powerbroker® servers
UNIX®+ Linux® Edition

beyondtrust®
securing the perimeter within

PowerBroker® Servers Best Practices

General Guidelines

1. Since PowerBroker for Servers is a network security solution, make sure that DNS or any hostname resolution technology allows an IP address to be derived from the hostname, for both forward and reverse lookups.

2. Be sure hostnames are used consistently in your environment. For example, if FQDN hostnames are configured in DNS, use FQDN everywhere.

3. Whenever possible, delegate a non-interactive task as opposed to a task that generates a session (i.e., an editor session or a shell).

4. Should an editor session be delegated, understand the security implications. Some editors allow a user to use the exclamation mark (!) and path to an application to invoke an external application (a.k.a. shell) from within the application.

5. Instead of delegating *vi*, *more*, or *emacs*, delegate *pbvi*, *pbless*, or ***pbumacs***, since these "hardened utilities" do not allow an external process to be executed.

6. Spread the masterhosts across multiple geographic or logical geographies. Sometimes it will make sense to have all submit/run hosts point to a single primary master, and have a secondary master serve as a failover masterhost.

7. The default outgoing port range is 600-1023. On a particular PowerBroker for Servers session, it is possible to have at least two outgoing connections, therefore a maximum of 200 concurrent PowerBroker for Servers sessions can be run. These default ranges can be increased by specifying a larger range of listening and outgoing ports in the settings file.

8. When possible, configure the use of Kerberos and/or SSL in your environment to enhance overall security and minimize any spoofing attacks.

9. Always encrypt network traffic between the PowerBroker for Servers components. Encrypting eventlogs and/or i/o logs is also recommended. **Note:** Archiving encrypted PowerBroker for Servers log files require that the associated "pb.key" file and "pb.settings" file be archived as well.

10. Take note of the security implications of configuring a submithost or runhost as a "masterhost" and/or "loghost," and the implementation of a PowerBroker for Servers policy. Also, take note of the security implications of allowing a remote request to be processed by PowerBroker for Servers.

Guidelines for Policy

1. When delegating tasks that may have optional switches or arguments that can cause accidental harm, it is a good idea to write a policy that filters command switches and/or arguments.

2. Since shell scripts are readable and relatively easy to modify, use "runcksum" on delegated tasks that are scripts. In general, a dynamic checksum relative to a known standard given by runcksum can ensure the integrity of the command being delegated.

3. When processing a user request, always control the path component of the user request. Let the policy enforce the PATH of commands that are delegated. This will improve the intended security designed in the policy for delegated tasks.

4. When the policy either accepts or rejects a user request, always use either unset() or logomit to remove any auxiliary variables that might contain auxiliary "patterns." This will minimize "noise" when searching through the eventlog or iolog when performing forensic analysis audits.

5. When configuring PowerBroker for Servers in an environment with a firewall, use pbrunreconnection or pblogdreconnection in the policy before either noreconnect or lognoreconnect. Using the latter policy expressions take up more resources.

6. Should the setkeystrokeaction() procedure be used in policy, avoid the use of long and complex regular expressions.

7. When searching for a string in a text file, use 'grep' invoked from the system() function to return a string for large text files (>25M) instead of using readfile()/search().

8. Minimize the use of either the system() function or procedure variant, since the context switch from PowerBroker for Servers to the shell and back can be expensive.

9. Should many system() calls be used, re-write the policy to instead invoke an external script that collects data that is then returned to the policy, thus using only one system call.

10. When iologging a session, put a cap on the amount of data that is captured on the standard input and output streams. This will ensure that file system space on the host acting as the logging host is not overwhelmed with logged data (see: logstdinlimit, logstdoutlimit and logstderrlimit).

Guidelines for a Submithost or Runhost

1. Since a standard PowerBroker for Servers pbrun request will launch the delegated task with pbrun and pblocald processes, the client (submithost/runhost) host will require additional physical memory to accommodate each of the thousands of tasks when each of the thousands of tasks are launched and monitored by PowerBroker for Servers simultaneously. To illustrate, if without PowerBroker for Servers thousands of users can login and have a ksh shell session then if the same number of users are to be monitored with PowerBroker for Servers, for each ksh shell invocation, a pbrun and pblocald process will run along with the ksh process when monitored using standard PowerBroker for Servers to monitor a process.

2. If, on the same client (submithost/runhost), thousands of processes need to be monitored by PowerBroker for Servers 4.0.0 and later, consider running the delegated task in localmode if physical memory is an issue. This approach works so long as each of the delegated tasks does not require "time monitoring" or process resource changes

3. All error logs pertaining to submithost/runhost (in pbrun.log or pblocald.log) can be reset in the event that entries in the error logs are not current.

Guidelines for a Masterhost

1. As each pbrun request will require one pbmasterd process on the masterhost, for thousands of requests a masterhost with a large physical memory to accommodate all the pbmasterd processes will be needed. This is dependent however on how PowerBroker for Servers is configured to start each of the delegated tasks. In the worst case scenario where pbmasterd is required to remain for each pbrun request, as in a tightly firewalled environment configuration, then the masterhost will be saturated with a lot of pbmasterd processes. If possible, ensure that each pbrun request requires a pbmasterd process only to authorize the request. This will reduce the need for additional physical memory from the masterhost.
2. Delegate all logging to a PowerBroker for Servers loghost. If none exists, logging will be done by pbmasterd and will cause pbmasterd to remain, especially when delegated tasks are monitored and logged. When a PowerBroker for Servers loghost is configured, pbmasterd will not assume the role of the logging agent and will exit when the task is delegated.
3. As the policy file must be protected from being changed, ensure the physical and network security of the masterhost.
4. Consider encrypting the policy file and possibly the settings file too. Encrypting the settings file requires that the pb.key file be put in /etc.

5. All error logs pertaining to masterhost(s) (in pbmasterd.log) can be reset in the event that entries in the error logs are not current.

Guidelines for a Loghost

1. Configure a separate loghost with all the necessary physical and network security to protect the host and the logged information.

2. Secure the file system where the logged information will be placed.

3. Control access to the PowerBroker for Servers loghost via any appropriate technology (for example, Kerberos, SSL, smart cards etc) or by selecting an appropriate loghost and configuring this box to use a stronger password hashing algorithm with an appropriately strong password It's best to think in terms of a good "pass phrase," and then use a different root password or "pass phrase" on the loghost.

4. As a pblogd process will be started for each pbrun request, thousands of pbrun requests will cause thousands of pblogd processes on the loghost. A large physical memory will be needed to accommodate all the pblogd agents.

5. Consider encrypting the PowerBroker for Servers eventlog and iologs. Archiving encrypted PowerBroker for Servers log files requires that the associated pb.key file and pb.settings file be archived as well.

6. Eventlogs can grow. Rotating and archiving the eventlog can save disk space. Also, as the PowerBroker for Servers eventlogs and iologs are not large-file aware, letting these files grow large can cause problems. Very large eventlogs and iologs can lead to slow searches when conducting forensic analysis.

7. All error logs pertaining to the log hosts (in pblogd.log) can be reset in the event that entries in the error logs are not current.

beyondtrust®
securing the perimeter within

www.beyondtrust.com

Corporate Headquarters
2173 Salk Avenue
Carlsbad, California 92008

Phone: +1 800-234-9072