National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme

TM

Validation Report

for

BeyondTrust PowerBroker
UNIX® + Linux® Edition V9.1

Report Number: CCEVS-VR-VID10691-2016
Dated: August 30, 2016
Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6940
Fort George G. Meade, MD 20755-6940

Table of Contents

List of Tables

List of Figures

1 Executive Summary

This report is intended to assist the end-user of this product and any security certification agent for that end-user in determining the suitability of this Information Technology (IT) product in their environment. End-users should review the Security Target (ST), which is where specific security claims are made, in conjunction with this Validation Report (VR), which describes how those security claims were evaluated and tested and any restrictions on the evaluated configuration. Prospective users should carefully read the Assumptions and Clarification of Scope in Section 4 and the Validator Comments in Section 10, where any restrictions on the evaluated configuration are highlighted.

This report documents the National Information Assurance Partnership (NIAP) assessment of the evaluation of BeyondTrust PowerBroker UNIX + Linux Edition V9.1. It presents the evaluation results, their justifications, and the conformance results. This VR is not an endorsement of the Target of Evaluation (TOE) by any agency of the U.S. Government and no warranty of the TOE is either expressed or implied. This VR applies only to the specific version and configuration of the product as evaluated and as documented in the ST.

The evaluation of the BeyondTrust PowerBroker UNIX + Linux Edition V9.1was performed by Leidos Common Criteria Testing Laboratory (CCTL) in Columbia, Maryland, in the United States and was completed in August 2016. The evaluation was conducted in accordance with the requirements of the Common Criteria and Common Methodology for IT Security Evaluation (CEM), version 3.1, revision 4 and assurance activities specified in the following Protection Profiles:

- Standard Protection Profile for Enterprise Security Management Access Control, Version 2.1, 24 October 2013 with no additional optional SFRs.
- Standard Protection Profile for Enterprise Security Management Policy Management, Version 2.1, 24 October 2013 and includes the additional optional SFRs: FAU_SEL.1, and FMT_MTD.1.

The evaluation was consistent with NIAP Common Criteria Evaluation and Validation Scheme (CCEVS) policies and practices as described on their web site (www.niap-ccevs.org).

The Leidos evaluation team determined that the BeyondTrust PowerBroker UNIX + Linux Edition V9.1 is conformant to the claimed Protection Profiles (PPs) and, when installed, configured and operated as specified in the evaluated guidance documentation, satisfies all of the security functional requirements stated in the ST. The information in this VR is largely derived from the Assurance Activities Report (AAR) and the associated test report produced by the Leidos evaluation team.

The Target of Evaluation (TOE) is a software solution that consists of the BeyondTrust PowerBroker ® UNIX® + Linux® Edition V9.1 (PBUL). PBUL is a security management product that provides the capability to delegate access to operating system functions available to specific privileged accounts (e.g., 'root') and offer those functions in a controlled and granular fashion to other specific and suitably trusted users. The TOE provides both Enterprise Security Policy Management and Access Control functions.

The network on which it resides is considered part of the operational environment.

The validation team monitored the activities of the evaluation team, examined evaluation evidence, provided guidance on technical issues and evaluation processes, and reviewed the evaluation results produced by the evaluation team. The validation team found that the evaluation results showed that all assurance activities specified in the claimed PPs had been completed successfully and that the product satisfies all of the security functional and assurance requirements stated in the ST. Therefore the validation team concludes that the testing laboratory's findings are accurate, the conclusions justified, and the conformance results are correct. The conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence produced.

Table 1: Evaluation Details

Item	Identifier
Evaluated Product	BeyondTrust PowerBroker UNIX + Linux Edition V9.1
Sponsor & Developer	BeyondTrust Software, Inc. 5090 N. 40th Street Phoenix, AZ 85018 United States
CCTL	Leidos Common Criteria Testing Laboratory 6841 Benjamin Franklin Drive Columbia, MD 21046
Completion Date	August 30, 2016
CC	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012
Interpretations	There were no applicable interpretations used for this evaluation.
CEM	Common Methodology for Information Technology Security Evaluation: Version 3.1, Revision 4, September 2012
Protection Profiles	• [PP_ESM_AC] Standard Protection Profile for Enterprise Security Management Access Control, Version 2.1, 24 October 2013 with no additional optional SFRs. • [PP_ESM_PM] Standard Protection Profile for Enterprise Security Management Policy Management, Version 2.1, 24 October 2013 and includes the additional optional SFRs: FAU_SEL.1, and FMT_MTD.1.
Disclaimer	The information contained in this Validation Report is not an endorsement either expressed or implied of the BeyondTrust PowerBroker UNIX + Linux Edition V9.1
Evaluation Personnel	Greg Beaver Cody Cummins
Validation Personnel	Marybeth Panock Daniel Faigin Jean Petty

2 Identification

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation.

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Product Compliant List (PCL).

The following table identifies the evaluated Security Target and TOE.

Table 2: ST and TOE Identification

Name	Description
ST Title	BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1 Security Target
ST Version	1.0
Publication Date	August 3, 2016
Vendor	BeyondTrust Software, Inc.
ST Author	Leidos
TOE Reference	BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1
TOE Software Version	BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1
Keywords	Linux, UNIX, Enterprise Security Management

2.1 Threats

The security target includes by reference the Security Problem Definitions (composed of organizational policies, threat statements, and assumptions) from the ESM PPs.

In general, the ESM PPs have presented Security Problem Definitions appropriate for Enterprise Security Management Access Control and Policy Management products, and as such are applicable to the BeyondTrust TOE.

The ESM PPs identify the following threats that the TOE and its operational environment are intended to counter:

- A malicious user or careless user may suspend or terminate the TOE's operation, thus making it unable to enforce its access controls upon the environment or TOE-protected data.
- A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.
- A malicious user can falsify the TOE's identity, giving the Policy Management product false assurance that the TOE is enforcing a policy.
- A malicious user may create a false policy and send it to the TOE to consume, adversely altering its behavior.
- A malicious user may attempt to mask their actions, causing audit data to be incorrectly recorded or never recorded.

- A malicious or careless user may cause the TOE to lose connection to the source of its enforcement policies, adversely affecting access control behaviors.
- A malicious user may attempt to provide incorrect policy data to the TOE in order to alter its access control policy enforcement behavior.
- A malicious or careless user may access an object in the Operational Environment that causes disclosure of sensitive data or adversely affects the behavior of a system.
- An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
- A careless administrator may create a policy that contains contradictory rules for access control enforcement resulting in a security policy that does not have unambiguous enforcement rules.
- A malicious user may exploit a weak or nonexistent ability for the TOE to provide proof of its own identity in order to send forged policies to an Access Control product.
- A malicious user could bypass the TOE's identification, authentication, and authorization mechanisms in order to use the TOE's management functions.
- A malicious user could be illicitly authenticated by the TSF through brute-force guessing of authentication credentials.
- A Policy Administrator may be incapable of using the TOE to define policies in sufficient detail to facilitate access control, causing an Access Control product to behave in a manner that allows illegitimate activity or prohibits legitimate activity.

2.2 Organizational Security Policies

The ST references the ESM PPs to identify following organizational security policy that the TOE and its operational environment are intended to fulfill:

- The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
- The organization will exercise due diligence to ensure that the TOE is updated with relevant policy data.

3 Architectural Information

PowerBroker is a software-only product suite that runs on numerous UNIX and Linux operating systems without modifying the kernel. The purpose of the product is to act as the "broker" between the user and the privileged operations on the system. To achieve this, the PowerBroker security policy is consulted each time the user attempts to run a privileged command through PowerBroker. The product provides two mechanisms through which this can be accomplished: the *pbrun* command and the PB Shells.

The *pbrun* command is used in a standard UNIX shell just like any other command. A user wishing to execute a privileged command invokes the desired privileged command through *pbrun*. For example, if the command *mount* is a privileged command delegated by PowerBroker, a user wishing to run *mount* would execute the command '`pbrun mount <mount options>`' from the regular shell. *PBRun* sends the secured task request to a policy server for processing. The TOE determines whether or not the user has permission to execute the *mount* command on the target host. If permission is granted, the command is executed on behalf of the user. Privileged commands requested by a user and authorized and executed by PowerBroker are known as 'secured tasks'.

The PB Shells are customized versions of the public domain pdksh '88 Korn shell (*pbksh*) and Bourne shell (*pbsh*). These modified shells contain the full functionality and features of the standard public domain shells, but they have been modified to verify all command operations through PowerBroker before allowing execution. Any user running *pbsh* or *pbksh* as the shell will be under the control of the PowerBroker access control mechanisms.

All attempted actions mediated by PowerBroker are logged in a detailed audit log. The administrator has control over whether or not the keystrokes and output of a particular action are audited. Security audit data is stored in two types of logs:

- Event Log—this PowerBroker audit file records when each requested task was accepted or rejected. For tasks not run in local mode, it also logs when the task terminated, and any configured keystroke-monitoring events that were triggered by that task attempt. These events are known as ACCEPT, REJECT, FINISH, and KEYSTROKE events. The Event Log is a binary file that can be encrypted, but is not encrypted by default
- IO Logs—optional logs that record I/O (i.e., keystrokes and output) information for specific secured tasks. Auditing of this type of data is not within the scope of the evaluation.
- Configuration Database—this database is a version controlled database that stores key configuration, settings and policy files, including auditing of activities such as the creation of new files and version changes within controlled files.

A typical PowerBroker configuration consists of the following primary components:

- *pbrun* (or *pbsh, pbksh*)—requests that a secured task is run in a controlled environment
- *pbmasterd*—receives secured task requests from *pbrun*, *pbksh*, and *pbsh* and evaluates them according to the current security policies. If the request is accepted, it directs *pblocald* to run the secured task
- *pblocald*—the daemon that runs secured tasks on behalf of the user, when instructed to do so by the master daemon (*pbmasterd*)
- *pblogd*—the log server daemon records event logs and I/O logs as directed by other PB programs.

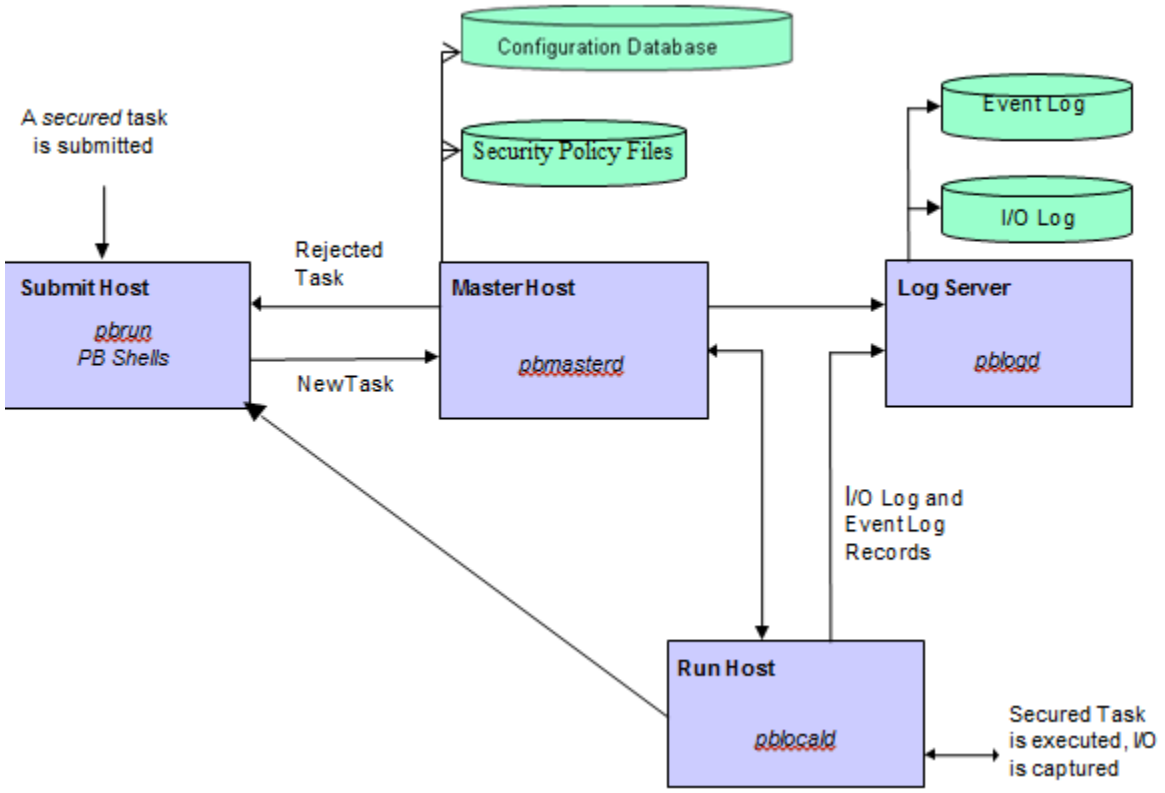

Figure 1: PowerBroker Component Interactions

Figure 1 depicts the interactions between the primary TOE components. Each blue box represents a logical operating environment (e.g., 'Submit Host') for the listed TOE components (identified by italics, e.g., '*pbrun*'). The policy files used by the TOE and the logs generated by the TOE are stored in files in the operational environment (the green 'disk drives').

The machine from which a task is submitted is referred to as the Submit Host. The machine on which Security Policy File processing takes place is referred to as the Master Host. The machine on which a task is actually executed is referred to as the Run Host. The machine on which Event Log records and I/O log records are written is referred to as the Log Server (or Host). It is possible to install any or all of these components on a single machine, or to distribute them between different machines. Use of a separate log server and *pblogd* daemon is optional, but highly recommended. When *pblogd* is not used, *pbmasterd* logs the audit records. For optimal security, the master hosts and log servers should be separate machines that are isolated from normal user activity. When the TOE components are deployed on separate machines, the TOE must be configured to encrypt communications between the separate components. The TOE uses TLS and FIPS validated algorithms provided by OpenSSL in the operational environment.

The typical sequence of PowerBroker processing is as follows:

- A user (or administrator) establishes a session with the UNIX/Linux machine running the Submit Host
- The user is authenticated by an authentication server in the operational environment
- From a normal shell on the Submit Host, a user submits a request via *pbrun*
- *pbmasterd* on the Master Host processes the security policy and either accepts or rejects the request

- The request acceptance or rejection is audited and an event sent to the Log Server. For rejected requests, processing ends here
- An accepted request is executed via *pblocald* on the Run Host
- If I/O logging was designated by the security policy, this data is sent to the Log Server.

Common variations to this processing sequence are as follows:

- If the Submit Host is the same server as the Run Host, "local mode" or "Optimized Run Mode" can be enabled. In these cases, if *pbmasterd* accepts the command, it is executed from the *pbrun* process rather than launching *pblocald*
- If there is no separate Log Server, *pbmasterd* performs the logging services
- *pbmasterd*, *pblocald* and *pblogd* can be configured to run continuously as daemons, or alternately can be configured to launch on a per-use basis by *inetd* or equivalent (e.g., *xinetd*, *SMF*, *launchd*).

The *pbmasterd*, *pblocald*, and *pblogd* components all run as 'root' (or equivalent, depending on the operational environment). The *pbrun*, *pbsh*, and *pbksh* components run as the invoking user but with setuid root.

As indicated above, all TOE components can be installed on a single machine, or can be deployed across a number of machines. Any machine that is to be used as a Submit Host requires *pbrun*, *pbsh*, or *pbksh* to be installed on it. Each Submit Host will have (in its configuration file) a list of one or more Master Hosts. Each Master Host requires *pbmasterd* to be installed on it to process secured task requests. Any machine that will be used as a Run Host requires *pblocald* to be installed on it. Use of a Log Host is optional—in the absence of a Log Host, *pbmasterd* is responsible for logging activities. Any machine that will be used as a Log Host requires *pblogd* to be installed on it.

In summary, in the TOE model, an access request originates at a network host (Submit Host) and is transmitted to the central Policy Manager (Master Host), which also acts as the policy decision point. If the Policy Manager determines the access control request complies with the defined policy, it forwards the access request (secured task) to the target host (Run Host) for action. The Run Host is part of the Access Control portion of the TOE that performs the requested operation and communicates the results back to the Submit Host. If the access request does not conform to policy, it is rejected and the originator (Submit Host) is notified. As such, the TOE model inverts the ESM model for PM and AC presented in the PP in that the ESM model assumes a central point where access control policies are created and managed and then distributed as appropriate to other computers on the network where the policy is enforced.

Other PowerBroker components comprise:

- PB Shells (*pbsh* and *pbksh*)—as indicated above, the PB shells function similarly to *pbrun* in Figure 1. The PB shells obtain approval from *pbmasterd* for every command issued at the PB shell prompt. The PB shells provide transparent authorization and event logging for every command, shell built-in, and shell I/O redirection, and control of shell scripts. Once accepted by *pbmasterd*, the commands are executed by either *pblocald* or by the PB shell
- PB GUIs (*pbguid* and *pbsguid*)—the PB GUI programs provide an HTTP (*pbguid*) and an HTTPS (*pbsguid*) server for browser-based administration of PowerBroker. The administrator accesses the GUI by starting a browser (in the operational environment) on their local machine and connecting to a host and port where *pbguid* or *pbsguid* is hosted. Administrators using the GUI are authenticated by the underlying OS on which the GUI programs are installed. The GUI allows an authorized user to change settings files, view events and keystroke logs, edit policy configuration files, run reports, and update the GUI configuration. *Pbguid* provides unencrypted GUI access, and *pbsguid* provides TLS-protected GUI access. Administrators must use HTTPS; HTTP is not permitted in the evaluated configuration.

- PB Administrative Utilities—used to administer PowerBroker. They provide the following capabilities:
    - *pbbench*—a diagnostic tool that helps solve configuration, file permission and network problems. It reads the information in the PowerBroker settings file on the local machine and uses system information to verify the information in the settings file
    - *pbcall*—allows a PowerBroker policy language function to be executed from the command line, allowing the administrator to test the effects of that function on the local machine
    - *pbcheck*—used to check the PowerBroker configuration file for errors
    - *pbencode*—encrypts a file using a key specified in the command line or in the settings file
    - *pbhostid*—used to display a computer's unique, hardware-dependent identifier, which is subsequently used in generating a product license string. This utility is used only during TOE installation
    - *pbkey*—used to generate symmetric encryption keys for protecting files and network traffic
    - *pblicense*—displays current licensing information and retires licenses
    - *pblog*—used to display entries from a PowerBroker event log (the PB GUIs also provide this capability)
    - *pbpasswd*—generates an encrypted password that can then be used in the policy file to provide password protection to secured tasks
    - *pbprint*—produces a formatted display of a PowerBroker policy file
    - *pbreplay*—used to display the contents of a PowerBroker keystroke log (the PB GUIs also provide this capability)
    - *pbreport*—used to extract data from PowerBroker event logs and generate reports (the PB GUIs also provide this capability)
    - *pbsum*—prints the checksum of one or more files, which can then be used in the policy file to check the requested program's integrity
    - *pbsync*—starts the log synchronization process
    - *pbsyncd*—a server that listens for log synchronization requests from one or more clients
    - *pbuvqrpg*—works with *pbreport* to generate text based reports.

  The *pbbench*, *pbcall*, *pbencode*, and *pbsum* utilities can be run on any host (i.e., Submit, Master, Log or Run). The *pbcheck*, *pbhostid*, *pbkey*, *pblicense*, *pbpasswd*, and *pbprint* utilities can be run only on the Master host, while the *pblog*, *pbreplay*, *pbsync*, *pbsyncd*, *pbreport*, and *pbuvqrpg* utilities can be run on the Master and Log hosts.

- PB User Utilities (*pbvi*, *pbnvi*, *pbless*, *pbumacs*, and *pbmg*)—the PB User Utilities are similar to standard UNIX *vi*, *emacs*, and *less* commands, except that they are 'hardened' such that they do not include the standard functions to allow access to other files, run other commands from inside the utility, or to access sub-shells from which other commands could be run. Note that these utilities would be run on the Run Host under the control of the rest of the TOE.
- PB REST API—the PB REST API developed for PowerBroker Servers UNIX/Linux to allow other software to configure, customize and retrieve data from PBUL. The API is web based and uses industry standard modern components, connectors and data elements within a distributed and secure enterprise environment. **The REST API is not included in the evaluated configuration and should not be enabled.**

4 Assumptions

The ST references the [PP_ESM_AC] and [PP_ESM_PM] to identify following assumptions about the use of the product:

- The TOE will use cryptographic primitives provided by the Operational Environment to perform cryptographic services.
- The TOE will be able to establish connectivity to other ESM products in order to share security data.
- The TOE will receive policy data from the Operational Environment.
- The Operational Environment will provide mechanisms to the TOE that reduce the ability for an attacker to impersonate a legitimate user during authentication.
- The TOE will receive reliable time data from the Operational Environment.
- The TOE will receive identity data from the Operational Environment.
- There will be a competent and trusted administrator who will follow the guidance provided in order to install the TOE.

4.1 Clarification of Scope

All evaluations (and all products) have limitations, as well as potential misconceptions that need clarifying. This text covers some of the more important limitations and clarifications of this evaluation. Note that:

1. As with any evaluation, this evaluation only shows that the evaluated configuration meets the security claims made, with a certain level of assurance (the assurance activities specified in the claimed PPs and performed by the evaluation team).
2. This evaluation covers only the specific device models and software version identified in this document, and not any earlier or later versions released or in process.
3. The evaluation of security functionality of the product was limited to the functionality specified in the claimed PPs. Any additional security related functional capabilities of the product were not covered by this evaluation.
4. This evaluation did not specifically search for, nor attempt to exploit, vulnerabilities that were not "obvious" or vulnerabilities to objectives not claimed in the ST. The CEM defines an "obvious" vulnerability as one that is easily exploited with a minimum of understanding of the TOE, technical sophistication and resources.
5. The following specific product capabilities are excluded from use in the evaluated configuration:
    a. Non-FIPS 140-2 mode of operation—this mode of operation allows cryptographic operations that are not FIPS-approved
6. The TOE can be configured to rely on and utilize a number of other components in its operational environment:
    a. Use of the external authentication methods in the operational environment require Lightweight Directory Access Protocol (LDAP) (e.g., Active Directory (AD)), and Remote Authentication Dial-In User Service (RADIUS) servers.
    b. The TOE relies on 3rd party FIPS capable OpenSSL 1.0.2a in conjunction with the TOEs FIPS mode (that disables non FIPS algorithms).

The security policy against which secured tasks are assessed is specified using the PBUL security policy scripting language. This provides the security administrator a flexible tool for specifying the security policy to be enforced by PBUL when mediating requests to run privileged commands submitted by users in the enterprise. The statements in the policy file are interpreted by PBUL to determine if the submitted request ('secured task') is accepted or rejected. The evaluation of PBUL did not cover all of the statements and functions within the scope of the policy language, but rather the ability of PBUL to satisfy the requirements specified in the ESM PM PP for defining access control policies. As such, testing of PBUL during the course of the evaluation covered the following aspects of the policy language: accept, reject and conditional (if-else) statements; variables, including strings and lists of strings; relational, logical and string operators; predefined variables; and some built-in functions (such as ldap_bind and printf). The policy language is fully defined in the PBUL Policy Language Guide. Any aspects of the policy language not specifically identified above were not covered in the course of the evaluation and no conclusions about their correctness or efficacy should be drawn from the result of the evaluation.

5 Security Policy

The TOE enforces the following security policies as described in the ST.

Note: Much of the description of the security policy has been derived from the ST and the Final ETR.

5.1 Enterprise Security Management

The TOE provides the ability to define access control policies for consumption by a compatible Access Control product: i.e., the TOE itself. Access control policies consist of subject, object, and attributes; policies are uniquely identified. The TOE ensures that policies are available to the TOE's Access Control component immediately following creation of a new or updated policy.

The TOE relies on LINUX/UNIX host, LDAP, RADIUS, and optionally Pluggable Authentication Module (PAM) in the operational environment for subject identification and authentication; and requires each subject to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that subject.

5.2 Security Audit

The TOE is designed to be able to generate logs for security relevant events including the events specified in ESM PPs. The TOE can be configured to store the logs locally. The audit records identify the date/time, event type, outcome of the event, responsible subject/user, as well as the additional event-specific content indicated in the Security Target Table 2.

Selective audit capability is exercised by the Policy Management portion of the TOE that configures the access-control related auditing functions by Administrator defined policy variables and by event type.

The TOE transmits audit records to TOE internal storage and uses TLS for distributed communications. The TOE protects the stored audit records in the TOE-internal audit trail from unauthorized deletion and modification. The cryptographic algorithms used in TLS are provided by the OpenSSL FIPS validated modules in the operational environment.

5.3 Communication

The TOE is both a Policy Management and Access Control product where policies are centralized and never transmitted. Policies are defined on a Master Host and available immediately as soon as it is saved. The policy files never leave this location or otherwise traverse across the TOE or outside the TOE. The administrator can verify the existence of the policy by performing a policy lookup using the policy file name; and can verify the location (Master Host) of the policy by viewing the Master Host field/attribute.

5.4 User Data Protection

The TOE controls access to commands that have been defined to be controlled on target hosts. The TOE's self-protection Security Function Policy restricts access to objects that reside in the Operational Environment that impact the TOE's behavior.

5.5 Identification and Authentication

The TOE associates the *uid* and *gid* user security attributes with subjects acting on the behalf of a user. The TOE uses an external LDAP or RADIUS server to authenticate users and enforces the result. The TOE determines the *uid* from the credentials presented at authentication and associates the *gid* retrieved from the authentication server with the corresponding *uid*.

5.6 Security Management

The TOE provides administrative functions available from a command line interface (CLI) and a graphical user interface (GUI) to access the management functions and for administrators to change their own passwords. Security management commands are limited to authenticated users with root access. The TOE provides the *AdminUser* role which provides root access.

The TOE also provides the ability for the Policy Management components to manage the Access Control components of the TOE. The TOE components must be configured to communicate with one another using TLS or HTTPS and as such can trust one another. The default values for security attributes used in the access control policies are restrictive and the Policy Management component can change these defaults. The TOE's policy management engine defines an unambiguous hierarchical method of implementing a policy such that no contradictions occur.

5.7 Protection of the TSF

The TOE uses external Identity and Credential Management products to define its administrator authentication data; the TOE does not store or cache the data. The TOE does not offer any functions that will disclose to any users a stored cryptographic key; and all keys are stored encrypted using AES-256.

Should the TOE or a TOE component encounter a failure state, all access control requests are denied. The TOE is both an Access Control and Policy Management product. If the TOE is in a failed state then no access control requests or decisions can be made. Policies are defined in a central location and are never transmitted. The TOE prevents replay attacks for secured tasks and rejects the secured task when replay is detected. The TOE relies on the implementation of TLS in the operational environment to provide secure transmission, including replay detection, of secured tasks.

5.8 Resource Utilization

The TOE is both an Access Control and Policy Management product. The most recent policy will always be enforced even in the event of a TOE failure. Should the TOE experience a failure, no access control is permitted until the system comes back up. Policies are defined and enforced on the same component and therefore it is not possible to lose communication during a policy transmission.

5.9 Trusted Path/Channels

The TOE protects interactive communication with remote administrators using HTTP over TLS. TLS ensures both integrity and disclosure protection.

The TOE protects communication with external LDAP servers and internal distributed TOE components using TLS connections to prevent unintended disclosure or modification of the transferred data.

The TOE uses FIPS capable OpenSSL v1.0.2a and requires FIPS mode to disable non FIPS algorithms. Customers are instructed to choose their own validated FIPS Object Module and link that with the FIPS capable OpenSSL v1.0.2a that is provided. The validated Object Module and FIPS capable OpenSSL are in the operational environment.

6 Documentation

BeyondTrust Software, Inc. offers a series of documents that describe the installation process for the TOE, as well as guidance for subsequent use and administration of the system security features.

- PowerBroker for UNIX & Linux Common Criteria Supplementary Guide
- PowerBroker Servers UNIX + LINUX Edition Browser Interface Guide, Version 9.1, July 2015
- PowerBroker Servers UNIX + LINUX Edition System Administrators Guide, Version 9.1, July 2015
- PowerBroker Servers UNIX + LINUX Edition Installation Guide, Version 9.1, July 2015
- PowerBroker Servers UNIX + LINUX Edition Policy Language Guide, Version 9.1, July 2015

Supporting TOE Guidance Documentation

- BeyondTrust PowerBroker UNIX® + Linux® Edition Security Target, Version 1.0, August 3, 2016

7 Independent Testing

This section describes the testing efforts of the evaluation team. It is derived from information contained in the following:

- BeyondTrust PowerBroker UNIX + Linux Edition V9.1 Common Criteria Test Report and Procedures, Version 1.6, August 25, 2016

The purpose of this activity was to confirm the TOE behaves in accordance with the TOE security functional requirements as specified in the ST for a product claiming conformance to the [PP_ESM_AC] Standard Protection Profile for Enterprise Security Management Access Control, Version 2.1, 24 October 2013 with no additional optional SFRs and [PP_ESM_PM] Standard Protection Profile for Enterprise Security Management Policy Management, Version 2.1, 24 October 2013 and includes the additional optional SFRs: FAU_SEL.1, and FMT_MTD.1.

The evaluation team devised a Test Plan based on the Testing Assurance Activities specified in the [PP_ESM_AC] and [PP_ESM_PM] Protection Profiles. The Test Plan described how each test activity was to be instantiated within the TOE test environment. The evaluation team executed the tests specified in the Test Plan and documented the results in the team test report listed above.

Independent testing took place at the Leidos facility in Columbia, Maryland and on vendor onsite from February 10, 2015 – August 25, 2016.

The evaluators received the TOE in the form that normal customers would receive it, installed and configured the TOE in accordance with the provided guidance, and exercised the Team Test Plan on equipment configured in the testing laboratory.

7.1 Evaluated Configuration

To simply the testing process the evaluator used two different configurations. In one configuration all the PowerBroker components were enabled on a single machine. In the second configuration the different PowerBroker components were enabled on separate machines. All platforms were tested in both configurations.

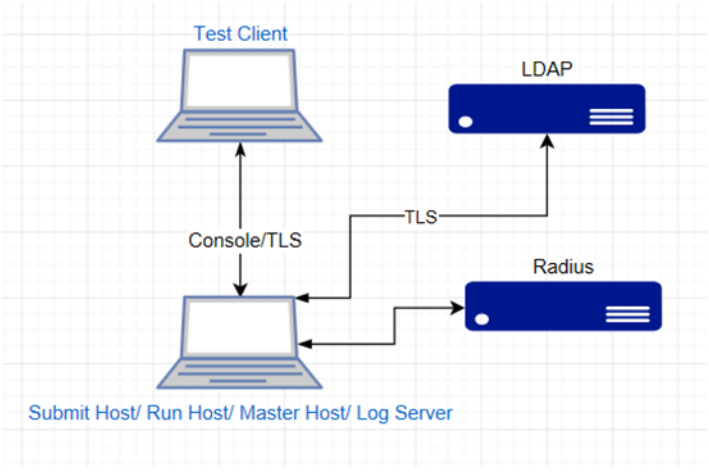

Figure 2 - Single Server Configuration

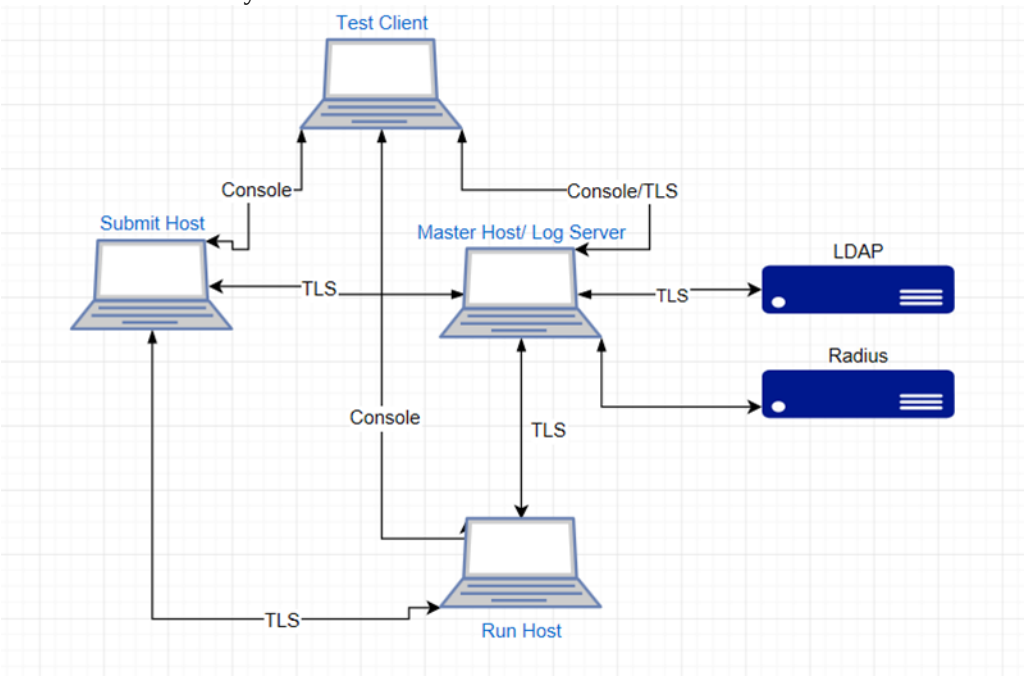

Figure 3 - Multi-Server Configuration

As documented in the diagram above, the following hardware and software components were included in the evaluated configuration during testing:

- TOE Components
  - PowerBroker UNIX + Linux v 9.1 running on the following platforms;
    - HP-UX 11i v3
    - Solaris 11
    - AIX v6.1
    - Ubuntu 14.4
- Non-TOE Components
  - Test Client Used for Administration
  - Server 2008 running Active Directory for LDAP communication
  - Radius Server
  - OpenSSL FIPS Object Module SE v2.0.12, CMVP #2398

The Ubuntu operating system was used in the single-server configuration depicted in Figure 2 above. Two separate multi-server configurations (as depicted in Figure 3) were established, one in which each server ran Ubuntu, and a second where HP-UX, Solaris and AIX were used, one per server. In this second multi-server configuration, the roles of Submit Host, Master Host, and Run Host were "rotated" around the three different servers so that testing covered each platform in each role.

To ensure that the TOE would operate correctly in an enterprise environment, the configuration in Figure 3 was modified to include a test that demonstrated that three SubmitHosts sending requests to the one MasterHost operated correctly.

The evaluated version of the TOE was installed and configured according to the PowerBroker for UNIX & Linux Common Criteria Supplementary Guide as well as the supporting guidance documentation identified in Section 6.

Given the complete set of test results from the test procedures exercised by the evaluators, the testing requirements for the [PP_ESM_AC] and [PP_ESM_PM] Protection Profiles are fulfilled.

7.2 Penetration Testing

The evaluation team conducted an open source search for vulnerabilities in the product. The open source search did not identify any obvious vulnerability applicable to the TOE in its evaluated configuration.

The evaluation team performed penetration testing in an attempt to create buffer overflows. The testing revealed that a user could not escalated privileges and the TOE was not vulnerable.

8 Results of the Evaluation

The evaluation was conducted based upon the assurance activities specified in [PP_ESM_AC] and [PP_ESM_PM] Protection Profiles; and the additional optional SFRs: FAU_SEL.1, and FMT_MTD.1, in conjunction with version 3.1, revision 4 of the CC and the CEM. A verdict for an assurance component is determined by the resulting verdicts assigned to the corresponding evaluator action elements.

The validation team's assessment of the evidence provided by the evaluation team is that it demonstrates that the evaluation team performed the assurance activities in the claimed PPs, and correctly verified that the product meets the claims in the ST.

The details of the evaluation are recorded in the Evaluation Technical Report (ETR), which is controlled by the Leidos CCTL. The security assurance requirements are listed in the following table.

Table 3 TOE Security Assurance Requirements

Assurance Component ID	Assurance Component Name
ADV_FSP.1	Basic functional specification
AGD_OPE.1	Operational user guidance
AGD_PRE.1	Preparative procedures
ALC_CMC.1	Labeling of the TOE
ALC_CMS.1	TOE CM coverage
ATE_IND.1	Independent testing - conformance
AVA_VAN.1	Vulnerability survey

9 Validator Comments/Recommendations

Every evaluation has limitations that should be noted. The following paragraphs attempt to capture those worth understanding for this validation

Inverted Policy Enforcement
The BeyondTrust PowerBroker UNIX + Linux Edition Version 9.1, as previously indicated, is conformant with the current Protection Profiles for Enterprise Security Management (ESM) Access Control and Policy Management. However, the product and the evaluation inverts the notional ESM model originally defined. The notional model has a central policy definition node distributing policy out to multiple enforcement points across an Enterprise. This product has multiple endpoints sending the decision to be made to a central policy definition/enforcement point, which makes the decision and then sends the response back out to the endpoints. This inversion was presented to the Technical Rapid Response Teams (TRRT) which gave approval with the conditions that the required Security Functional Requirements (SFRs) were met and the testing performed met the intent of these SFRs, though inverted. The testing was successful and did show that the SFRs were appropriately satisfied for this product.

Multiple Submit Hosts
The validators recommended that in order to verify that the PowerBroker product would truly work in an enterprise environment, the testing configuration should include more than one Submit Host sending requests to the Master Host. This was successfully demonstrated with three Submit Hosts sending requests to the one Master Host. The evaluator sent a *pbrun* request from each of the 3 Submit Hosts. Two of the requests were to be accepted by the policy and one was request was to be rejected by the policy. The evidence collected shows that each request was handled and executed successfully.

All Components on All Platforms
In order to ensure that the all of the evaluated platforms, Solaris 11, AIX v6.1, HP-UX 11i v3, and Ubuntu 14.4, could adequately function as each of the components, Submit Host, Master Host, and Run Host, throughout the course of testing an explicit effort was made to ensure that each of the platforms were tested as a Submit Host, the Master Host, and a Run Host.

Equivalency Argument
Red Hat Enterprise Linux was not tested because an acceptable equivalency argument was made that because both Ubuntu and Red Hat use the same TOE Version and install the same image of PowerBroker, testing Ubuntu is sufficient. Both Red Hat and Ubuntu run Linux distributions, and therefore, handle root access in the same manner. However, there are differences in the distributions. To ensure that the evaluated configuration is used, guidance for the Administrator, "PowerBroker for Unix & Linux Common Criteria Guide" specifies that only TOE provided libraries be used.

Limit of testing of the scripting language
The purpose of the evaluation is to show that SFRs specified in the ESM AC PP and ESM PM PP are satisfied by PowerBroker, not to comprehensively test the scripting language implemented by PowerBroker for the administrators use to manage the security policy. The policy language is a mechanism that supports the TOE's ability to satisfy the SFRs. As such, testing of PBUL during

the course of the evaluation covered the language aspects *accept*, *reject*, and conditional (if-else) statements; variables, including strings and lists of strings; relational, logical and string operators; predefined variables; and some built-in functions (such as ldap_bind and printf). In addition, the evaluators conducted an open source search for vulnerabilities of the scripting language and conducted buffer overflow testing to demonstrate that it is safe to use and does not pose an additional security risk.

10 Annexes

Not applicable

11 Security Target

- BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1 Security Target, Version 1.0, August 3, 2016

12 Abbreviations and Acronyms

This section identifies abbreviations and acronyms used in this document.

Abbreviation	Description
AD	Active Directory
API	Application Programming Interface
CC	Common Criteria for Information Technology Security Evaluation
ESM	Enterprise Security Management
ESM AC	Enterprise Security Management Access Control
ESM PM	Enterprise Security Management Policy Management
ESMPPs	The ESM AC and ESM PM Protection Profiles
GID	Also referred to as gid: Group ID or Group Identity
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP(S)	Hypertext Transfer Protocol (Secure)
LDAP	Lightweight Directory Access Protocol
OpenLDAP	A free, open source implementation of the Lightweight Directory Access Protocol (LDAP).
OS	Operating System
PB	PowerBroker
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SMF	Service Management Facility
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
UID	Also referred to as uid: User ID or User Identity

13 Bibliography

The Validation Team used the following documents to produce this Validation Report:

[1] Common Criteria for Information Technology Security Evaluation Part 1: Introduction, Version 3.1, Revision 4, September 2012.

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1 Revision 4, September 2012.

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 4, September 2012.

[5] BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1 Security Target, Version 1.0, August 3, 2016

[6] Common Criteria Evaluation and Validation Scheme - Guidance to CCEVS Approved Common Criteria Testing Laboratories, Version 2.0, 8 Sep 2008.

[7] Evaluation Technical Report For BeyondTrust PowerBroker UNIX® + Linux® Edition V9.1 Part 2 (Leidos Proprietary), Version 1.2, May 12, 2016