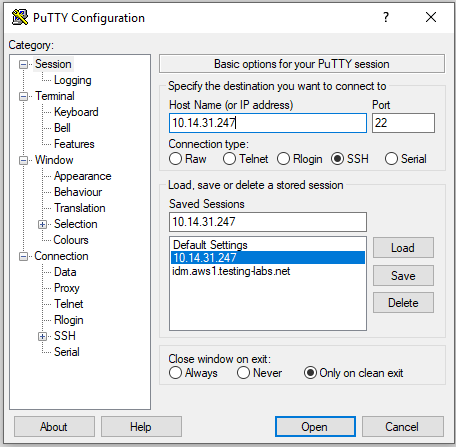
$ ./psrun.sh batch\_reg\_new\_server /apps/pamonboard/script/newservers.csv

To execute the script :

1. Login to the jumpserver using an SSH client like PuTTY:

Hostname: **rhldajumpmes101.na.rccl.com** or 10.14.31.247

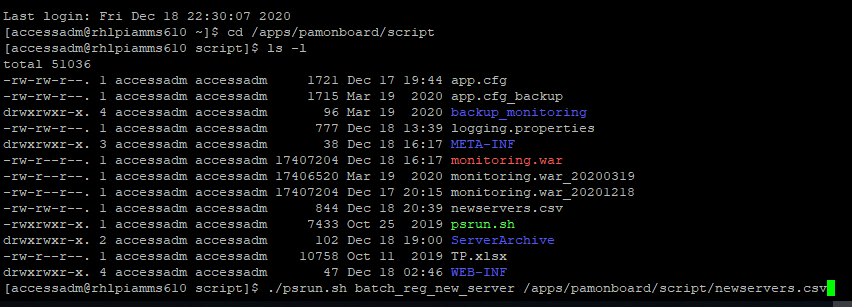
Login as: use NA domain userid and password eg. 700xxxx



1. Switch user to accessadm account by execute ‘sudo su – accessadm’
2. SSH to the monitoring server by executing ‘ssh rhlpiamms610’. Type *yes* on the prompt ‘Are you sure you want to continue connecting (yes/no)?’
3. At the prompt, change to the ‘/apps/pamonboard/script’ path by executing ‘cd /apps/pamonboard/script’
4. List the files in the folder for references by executing ‘ls-l’



1. Run the script by executing ‘./psrun.sh batch\_reg\_new\_server /apps/pamonboard/script/newservers.csv’



1. Logout of all sessions by executing ‘exit’.

Upon execution, the utility process is described as below:

1. The api user (api\_pam) signs in through an API call.
2. The asset is added into BeyondInsight by passing the asset name, IP and platform.
3. The asset is tagged based on the platform ie. Windows, Linux, AIX
4. The asset smart rule is processed (based on the platform) to add the asset into PasswordSafe. (This also involves moving the asset into the correct asset smart group).
5. The default platform account is added with a dummy password.
6. The managed account smart rule is processed to add the default account along with the policies in the smart rule ie. password policy, auto management flag, rotation policy, etc. (This also involves moving the managed account into the correct managed account smart group).
7. The api user signs out.

At this time, only Prod assets that are in the na.rccl.com domain is managed in Prod through the script. For non prod assets, the same process is executed in Stage environment.

## Logging/Audit

When the utility runs, the status of the request is displayed on the console. Upon completion of the utility, it will return to the command prompt.

