

# PUNKBUSTER FOR SERVER ADMINISTRATORS



Wolfenstein: Enemy Territory Edition

Last updated May 17, 2004

The following people contributed to this publication:

Björn Christoph, Ken Creedon, Stuart Dunsmore, Daniel Ferguson, Tony Ray, Jason Roman, Paul Seale, and Hendrik Thole.

Even Balance, Inc.  
P O Box 11395  
Spring, Texas 77391-1395

This publication is © Copyright 2003-2005 by Even Balance, Inc. All Rights Reserved.  
PunkBuster™ Software and the PunkBuster Logo are trademarks of Even Balance, Inc.  
Other trademarks referenced by this publication are the property of their respective owners.

## TABLE OF CONTENTS

- [Introduction](#)
- [What PunkBuster Is All About](#)
- [Quick Start Guide](#)
- [The Basics](#)
  - [Game Integration](#)
  - [PB Subdirectory and File Information](#)
  - [The PunkBuster Interface](#)
- [Enabling and Disabling PunkBuster](#)
- [Updating PunkBuster](#)
- [WebTool](#)
- [Commands and Settings](#)
  - [Commands](#)
  - [Settings](#)
- [Player Name Management Facility](#)
- [Alias Tracking Facility](#)
- [PB RCon Facility](#)
- [Monitoring Game Variables](#)
- [Advanced Cvar Monitoring](#)
- [Screen Capture Facility](#)



- [PunkBuster Player Power](#)
- [Remote Logging Facility](#)
- [Frequently Asked Questions](#)

## INTRODUCTION

This publication is for Server Administrators of online games who are interested in using or finding out more about the PunkBuster Anti-Cheat system. This publication is meant to document PunkBuster for the following games: Wolfenstein: Enemy Territory, by id Software, Inc. PunkBuster editions for other games may vary from the edition described herein.

A great effort has been made to make this publication useful for those new to PunkBuster and also for advanced users of PunkBuster. Any comments or suggestions will be gladly received via email to [docs@evenbalance.com](mailto:docs@evenbalance.com).

Since this manual is for Server Administrators, we will assume for the most part that the reader is familiar with the Game Console, entering commands, and configuration files.

If you wish to learn about using the PunkBuster Client software for use with playing the game, please see our related publication entitled 'PunkBuster for Players'.

## WHAT PUNKBUSTER IS ALL ABOUT

In September of 2000, a handful of online computer game playing enthusiasts became concerned with the increased activity of cheating in online multiplayer gaming. We decided to do something about the problem. We were aware that a few attempts had been made in the past by others (including the makers of the games themselves) and that they were all generally considered to have failed for the most part. We had a vision of a new way to approach the problem and decided to develop an experimental software system to see if our new approach would work. We believed that once we had submitted and maintained a product for open use that the response from affected online communities would tell us whether or not the experiment was a success. After several months of growth, much hard work, acceptance, and the appreciated beta testing and support of hundreds of thousands of users, we believe that we have a viable solution to fighting the problem of cheating in online multiplayer games: PunkBuster.

PunkBuster is an automatically self-updating client/server Anti-Cheat software system. That means that players run the PunkBuster Client software while they are playing online games and also, PunkBuster Server software is running on the game server that players connect to for gameplay. The PunkBuster system is designed to hold all participants accountable by scanning the game computers looking for known cheats, game hacks, and exploits similar to the way Anti-Virus software would scan a computer looking for a virus. PunkBuster does not modify any files or settings on your computer even if it detects some type of violation, it reports what it finds and, in some cases, will remove offending players from the current game. PunkBuster is optional. A Server Admin who decides to run PunkBuster on his or her Game Server is simply choosing to limit players on said Game Server to players who have chosen to enable PunkBuster on their playing computers. You do not have to enable PunkBuster if you are uncomfortable with the idea of such software. However, PunkBuster is not "spyware" nor is it a trojan - it is designed for groups of honest people to use together in an effort to keep out players who are unwilling to subject their system to an objective third-party software system scanning their computer during gameplay. The activities performed by PunkBuster are generally described on our website and we have also developed and published a Privacy Policy Statement. We take the privacy and security of our users and their computers very seriously and there is no featured provision (documented nor undocumented) whatsoever in our software whereby anyone outside of your computer can gain control of your computer or view / change your private information with or without your knowledge. Additionally, PunkBuster does not transmit your private data files to any other computer nor keep any type of centralized database that tracks any information about your personal files. PunkBuster basically just looks for known cheats and game-hacks while you are playing a PunkBuster enabled game in an effort to authenticate your installation as "clean" for the purpose of consensual multiplayer online gaming.

In mid-2001, a privately funded corporation, Even Balance, Inc. was formed to handle the commercial application of the PunkBuster system. We plan to offer several types of products and services to online gaming communities that are built up around the games we support. We are actively seeking relationships with game software developers who would like to have us



provide anti-cheat support for their online multiplayer game titles.

There is no cost for a personal, non-commercial license to use PunkBuster software. Our End User License Agreement must be accepted on-screen before PunkBuster is installed. If you wish to use PunkBuster in any type of commercial environment or in connection with any commercial event, a separate commercial licensing arrangement must be obtained from Even Balance, Inc. To find out more about commercial licensing, send an inquiry via email to [license@evenbalance.com](mailto:license@evenbalance.com).

More information about Even Balance, Inc. and the PunkBuster system can be found on our website at [evenbalance.com](http://evenbalance.com).

## QUICK START GUIDE

If you create a Game Server using the In-Game Browser screen, make sure the PunkBuster setting is set to 'Yes' before proceeding to create the Server. If running the Game Server strictly in console mode, enter `PB_SV_ENABLE` to ensure that the PunkBuster Server software is enabled before proceeding. When PunkBuster is Enabled for the first time, it will begin operation using the default configuration. This configuration has been designed to work with all types of game servers. Optimizations can later be made by adjusting the PunkBuster settings described later in this publication to better match the server connection speed, user load, and Admin preferences. The following are a few options that can greatly enhance server cheat protection with a relatively easy learning curve.

Once PunkBuster has been enabled, automatic screenshot capturing can be turned on by setting `PB_SV_AUTOSS` to 1. To make this (and/or most other PunkBuster setting changes) permanent, enter the command `PB_SV_WRITECFG`. This will save most current PunkBuster settings to the PunkBuster configuration file "pbsv.cfg". PunkBuster loads this configuration file every time it is run. When the automatic screenshot capturing has been turned on, screenshots will begin to be routinely taken of all the servers' connected players and stored in the "svss" folder. The default limit to the number of captured screenshot files is 100 which can be changed with the command `PB_SV_SSCEILING`. Once the limit has been reached, the PunkBuster server will begin to overwrite the current files starting with the lowest numbered file first in an endless cycle.

For Server Admins who wish to log important (but verbose) information about each connected player's cvars during gameplay, enter the following four setting changes into the Game Console. These will have the effect of logging (every 15 minutes) all user-created and non-default player cvars to special log files with a `.var` extension (in the "svlogs" folder).

- `PB_SV_CvarLogging 2`
- `PB_SV_CvarWalk 4`
- `PB_SV_CvarUserPulse 15`
- `PB_SV_CvarChangedPulse 15`

## THE BASICS

### GAME INTEGRATION

---

Although there is code inside the Game to allow PunkBuster to be integrated tightly with the Game, the PunkBuster software is completely separate from the Game software. It resides in its own folder called "pb" inside the game folder. For example, if your game is installed into a folder called "c:\games\et", then the PunkBuster files will be in "c:\games\et\pb". To remove or uninstall PunkBuster from your system, simply delete the "pb" folder using the explorer program on your computer. PunkBuster does not modify the system registry nor hide files or other information anywhere else on your system.

There are only a few files in the "pb" folder when PunkBuster is first installed. When the game is first launched after PunkBuster has been installed, PunkBuster will add new files and folders for later use.

### PB SUBDIRECTORY AND FILE INFORMATION

---



The "pb" folder normally contains the following file types:

- Files with a ".dll" or ".so" extension (example: pbsv.dll) are PunkBuster system files. Removing any or all of these files may cause your PunkBuster system to stop working. Modifying any or all of these files is a violation of our End User License Agreement and permanently terminates your license (and legal ability) to use our software.
- Files with a ".db" extension are PunkBuster database files. The PunkBuster databases are automatically created and maintained by the PunkBuster software. If you delete one or all of these files, PunkBuster will recreate and re-populate them with information obtained from Master PunkBuster Servers if possible. Modifying any of these files may cause PunkBuster to act in an unexpected manner.
- Files with a ".dat" extension are PunkBuster data files. They should not be removed or modified. The PunkBuster system will maintain them automatically.
- Files with a ".log" extension are files into which PunkBuster logs information during the course of using the software. By default, the PunkBuster Client only logs severe error information to a log. The PunkBuster Server logs important activity into log files automatically.
- Files with an ".htm" extension are special log files that are designed to be viewed by a web browser such as Internet Explorer or Netscape.
- Files with a ".cfg" extension are configuration files that contain PunkBuster commands and settings. You can create and modify your own configuration files and have them loaded either automatically or manually. Details about PunkBuster commands and settings are given later in this publication. Two specific ".cfg" files: pbsv.cfg and pbsvlog.cfg may be used and overwritten by the PunkBuster system under some circumstances. Each time PunkBuster starts, the pbsv.cfg and pbsvuser.cfg configuration files are loaded automatically. Note that 'File Not Found' messages related to either of these files does not point to a problem or an error. Using the pb\_sv\_writcfg command will create a pbsv.cfg and overwrite anything that was previously in that file. Admins should create (with any text editor) and use pbsvuser.cfg for PB settings and commands that they don't want to get overwritten with a pb\_sv\_writcfg command.

PunkBuster also creates subfolders inside the "pb" folder to organize information; the following is a list of these folders and what they contain:

- The "htm" folder contains htm files that are used during PunkBuster's auto-update system. Over time and after many updates, this folder may contain old files that may be deleted to recover disk space if desired.
- The "dll" folder contains various PunkBuster system files obtained during the auto-update process from prior version installations of PunkBuster. As with the "htm" files described above, old files in this folder may be deleted to recover disk space if desired.
- The "svlogs" folder contains PunkBuster Server logs. Log files are named in sequence using an 8 digit number and the ".log" file extension.
- The "svss" folder contains screenshots (PNG files) and "helper" HTM files that have been captured from Players connected to the Server - the file pbsvss.htm located in this subdirectory is a sequential index (log) that can be used to quickly and easily view captured screenshots.

## THE PUNKBUSTER INTERFACE

---

All interaction with PunkBuster is performed inside the game itself. The PunkBuster Server logs information both to the Game Server Console and PB Server Logs located in the "svlogs" folder. When you need to issue commands and setting changes to PunkBuster, do so by typing directly into the game's interface / console. Groups of commands and settings can also be placed into .cfg files for easy loading. All PunkBuster Server commands and settings start with the same six characters: "PB\_SV\_". Specific PunkBuster commands and settings are described in a subsequent section of this publication. PunkBuster doesn't care if commands and settings are typed in lower case, upper case, or in some combination. ENABLE, enable, and EnAbLe are all the same as far as PunkBuster is concerned. Any time the PunkBuster Server needs to display a message, the message will be preceded by the words "PunkBuster Server:".

## ENABLING AND DISABLING PUNKBUSTER

There are two ways to enable / disable PunkBuster software. One is by typing commands into the game console. The other way is built into the game's user interface. Only the command line method will be described below, the GUI method is described in our publication "PunkBuster for Players". Note that when PunkBuster is Disabled, it will ignore commands and setting changes except for the specific command to Enable after which it will begin full and proper operation.

To Enable the PunkBuster Server from the console, type in "PB\_SV\_ENABLE" without the



quotes. To Disable PunkBuster, type "PB\_SV\_DISABLE".

NOTICE: PunkBuster cannot be Disabled immediately during the middle of gameplay. Choosing to Disable PunkBuster by any method will set PunkBuster into Disabled mode but that does not take effect until the entire game is restarted. Neither Players nor Server Admins can simply turn PunkBuster off and on during the middle of gameplay. Once PunkBuster Client and/or Server has been Enabled for a gaming session, it becomes Enabled immediately and then stays Enabled regardless of how many times the Enabled/Disabled mode is toggled. When the game is exited, the current PunkBuster status is saved and the next time the game is started, PunkBuster will initialize to the mode in which it was last set, either Enabled or Disabled.

## UPDATING PUNKBUSTER

One of the core aspects of the PunkBuster system is the automatic update feature. As new versions of PunkBuster files are made available for download from our Internet-based Master PunkBuster Servers, the PunkBuster software running on Players' and Admin's computers will attempt to retrieve those new versions and perform an automatic "self-update" function in the background without interrupting gameplay. For most users, this occurs automatically and seamlessly. Since PunkBuster Servers require all connected Players to be running identical versions of all PunkBuster files in order to be authenticated for "clean" online play, the automatic update facility inside PunkBuster is an important component of the overall system. PunkBuster will retain older versions in case there is ever a need to connect to a Server that is still running old versions of the PunkBuster software.

All PunkBuster update files are stored in special compressed HTM files and can be opened with any Internet Web Browser. We have designed and implemented a robust method for securely delivering updates from our Master PunkBuster Servers to Regular PunkBuster Servers and then on to computers that have Enabled the PunkBuster Client software. Great care has been taken to design and develop the system such that only authentic PunkBuster files will ever be accepted and used by the PunkBuster System during the auto-update process.

There may be times, however, when it is necessary to manually update a PunkBuster System. For instance, if there is no Internet Access available and multiplayer gameplay is occurring in a LAN environment. Or perhaps, there is a temporary routing problem causing locally running PunkBuster software to not be able to "find" any Master PunkBuster Servers whereby an update can be obtained. There may also be times that the PunkBuster Team offers pre-release beta versions of PunkBuster software for manual download so that it can be tested before broad distribution. For whatever reason that may come up that requires a manual update, we have prepared a [page](#) on our website designed to help with manual updates.

## WEBTOOL

The PunkBuster WebTool provides the means to remotely administer game servers from any connected computer using a web browser (such as Netscape's Navigator or Microsoft's Internet Explorer). The PunkBuster Server software includes a mini web server that takes input from Web-based forms and translates that input into game commands. This type of access is similar to using rcon (remote console) commands, except it is done through an html interface and works even from computers that do not have the game or PunkBuster installed. Just about anything that can be done at a server's console or via rcon can be achieved easily by using the WebTool. By default, the WebTool is deactivated. The remainder of this section explains how to activate and use the WebTool.

There are five PunkBuster settings associated with the WebTool: pb\_sv\_httpPort, pb\_sv\_httpAddr, pb\_sv\_httpKey, pb\_sv\_httpRefresh, and pb\_sv\_httpMaps. The pb\_sv\_httpPort setting defaults to a value of 0 which means that the WebTool is deactivated. To activate the WebTool, pb\_sv\_httpPort must be set to a port number between 1 and 65535 that is not already in use by another program on the computer running the game server. Most web servers use port 80. If you choose not to use 80, then it is probably best to use a number that is at least 1024 or higher to avoid conflicts with other software. Once pb\_sv\_httpPort is set, PunkBuster will respond saying either that it is listening on the specified port number or that there was some problem trying to listen on the specified port number which means that a different port number should be specified. After setting pb\_sv\_httpPort, you should test the WebTool functionality, if possible, by launching a web browser on the same computer as the game server and enter this URL: <http://127.0.0.1:>. For example, if you chose port number 8000, then you would enter <http://127.0.0.1:8000> for the URL. At this point, you should see the opening login screen for the



If you wish to be able to use the WebTool from another computer connected via network to the game server (even across the Internet if desired), then the pb\_sv\_httpAddr setting needs to be set to hold the routable IP Address of the game server. Please note that the WebTool does not use a secure protocol (such as https) for performance reasons. That means that any information you send from your web browser to the WebTool is sent in plain text (including the Key/Password).

The game server's rconpassword is used to allow remote console access to the game server. By default, the rconpassword is also used for WebTool access. However, if you wish to use a different password for WebTool access, then the pb\_sv\_httpKey should be set accordingly.

One of the screens available inside the WebTool is the Player List screen. This screen shows the list of players and allows Admins to kick and/or ban players easily. By default, this screen automatically refreshes inside the web browser every 30 seconds to show when players leave or when new players join the game. To change the auto-refresh rate, set the pb\_sv\_httpRefresh setting accordingly. Setting pb\_sv\_httpRefresh to 999 will disable the auto-refresh feature for the Player List page.

One of the features of the WebTool allows the Admin to change to a new map by choosing the map name from a drop down list in the web browser. By default, the pb\_sv\_httpMaps setting is empty which means the standard "official" maps are listed in the dropdown. If you wish to customize the map dropdown list, then set the pb\_sv\_httpMaps setting to hold the maps, separated by a space, that you wish to see in the list. For example, setting pb\_sv\_httpMaps to "mp\_beach mp\_ice" means that only those 2 maps will show up in the drop down. Note that you are not limiting the maps available to the server in any way, only what will show up in the WebTool's drop down list.

## LOGIN SCREEN

**PunkBuster Server WebTool for ET**

*Login Screen*

version = ET 2.32 win-x86 Apr 23 2003  
sv\_hostname = ETHost  
net\_ip.net\_port = localhost:27960  
mapname = fueldump

Web Key

Command Screen

Game Settings

Player List

PB Settings

PunkBuster Server (v1.033 | A1313 C1.030) Enabled

© Copyright 2002-2003 by Even Balance, Inc. All Rights Reserved.

The Login Screen is where the Web Key is entered for access to the other screens. Remember that the Web Key is the value of the pb\_sv\_httpKey setting or, if that is blank, the value of the rconpassword game server setting. The four buttons at the bottom of the page take you to the various screens available inside the WebTool provided that the Web Key is entered correctly. These four buttons appear at the bottom of all screens inside the WebTool to make it easy to navigate to the various areas.

## COMMAND SCREEN

**PunkBuster Server WebTool for ET**

*Command Screen*



```
version = ET 2.32 win-x86 Apr 23 2003
sv_hostname = EThost
net_ip:net_port = localhost:27960
mapname = fueldump
```

Web Key

map

map\_restart fueldump

serverinfo

To console->

pb\_sv\_writectg

pb\_sv\_restart

pb\_sv\_update

The Command Screen is provided so that commonly used game commands can be issued very easily. The name on each button corresponds to the exact name of the game server command that it represents. This is also the screen with the map list drop down to make changing maps very easy. If you wish to issue a specific game command that is not covered by one of the buttons, enter your desired command into the "To console->" field and then click the "To console->" button to send that command to the game server.

## PLAYER LIST

### PunkBuster Server WebTool for ET

#### Player List Screen

```
version = ET 2.32 win-x86 Apr 23 2003
sv_hostname = EThost
net_ip:net_port = localhost:27960
mapname = fueldump
```

This page auto-refreshes every 30 seconds

Web Key

	Player	IP Address	Status	Auth	SS	O/S
Slot 01	John	loopback	OK	3.0	0	W
Slot 02	Michael	192.168.2.103:27960	OK	3.0	0	L
Slot 03	[PBStaff]Bjoern	192.168.2.101:27960	OK	3.3	0	W
3 Players						

Slot #

GUID

Reason

Removed by Web Admin

Kick Minutes

Allowed range: 1 to 60

pb\_sv\_kick

pb\_sv\_ban

pb\_sv\_getss

The Player List Screen shows all players and their PunkBuster slot numbers along with the same information displayed by the PunkBuster pb\_sv\_plist command. The bottom of this screen has buttons to allow you to kick, ban or get a screenshot from a specific player. To perform one of these actions, first click the button corresponding to the slot number of the player you wish to reference. That player's information will then automatically be filled into the Slot # and GUID fields at the bottom of the screen. If you plan to kick or ban that player, you may edit the Reason field and if kicking, the Kick Minutes field which defaults to 2 minutes. Press the



appropriate button to take action. Note that pressing the pb\_sv\_getss button will cause the PB Server to request a screenshot from the referenced player. However, screenshots are not viewable through the WebTool interface at this time.

## GAME SETTINGS

### PunkBuster Server WebTool for ET

---

#### *Game Settings Screen*

version = ET 2.32 win-x86 Apr 23 2003  
sv\_hostname = ETHost  
net\_ip net\_port = localhost:27960  
mapname = fueldump

Web Key

sv_maxclients	<input type="text" value="20"/>	Latched (Changes effective upon restarting)
timelimit	<input type="text" value="30"/>	
g_gametype	<input type="text" value="4"/>	Latched (Changes effective upon restarting)
g_friendlyfire	<input type="text" value="1"/>	
g_soldierchargetime	<input type="text" value="20000"/>	Latched (Changes effective upon restarting)
g_engineerchargetime	<input type="text" value="30000"/>	Latched (Changes effective upon restarting)
g_medicchargetime	<input type="text" value="45000"/>	Latched (Changes effective upon restarting)
g_lrchargetime	<input type="text" value="40000"/>	Latched (Changes effective upon restarting)
g_covertopschargetime	<input type="text" value="30000"/>	Latched (Changes effective upon restarting)
g_useralliedrespawntime	<input type="text" value="0"/>	
g_useraxisrespawntime	<input type="text" value="0"/>	

The Game Settings Screen allows common game server settings to be changed easily. Simply change the value of one or more settings and click the "Update" button at the bottom of the screen. The field tags correspond exactly to the game server's cvar names. If you wish to change a setting that is not shown on this page, you can use the "To console->" method found on the Commands Screen (described above).

## PB SETTINGS

### PunkBuster Server WebTool for ET

---

#### *General PB Settings Screen*

version = ET 2.32 win-x86 Apr 23 2003  
sv\_hostname = ETHost  
net\_ip net\_port = localhost:27960  
mapname = fueldump

Web Key

pb_sv_httpRefresh	<input type="text" value="30"/>
pb_sv_httpAddr	<input type="text" value="127.0.0.1"/>
pb_sv_httpPort	<input type="text" value="8080"/>
pb_sv_httpMaps	<input type="text"/>
pb_sv_MsgPrefix	<input type="text" value="^3PunkBuster Server"/>



pb_sv_Sleep	60
pb_sv_KickLen	2
pb_sv_ChangePeriod	999
pb_sv_ChangeMax	1
pb_sv_CvarFreq	6
pb_sv_CvarLogging	1
pb_sv_CvarWalk	1
pb_sv_CvarChangedPulse	99
pb_sv_CvarUserPulse	99
pb_sv_MaxDIRate	4
pb_sv_MaxConDls	4

The PB Settings Screen is used to change PunkBuster settings. Simply change the value of one or more settings and click the "Update" button at the bottom of the screen. Except for the Web Key field, the field tags correspond exactly to the PunkBuster Server's setting names. Please note that the behavior is the same as changing PB Settings from the server console in that they are not saved to the PB config files until/unless you issue a `pb_sv_writcfg` command after making the change. There is a button for this command on the Commands Screen (described above). If you change a PB Setting and want PunkBuster to "remember" the new value even after the server is exited, then you should issue a `pb_sv_writcfg` command after making the change. You should also be aware that the WebTool settings can be changed on this page. If you change the Web Key, the Port number and/or the Address values, then that will affect your current WebTool session. In that case, be prepared to retype any new URL into your web browser and start over with the Login process to continue using the WebTool.

## COMMANDS AND SETTINGS

The easiest way to quickly test that your PunkBuster Server is properly installed, Enabled, and working is to type `"PB_SV_VER"` into the game console. If PunkBuster is working properly, it will respond by displaying the version number that is currently installed and running. If there is no response, then PunkBuster is either not installed properly or has become corrupted and should be reinstalled.

Some commands are standalone and others may require or accept additional parameters. For example, the command `"PB_SV_VER"` is standalone, typing in `"PB_SV_VER"` tells PunkBuster that you would like to know the currently running Version number. The command `"PB_SV_LOAD"` takes one parameter (a filename), typing in `"PB_SV_LOAD ABC.CFG"` tells PunkBuster to load the PB configuration file called `"ABC.CFG"`.

Many Server Admins remotely control their game servers via the `rcon` facility built into the game. Most PunkBuster commands will operate the same way whether issued directly into the game console or remotely. A few of the Server commands listed below send queries to Clients and the responses are not directed back to the user when issued remotely via `rcon`. These commands are denoted by a `{rcon limited}` designation after the command name in the descriptions below.

Most Server Admins will not use the `PB_SV_LOAD` command, but rather will use the game-specific `exec` command so that configuration files can be loaded that have both Game and PunkBuster commands and settings changes. Configuration files loaded by the `PB_SV_LOAD` command may contain only PunkBuster related commands and settings.

PunkBuster settings, sometimes also called variables, hold numbers or textual information that PunkBuster uses while it is operating. Changing PunkBuster settings will affect the way PunkBuster runs in specific ways. All PunkBuster settings start out with default values that are the recommended values for most users. To find out the current value of a setting, simply type in the name of the setting by itself. For example, typing `"PB_SV_AUTOSS"` will cause PunkBuster to display the current value. Also displayed is the allowable range that the value can contain. This particular setting can be set either to 1 or 0.. To set it to 1, type in `"PB_SV_AUTOSS 1"`, PunkBuster will then respond by showing the setting name along with its new value. Trying to set a PunkBuster setting outside of its allowable range will cause



PunkBuster to give the setting the closest allowable value to what was originally specified.

Listed below (in alphabetical order) is a list of PunkBuster Commands and Settings along with a general description and usage instructions where necessary.

## COMMANDS

---

### **PB\_SV\_AliasAutoload [0=no, 1=yes]**

Defaults to 0; if set to 1, PB will reload from the Alias database file after each map change; this is useful for cases where multiple servers share an alias database file so that each instance of PB gets updates from other servers sharing the same file.

NOTE: on busy servers, using this function will increase map load times and should be used only after careful experimentation on your server(s)

### **PB\_SV\_AliasFn [filename]**

Can optionally be set to any file on the file-system to direct PB to use instead of the default pbalias.dat file (must specify full path or else the file will be stored in fs\_basepath location)

### **PB\_SV\_AliasLoad [optional filename]**

Loads the Alias database from the PB\_SV\_AliasFn file (or default pbalias.dat file accordingly); PB will load the database automatically on startup if PB\_SV\_AliasMax is set to non-zero in the pbsvuser.cfg file; this command is provided only as manual method to load an Alias database for a running server

### **PB\_SV\_AliasMax [0-99]**

Setting defaults to 0 which means alias tracking is disabled; set to some value above 0 to enable and to specify how many aliases will be stored for each GUID; if the player uses more aliases than the max, the later ones will be ignored

### **PB\_SV\_AliasMaxEnforce [0-99]**

Defaults to 0; if set to 1, PB will kick players who try to use more than PB\_SV\_AliasMax number of aliases; also if set to 1, PB will warn joining players just like it does for extended ascii characters, etc.

### **PB\_SV\_AliasWrite [optional filename]**

Used to write all the aliases in memory to disk; if desired, an alternate filename (other than pbalias.dat) may be specified after the pb\_sv\_aliaswrite command to have PB write the Alias database to the alternate filename.

NOTE: using PB\_SV\_AliasWrite is optional as PB updates the database at every map change automatically; however, the map change updates only append new information to the file and thus cause it to grow greatly in size on busy servers; conversely, the pb\_sv\_aliaswrite command writes a fresh copy of the entire Alias database to disk, this may take a number of seconds on very busy servers - for best results, we recommend starting up a 'dummy' game server just to issue this command and then exit so as not to adversely affect gameplay on populated game servers

### **PB\_SV\_AList [slot #]**

Lists all aliases used by each GUID; the PB slot# must be specified after the command (the slot# is shown by the PB\_SV\_PList command); the first column in the output states how many times a player (GUID) has used that alias; the 2nd and 3rd columns state when the name was used the first and last time (year-month-day format); the 4th column displays the actual alias used.

NOTE: to search for a specific GUID even if the player is not currently connected, use PB\_SV\_AList (same as above, just specify full 32 character GUID instead of slot#)

### **PB\_SV\_AutoUpdBan [0=no, 1=yes]**

Set to 1 (defaults to 0) if you want PB to automatically update the permanent ban file (pbbans.dat) after each change to the banlist in memory

### **PB\_SV\_BadName [grace\_seconds] [text\_filter]**

Adds a bad name to the list of bad names for the server to disallow in player names

### **PB\_SV\_BadNameDel [slot #]**

Deletes a bad name from the list of bad names

### **PB\_SV\_BadNameList**

Displays the current bad name list for the Game Server; see the section dealing with the Player Name Management Facility

### **PB\_SV\_Ban [name or slot #] [displayed\_reason] | [optional\_private\_reason]**

Removes a player from the game and permanently bans that player from the server based on the player's guid (based on the cdkey); the ban is logged and also written to the pbbans.dat file in the pb folder

### **PB\_SV\_BanEmpty**

Empties the current ban list stored in memory



**PB\_SV\_BanGuid [guid] [player\_name] [IP\_Address] [reason]**

Adds a guid directly to PB's permanent ban list; if the player\_name or IP\_Address are not known, we recommend using "???"

**PB\_SV\_BanList [optional\_search\_text]**

Displays the list of bans in the console, including kicks with a temporary ban; if the [optional\_search\_text] parameter is specified, then only bans/kicks that include the search text are listed (useful for searching for a specific name / guid).

**PB\_SV\_BanLoad [optional\_filename]**

Loads a PB Ban List from the specified file; if no filename specified, then pbbans.dat is loaded

**PB\_SV\_BanMask [IP Address / Subnet Mask]**

Permanently bans players from joining the server from the specified IP Address / Subnet Mask; for example: issuing PB\_SV\_BanMask "12.34." will cause PB to deny access to all players trying to join from an IP Address that begins with "12.34."; These bans are written to the pbbans.dat file just like bans issued with the PB\_SV\_Ban command

**PB\_SV\_BindSrch [search\_text] [player\_name\_or\_slot#] {rcon limited}**

Sends a request to all applicable connected players asking for a report on which local player key bindings contain the specified [search\_text]; to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

**PB\_SV\_ChangePeriod [1-999]**

This setting works in combination with pb\_sv\_changemax. It defines a period of time (in seconds) during which a player may do up to pb\_sv\_changemax name changes. Default is 999 which means disabled.

**PB\_SV\_ChangeMax [1-50]**

This setting works in combination with pb\_sv\_changeperiod. This setting defines how many name changes can be done over a specified period of seconds (pb\_sv\_changeperiod). If the player does more name changes during this period the player will be kicked.

**PB\_SV\_Cvar [Cvar\_name] [IN/OUT/INCLUDE/EXCLUDE] [Param1] [optional\_Param2]**

Adds an automatic cvar check to the list of cvars for the server to check for during gameplay

**PB\_SV\_CvarChanged [player\_name\_or\_slot#] {rcon limited}**

Sends a request to all applicable connected players asking for a list of which local player cvar values have been changed from their original default value; to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

**PB\_SV\_CvarDel [Cvar slot #]**

Deletes a Cvar range check from the list of checks stored in memory; the Cvar slot # can be obtained via the pb\_sv\_cvarlist command

**PB\_SV\_CvarEmpty**

Empties the list of Cvar range checks in memory

**PB\_SV\_CvarList**

Displays the allowed cvar ranges for all cvars that the Game Server is monitoring; see the section dealing with the Variable Monitoring Facility

**PB\_SV\_CvarSrch [search\_text] [player\_name\_or\_slot#] {rcon limited}**

Sends a request to all applicable connected players asking for a list of which local player cvar values contain the specified [search\_text]; to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

**PB\_SV\_CvarUser [player\_name\_or\_slot#] {rcon limited}**

Sends a request to all applicable connected players asking for a list of which local player cvars were created by the user (as opposed to being part of the game itself); to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

**PB\_SV\_CvarVal [Variable\_Name] [player\_name\_or\_slot#] {rcon limited}**

Sends a request to all applicable connected players asking for the value of the specified variable (cvar); to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

**PB\_SV\_Disable**

Disables the PunkBuster Server Software - the disabling does not take effect until the game server is exited and restarted

**PB\_SV\_Enable**

Enables the PunkBuster Server Software

**PB\_SV\_GetSs [player\_name\_or\_slot#]**

Sends a request to all applicable connected players asking for a screen shot to be captured



and sent to the PB Server; to specify a player name or substring (as opposed to slot #), surround the text with double-quote marks

#### **PB\_SV\_IPGuard [IP Address]**

The PunkBuster Server automatically adds suspicious IP Addresses to the IP Guard list when players join with a non-functioning PunkBuster client; Players joining from a guarded IP with an old version of PunkBuster are removed from the server; The PB\_SV\_IPGuard command allows admins to add their own suspicious IP Addresses to the list

#### **PB\_SV\_Kick [name or slot #] [minutes] [displayed\_reason] | [optional\_private\_reason]**

Removes a player from the game and won't let the player rejoin until specified [minutes] has passed or until the server is restarted, whichever comes first - kicks are not written to the pbbans.dat file but they are logged and will show up in the output from the pb\_sv\_banlist command

#### **PB\_SV\_Load [File Name]**

Loads the specified PunkBuster configuration file which can contain PunkBuster commands and/or setting changes

#### **PB\_SV\_NameLock [GUID or partial match] [Name to Lock]**

Causes the PunkBuster Server to kick players wearing a locked name unless the player's GUID contains the specified GUID or partial match associated with a locked name; NameLocks are not stored to disk automatically so Admins should add these to the pbsvuser.cfg file in order to have NameLocks extend to future server sessions

#### **PB\_SV\_NameLockEmpty**

Empties the list of NameLocks in memory

#### **PB\_SV\_NameLockList**

Displays the list of NameLocks to the server console

#### **PB\_SV\_NewLog**

Causes PunkBuster to close the current PunkBuster log and open a new one

#### **PB\_SV\_PList**

Displays a list of connected players and their current status

#### **PB\_SV\_Power [slot #] [power rating]**

Adds player in specified slot # to the locally stored PB Player Power database with the specified power rating. (NOTE: The order of the arguments for this command was reversed for consistency starting in version 0.996).

#### **PB\_SV\_PowerList [filter]**

Displays a list of Power Players in the database; if filter is specified, then only entries where the filter is either in the guid and/or name fields are displayed

#### **PB\_SV\_PowerPoints [power slot #] [points]**

Changes the number of Power Points assigned to a Power Player in the database; use the PB\_SV\_PowerList command to get the power slot #

#### **PB\_SV\_ProtectName [Unique ID] [Registered Name]**

Protects the specified PunkBuster Registered Name on the local server so that players who wear the protected name without having the correct Registration Credentials will be removed from the server

#### **PB\_SV\_ProtectTag [Unique ID] [Registered Tag]**

Protects the specified PunkBuster Registered Tag on the local server so that players who wear the protected tag without having the correct Registration Credentials will be removed from the server

#### **PB\_SV\_RCon [min power points] [command prefix]**

Add a new command to the PB RCon list and define the power points required to issue the command.

NOTE: you are entering prefixes - if a player enters a command that "starts with" an exact entry in the list, then it will be considered valid as long as the player has enough points to issue that command. For example, entering `PB_SV_RCon 75 PB_SV_` means that any player with 75 or more power points can issue any PB server command (not recommended by the way unless you are the only player with 75 or more power points and you wish to do this for yourself).

#### **PB\_SV\_ReBan [slot #]**

Rebans a player who has been mistakenly Unbanned with the pb\_sv\_unban command; use pb\_sv\_updbanfile to update the permanent ban file after using this command

#### **PB\_SV\_RList**

Display the list of all currently defined RCon command prefixes as well as the power points required to issue each command prefix.



**PB\_SV\_Task [X] [Y] [command]**

Adds a task to PB's Task List; The Task will be executed X seconds after entry and every Y seconds thereafter; use -1 for Y if a one-time task is desired; this can be used to execute game server commands as well as PB commands

**PB\_SV\_TaskDel [task slot #]**

Removes the task from PB's Task List that corresponds to the specified task slot #

**PB\_SV\_TaskEmpty**

Empties the list of Tasks in memory

**PB\_SV\_TList**

Displays a list of PB Tasks along with a task slot # for each task

**PB\_SV\_UnBan [slot #]**

Unbans a player from the ban list stored in memory; use pb\_sv\_updbanfile to update the permanent ban file after using this command

**PB\_SV\_UnBanGuid [guid]**

Unbans a guid from the ban list stored in memory; use pb\_sv\_updbanfile to update the permanent ban file after using this command

**PB\_SV\_Update**

Forces the PB Server to attempt a PB software update even if no players are currently connected

**PB\_SV\_UpdBanFile [optional filename]**

Updates the specified ban file (pbbans.dat is used if none specified) with regard to recent unbans/rebans

**PB\_SV\_Ver**

Displays the currently running version of the PunkBuster Server software

**PB\_SV\_WriteCfg**

Writes the current values of the PunkBuster Server settings to the local hard drive (creating or overwriting a file called pbsv.cfg) in such a way that they will be loaded automatically the next time the PunkBuster Server starts; server admins who wish to manage multiple config files for different situations will usually not use this command at all

## SETTINGS

---

**PB\_SV\_AutoSs [0=no, 1=yes]**

Set to 1 (default is 0) if you want the PB server to regularly retrieve screen shots from connected players

**PB\_SV\_AutoSsFrom [Seconds]**

Minimum number of seconds (default is 60) PB will wait before requesting a screen shot after the previous request from each player

**PB\_SV\_AutoSsTo [Seconds]**

Maximum number of seconds (default is 1200 = 20 minutes) PB will wait before requesting a screen shot after the previous request from each player

**PB\_SV\_CQC [0=no, 1=yes]**

CQC means Client Query Capability - setting this to 0 (default is 1) means that connected players cannot use PB to check the value of game server cvars (we recommend leaving this set to the default of 1 to promote goodwill); NOTE that PB doesn't let players see the value of any server-side cvars that include the text "pass" nor any PB settings

**PB\_SV\_CvarChangedPulse [Number]**

The number of minutes (default is 99 which means disabled) between automatic sends of the CVARCHANGED command described in the command section above

**PB\_SV\_CvarFreq [Number]**

The number of times per minute (default is 6) that each player has one cvar value checked against the current list of cvar ranges on this server

**PB\_SV\_CvarLogging [0/1/2/3]**

Determines the target destination of log output related to player cvar checks; The value of 1 (default) specifies the server console and normal log file, the value of 2 specifies separate log files in the "svlogs" folder with the var file extension, and the value of 3 specifies both

**PB\_SV\_CvarUserPulse [Number]**

The number of minutes (default is 99 which means disabled) between automatic sends of the CVARUSER command described in the command section above



**PB\_SV\_CvarWalk [Number]**

The default setting of 0 tells PunkBuster to simply log lists of cvar names returned by the `CVARSRCH`, `CVARUSER` and `CVARCHANGED` commands so that the Admin would have to manually query the individual values if desired; a non-zero setting tells PunkBuster to "walk" through each cvar returned in such a list automatically and log the values of each cvar - the higher the number for this setting, the more cvars PunkBuster walks through during each processing cycle

**PB\_SV\_DupNameGrace [Seconds]**

Set to the number of seconds (default is 0 which means duplicate name kicking is disabled) of grace period a player should be granted to change names before being kicked for having the same (duplicate) name as another player (first come first served) on the server

**PB\_SV\_EmptyName [0=no, 1=yes]**

When set to 0 (which is the default), PunkBuster will remove players who join with an Empty Name or who change to an Empty Name during gameplay

**PB\_SV\_ExtChar [0=no, 1=yes]**

The default setting of 0 tells PunkBuster to disallow extended ASCII Characters in player names; for the purposes of this command, characters that cannot be easily entered with simple keystrokes are considered extended

**PB\_SV\_GUIDRelax [0-7]**

Defaults to 0; Controls PunkBuster's kicking behavior related to GUIDS; A Value of 1 means PB will not kick for UNKN (Unknown) GUIDs; A Value of 2 means PB will not kick for WRONGIP GUIDs (these are GUIDS which are valid but not from the IP Address the player is connecting from); A Value of 4 means PB will not kick for DUPLICATE GUIDs; These values (1, 2 and 4) can be combined to achieve the desired behavior

**PB\_SV\_HttpAddr [IP Address]**

The IP Address of the computer running the PunkBuster WebTool; if this setting is left empty ("" which is the default), then the WebTool will only operate properly when used with a web browser running on the same machine as PunkBuster itself (i.e. using 127.0.0.1 as the IP address)

**PB\_SV\_HttpKey [Key (password) for WebTool use]**

The Key or Password used by the WebTool to limit access to its features; if this setting is left empty ("" which is the default), then the game server's rconpassword will be used instead; if both are empty, then the WebTool will not be usable at all

**PB\_SV\_HttpMaps [List of Maps]**

By default, this setting is empty which means that the standard official maps will be displayed in the 'change map' dropdown list inside the WebTool; if you wish to customize the entries in the dropdown, then use this setting to store the list of maps (each separated by a space)

**PB\_SV\_HttpPort [Port Number]**

The TCP port number used by the WebTool's http server; this setting defaults to 0 which means the WebTool is deactivated

**PB\_SV\_HttpRefresh [Number]**

The number of seconds between auto-refreshes of the Player List screen inside the WebTool; setting this to 999 means that the page will not auto-refresh at all

**PB\_SV\_KickLen [Minutes]**

The number of minutes (default is 2) a player will be kept from being able to rejoin after getting kicked by PunkBuster

**PB\_SV\_LanMask [IP Subnet Mask]**

This setting is used for Internet Servers that also have local LAN game clients connecting; it should be set to hold the subnet of the LAN; for example, if the LAN IP Addresses are 192.168.1.x, then set pb\_sv\_lanmask to 192.168.1

**PB\_SV\_LogAddr [Address / Internet Hostname]**

Holds the address of the machine to which remote logging is sent (default is empty meaning the feature is not used)

**PB\_SV\_LogCeiling [Number]**

The highest serial number (default is 1000) that PunkBuster will use in numbering its log files before starting over at 1

**PB\_SV\_LogNext [Number]**

The next serial number that PB will use to name a PB log file; this setting is automatically maintained by PunkBuster and is incremented after each map change by the Game Server as PB opens a new log file

**PB\_SV\_LogPort [Port #]**

Holds the listen port of the machine to which remote logging is sent (default is 0 meaning the feature is not used)



**PB\_SV\_LogPw [Password]**

Holds the password required to send logging output to a listening remote logging machine

**PB\_SV\_LogSync [0=no, 1=yes]**

When set to 1 (default is 0), PB will name Game log files in accordance with PB log files using the same serial number system; Game log files will still be stored in the same location inside the game directory either way, but will have the same filename as the corresponding PB log file and both will be closed at the end of each map and a new log file will be opened

**PB\_SV\_LogUser [Username]**

Holds the username required to send logging output to a listening remote logging machine

**PB\_SV\_MaxConDls [Number]**

The maximum number (default is 3) of PB updates that PB will attempt to download at the same time

**PB\_SV\_MaxDIRate [KB/sec]**

The (roughly) maximum bandwidth (default is 1 KB/sec) requested per file for PB update downloads

**PB\_SV\_MinName [0-4]**

Holds the minimum player name length (default is 0) allowed on the server (after stripping color codes)

**PB\_SV\_MsgPrefix [New Prefix]**

Holds the text that the PunkBuster Server displays in front of every output line displayed; the default is "^3PunkBuster Server" (note the ^3 means output in the color GOLD); If the game server is running in non-dedicated (listen) mode, all PunkBuster Server output goes to both the top of the playing screen and to the console by default; To prevent PunkBuster Server messages from being displayed on the playing screen, insert the text [skipnotify] before any desired message prefix text. For example, the command pb\_sv\_msgprefix [skipnotify]^3PunkBuster Server will cause all PunkBuster Server output to go only to the console and not to the playing screen.

**PB\_SV\_NoGuidGrace [Seconds]**

Holds the number of seconds PB will wait (default is 1) before kicking players who join without having a GUID; Refusing to enter a CDKey into the game causes the "no guid" condition

**PB\_SV\_PowerDef [power points]**

Holds the default number of Player Power points (default is 1) assigned to players who are not in the locally stored PB Player Power database

**PB\_SV\_PowerKickLen [minutes]**

The number of minutes (default is 5) a player will be kept from being able to rejoin after being removed via the PB Player Power facility

**PB\_SV\_PowerMin [power points]**

A player is removed from the server when the number of power points applied against him/her is equal to or greater than the value of this setting (default is 10)

**PB\_SV\_ScoreKick [low\_negative\_score]**

Players whose score drops below this setting will be kicked from the server; the default of 0 means disabled; the allowable range is -1 to -20

**PB\_SV\_Sleep [Period]**

Holds the period of time (milliseconds) that the PunkBuster Server "sleeps" between processing cycles; lower numbers will cause PunkBuster to process more times each second which also may have the effect of marginally increasing the bandwidth used by PunkBuster

**PB\_SV\_Specname [text]**

If a special spectator client "bot" is used on the server, this setting should hold the player name corresponding to that client so that the PB Server will know this is not a real player; if the player with this name ever has a non-zero score, the "special" status is lost

**PB\_SV\_SsCeiling [Number]**

The highest serial number (default is 100) that PB will use in numbering Screenshot (PNG) files obtained from players before starting over at the PB\_SV\_SsFloor value

**PB\_SV\_SsCmd [Filename]**

The name of the script file (default is "" empty which means "not used") that PB calls after each screen shot PNG image file is received; this can be used by admins to copy or otherwise process screenshot files that are obtained automatically during gameplay

**PB\_SV\_SsDelay [Seconds]**

When this is non-zero (default is 0), then each PB client will wait a random number of seconds up to the value of this setting after receiving the request before actually capturing a screen image for sending back to the server



**PB\_SV\_SsFloor [Number]**

The lowest serial number (default is 1) that PB will use in numbering Screenshot (PNG) files obtained from players

**PB\_SV\_SsHeight [Pixels]**

The requested height (default is 240 pixels) of images captured by PunkBuster Clients for sending to the PB Server

**PB\_SV\_SsNext [Number]**

The next serial number that PB will use to name a PNG screen shot image file

**PB\_SV\_SsPath [Path]**

If specified (default is "" empty), PB will write captured screen shot images obtained from connected players to this alternate location, this can be a network share or some other location where you would prefer to place screen shot images and the "helper" htm files that go with them

**PB\_SV\_SsRate [Number]**

The sample rate (default is 1) used for capturing screen shots, specifies how many pixels get skipped in the processing of the image to keep file sizes down; if set to 2, then only every 2nd pixel is taken (in both horizontal and vertical directions); if set to 4, then only every 4th pixel is taken

**PB\_SV\_SsWidth [Pixels]**

The requested width (default is 320 pixels) of images captured by PunkBuster Clients for sending to the PB Server

**PB\_SV\_SsXPct [Percentage]**

The percentage across the screen (default is 50%) where the center of the requested screenshot should be captured from

**PB\_SV\_SsYPct [Percentage]**

The percentage down the screen (default is 50%) where the center of the requested screenshot should be captured from

**PB\_SV\_UpdateGrace [Seconds]**

Holds the number of seconds (default is 600) that PunkBuster allows for a player to successfully update to the version of PunkBuster currently in use at the server

## PLAYER NAME MANAGEMENT FACILITY

Some Server Admins wish to try to limit profanity / offensive language on their game servers. This is an individual preference issue. For Admins who wish to prevent players from using certain offensive text in their playing names, PunkBuster includes the Player Name Management Facility also known as the BadName Facility.

By default, PunkBuster does not do any checking for BadNames. If you wish to construct a list of offensive text filters in PunkBuster so that players will not be able to wear names containing the offensive text, use the BadName command. The format of the BadName command is as follows:

```
PB_SV_BADNAME [grace_period] [filtered_text]
```

You can fill up a configuration file with nothing but these types of statements if desired and then issue a PB\_SV\_LOAD command either manually or inside your pbsv.cfg file to have the whole list executed easily at one time. If preferred, you can also type them into the console and do a PB\_SV\_WRITECFG to have the whole list written, along with your other current PB settings, to the pbsv.cfg file which is automatically loaded every time PunkBuster starts.

If PB detects that a player has a name that includes the exact [filtered\_text] from any BadName entry (case insensitive), then PunkBuster will send a warning message to the player that the playing name is not allowed on this server and that the player should change the name. If the name is not changed to an acceptable name within the number of seconds specified by [grace\_period] for that text filter, then the player is removed from the game. Note that setting [grace\_period] to 0 has the effect of immediately removing a player before any warning message has time to display.

Note that PB strips player names of color codes before searching for [filtered\_text].



# ALIAS TRACKING FACILITY

The Alias Tracking Facility allows admins to set up PB for tracking different names used by the same player (GUID). Every time a player joins or changes player name, the PB server will check to see if the name being used is already logged in the Alias database; if so, the count is incremented and the Date last used field is updated; if it is the first time for a given alias/GUID combination, then a record is appended to the pbalias.dat file in the 'pb' folder.

To enable the Alias Tracking Facility, set `PB_SV_ALIASMAY` to some value above 0 and to specify how many aliases will be stored for each GUID. If the player uses more aliases than the max, the later ones will be ignored.

You may set `PB_SV_ALIASMAYENFORCE` to 1 if you wish to kick players who try to use more than `pb_sv_aliasmax` number of aliases. In this case, PB will warn joining players just like it does for extended ascii characters, etc.

If you wish to see all aliases used by some connected player based on their GUID, enter `PB_SV_ALIST slot#`. The output has the following meaning: The first column in the output states how many times a player (GUID) has used that alias; the 2nd and 3rd columns state when the name was used the first and last time (year-month-day format); the 4th column displays the actual alias used. If the player is not currently connected, you can also use the full 32 character GUID instead of slot#.

# REMOTE CONSOLE FACILITY

The PB RCon Facility allows admins to give specified players something similar to the normal RCon ability of the game. The Facility is based on guids that are generated for each player, so passwords are not used for this feature at all. This will work with anything that would normally be entered at the server console including commands, settings, etc. even specific to whatever mods you are running at your server. This Facility is based on the [PunkBuster Player Power](#) system whereby PB Admins can give power points to their regular players and deputy admins. Admins therefore have to assign Power Points to players before they can use this facility.

The PB Server keeps a list of commands for use by the PB rcon facility; by default, that list is empty; use `PB_SV_RList` to see the current list at any time.

First you need to decide levels of control for your server. Decide which commands you want to make available with this facility and group them by "power points".

An example:

- 100 power points required to permanently ban players
- 50 points required to change the map
- 25 points required to change `sv_minping` and `sv_maxping` server cvars.

You now have to use the `pb_sv_rcon` command to set up your list: `PB_SV_RCon [min power points] [command prefix]`. PB will automatically update the `pbrcon.dat` file when the `pb_sv_rcon` command is issued. That file is loaded every time the PB Server is started and restarted. So for our example, you would enter the following 4 lines:

- `PB_SV_RCon 100 pb_sv_ban`
- `PB_SV_RCon 50 map`
- `PB_SV_RCon 25 sv_minp`
- `PB_SV_RCon 25 sv_maxp`

Note that you are entering prefixes - if a player enters a command that "starts with" an exact entry in the list, then it will be considered valid as long as the player has enough points to issue that command. For example, entering `PB_SV_RCon 75 PB_SV_` means that any player with 75 or more power points can issue any PB server command (not recommended by the way unless you are the only player with 75 or more power points and you wish to do this for yourself).

The PB server will announce to all players when a command has been executed and say who executed it. If a player tries to use `PB_RCON` and does not have the requisite power points or tries to issue a command that is not in the list, then the PB server logs that activity but otherwise ignores it. No feedback is sent to players - successful or unsuccessful. There is no "secrets"



with this command. Players can't use it to do anything anonymously. Any output associated with an issued command is logged to the server console and log files, but is not returned to the player's console.

This will work with anything that would normally be entered at the server console including commands, settings, etc. even specific to whatever mods you are running at your server.

So basically when you bestow enough power points to a player so that he/she can use one or more PB rcon commands, you should inform/educate that player on the specific commands that are available to him/her. There is no way for a player to "find out" what can be issued with a certain number of points unless the admin or other power players gives the info i.e. there is no client command to list what is available. Players just have to know what they are allowed to do and on which servers with pb\_rcon.

If you re-issue the same command with pb\_sv\_rcon, then PB will update the existing record so that you can change the number of power points required. If you want to "delete" a command, simply set the required points very high so that no players on your server have that many points (or edit pbrcon.dat with a text editor, remove the command(s) you want to delete and then do pb\_sv\_restart).

One advanced idea that you may want to experiment with is to create short scripts to do various things on your server. You can name the scripts anything you want and just make your list of PB rcon commands to simply call your scripts. So players don't have to know any real commands at all, they would just do 'pb\_rcon exec script\_name' based on what you tell them they can do and then whatever you have in the script will be exec'd when the rcon is issued.

If you wish to learn how to use PB\_RCon as a Client please see our related publication entitled 'PunkBuster for Players'.

---

The PB RCon Facility allows admins to give specified players something similar to the normal RCon ability of the game. The Facility is based on guids that are generated for each player, so passwords are not used for this feature at all. This will work with anything that would normally be entered at the server console including commands, settings, etc. even specific to whatever mods you are running at your server.

We assume that the Admin of the PB Server has set up a list of pb\_rcon commands, and has also assigned you Power Points so you can execute pb\_rcon commands. For more information on Power Points check the [PunkBuster Player Power](#) section of this documentation. If you wish to learn how to enable the PB\_RCon as an Admin please see our related publication entitled 'PunkBuster for Admins'.

As example, the admin has set up following list of RCon commands:

- 100 power points required to permanently ban players (pb\_sv\_ban)
- 50 points required to change the map (map)
- 25 points required to change sv\_minping and sv\_maxping server cvars (command prefixes used here are sv\_minp and sv\_maxp).

To issue PB\_RCon commands, you will need to use `PB_RCon` in your console while playing - you must be connected to the server and past the point where the PB server issues their GUID. You may now type PB\_RCon followed by the command you wish to execute.

If you have 30 power points you can successfully issue this command using our example: `/PB_RCon sv_maxping 150.`

As a player with 50 points you could as well successfully issue `/PB_RCon map q3dm1.`

Once a PB\_RCon command has been executed by you, the PB server will announce to all players that the command has been executed and say that you executed it. If you are trying to use PB\_RCon and do not have the requisite power points or try to issue a command that is not in the list, then the PB server logs that activity but otherwise ignores it. No feedback is sent to players - successful or unsuccessful. There is no "secrets" with this command. You can't use it to do anything anonymously. Any output associated with an issued command is logged to the server console and log files, but is not returned to your console.

## MONITORING GAME VARIABLES

PunkBuster provides the ability for Server Admins to specify a list of acceptable player cvar value ranges while playing on the Game Server. When each player connects to a PB-Enabled Server, the PB Client will download the list of acceptable cvar ranges from the PB Server and will check



all of the player's cvars for compliance. The player is warned about any cvars that are unacceptable to the Game Server. The player then will have a few seconds to make any necessary corrections before the PB Server will begin methodically and regularly checking that player's cvars against the list of acceptable ranges. If a player is found to have a cvar that is outside the acceptable range for that cvar, then that player will be removed from the game immediately.

To add a cvar check to your PB Server, use the `PB_SV_CVAR` command. There are four different types of cvar checks that can take place: `IN`, `OUT`, `INCLUDE`, and `EXCLUDE`. Each is explained below along with examples and the format of the `PB_SV_CVAR` usage required to properly add that type of check to the PB Server's list. As with any other PB Command or Setting, cvar checks can be stored in configuration files and loaded either automatically during startup or upon demand during gameplay. In the case of cvar checks, we recommend adding new entries in such a way that they take effect the next time the server is started rather than adding them during gameplay.

Note that once a cvar check has been added, it cannot be removed or changed without restarting the PB server. If a subsequent entry is made with the same cvar name, it is added to the list without affecting the prior entry. Also note that checking for non-existent cvars will return empty strings (which also evaluate to the value of 0).

```
PB_SV_CVAR [cvar_name] IN [from_value] [optional_to_value]
```

If you wish to require that a particular cvar always equals a certain value or falls within a specified range of values, use the "IN" type. For example, "`PB_SV_CVAR handicap IN 10`" means that each player's handicap cvar must always equal 10 on your game server. Issuing "`PB_SV_CVAR handicap IN 5 15`" means that handicap must always fall inside the 5 to 15 range (inclusive).

```
PB_SV_CVAR [cvar_name] OUT [from_value] [optional_to_value]
```

If you wish to require that a particular cvar never equals a certain value or falls within a specified range of values, use the "OUT" type. For example, "`PB_SV_CVAR handicap OUT 0`" means that each player's handicap cvar must never equal 0 on your game server. Issuing "`PB_SV_CVAR handicap OUT 11 99`" means that handicap must never fall inside the 11 to 99 range (inclusive).

```
PB_SV_CVAR [cvar_name] INCLUDE [text]
```

If you wish to require that a particular cvar always includes the specified [text] inside its value, use the "INCLUDE" type. For example, "`PB_SV_CVAR r_drawbuffer INCLUDE gl_back`" means that each player's `r_drawbuffer` cvar must always have the text "gl\_back" somewhere in the value.

```
PB_SV_CVAR [cvar_name] EXCLUDE [text]
```

If you wish to require that a particular cvar never includes the specified [text] inside its value, use the "EXCLUDE" type. For example, "`PB_SV_CVAR name EXCLUDE ^`" means that each player's name cvar must never have the text "^" somewhere in the value (note: the ^ symbol is used to change text colors so using the example would mean that no one could put fancy colors in their playing name on your server without getting kicked once detected by PunkBuster).

## ADVANCED CVAR MONITORING

In addition to the Variable Monitoring Facility described in the previous section, PunkBuster also includes several advanced tools that Admins can use when desired. Five such commands are `PB_SV_BINDSRCH`, `PB_SV_CVARSRCH`, `PB_SV_CVARVAL`, `PB_SV_CVARUSER`, and `PB_SV_CVARCHANGED`. The first three take two parameters: [text] [player] and the others take one: [player]. If the [player] parameter is not specified for any of these commands, then PB will direct the command at all currently connected players. There are 2 ways to specify specific players: by slot # or by name substrings. To get player slot numbers, use the `PB_SV_PLIST` command. Each player's slot # is in the first column of the resulting display. If you wish to specify name substrings, be sure to surround the search text with double quotes. For example, `PB_SV_CVARVAL handicap "et"` means to find the value of the handicap cvar for all players who have the text "et" in their playing name.

The `PB_SV_BINDSRCH` command searches through all of the player's key bindings looking for the specified [text] and reports back matching keys and the full binding attached to each matching key. This is useful for examining the key bindings of a player to see if that player is trying to get around the cvar checks by setting a bind to set a cvar value to an unacceptable value and then quickly set it back to acceptable after some other processing has occurred. For example, use "`PB_SV_BINDSRCH back`" to get a list of all keys that have the text "back" in the key binding from your connected players.

The `PB_SV_CVARSRCH` command is similar to the `BINDSRCH` command described above. However, instead of searching key bindings, it searches through the values of all cvars defined



in the player's system, including player-defined cvars. Since cvars can be used to execute commands by players, this type of search is useful to find out if a player has set up a cvar as an alias to execute in order to set other cvar values outside of acceptable ranges. When this command is issued, the full list of player cvar names that contain the search text is returned (the values of each cvar may be displayed/logged either manually with the `PB_SV_CVARVAL` command or automatically by setting the `PB_SV_CVARWALK` setting).

The `PB_SV_CVARVAL` command can be used to check the exact value of any player cvar at any time. For example, use `"PB_SV_CVARVAL cg_drawallweapons"` to find the value of the `cg_drawallweapons` cvar for all connected players. Use `"PB_SV_CVARVAL cg_drawallweapons 1"` to find the value only from the player in slot #1.

The `PB_SV_CVARUSER` command requests a list of cvars that are user-created.

The `PB_SV_CVARCHANGED` command requests a list of cvars that are not currently set to their default value.

There are also a few advanced settings that can be used to automate routine logging of cvar information supplied by using the above-listed commands:

The `PB_SV_CVARLOGGING` setting determines how and where player cvar information is displayed and/or logged to a file for later viewing. A busy server with several players can generate lots of cvar-related information within a relatively short period of time if the PB Admin decides to take advantage of some of the above-listed features. If the `PB_SV_CVARLOGGING` setting is left at the default value of 1, then all cvar-related information is treated just like other events and is displayed in the server console as well as logged to the current PunkBuster log file. To redirect cvar-related logging to a separate (var) file, set `PB_SV_CVARLOGGING` to 2. In this case, PunkBuster actually maintains two separate log files for each map, one with the `.log` extension (normal logs) and one with the `.var` extension (cvar logs). Both files for a given map will have the same filename based on the sequential log file number, only the file extensions are different (for example, `00000001.log` and `00000001.var`). Setting `PB_SV_CVARLOGGING` to 3 has the effect of logging cvar-related information in both manners described above for values of 1 and 2. Setting to 0 means PunkBuster does not log cvar information.

The `PB_SV_CVARUSER` and `PB_SV_CVARCHANGED` commands described above are designed to catch players who have figured out how to exploit the game using unpublished combinations of settings. If an Admin is suspicious of a certain player who seems to be able to do things that are not a normal part of the game, that player's cvars can be examined for any unusual combinations that may bring to light the source of an exploit. In some cases, Admins (or Leagues) may want to routinely and automatically log all such information for all players and then go back later to examine the logs rather than try to check players in real time while they are playing. To automate the sending of `PB_SV_CVARUSER` and `PB_SV_CVARCHANGED` commands to all players, use the `PB_SV_CVARUSERPULSE` and `PB_SV_CVARCHANGEDPULSE` settings. For example, set `PB_SV_CVARUSERPULSE` to 15 in order to have the PunkBuster Server automatically send a `PB_SV_CVARUSER` command to each player once every 15 minutes (staggered randomly). Both PULSE settings default to 99 which means disabled. Setting either or both to a number up to 98 activates the automatic feature.

Another advanced setting that is designed to be used in combination with the PULSE settings mentioned above is called the `PB_SV_CVARWALK` setting. This setting normally defaults to 0 which means disabled. Setting it to a non-zero number causes PunkBuster to automatically "walk" through all cvars returned in a list (such as from a `PB_SV_CVARUSER` command) and query the actual current value of each player cvar in the list.

Listed below is an example set of settings that would cause the PunkBuster Server to log all user-created and changed cvars along with their values to a separate log file with the `.var` file extension every 15 minutes (per player) during gameplay:

```
PB_SV_CvarLogging 2
PB_SV_CvarWalk 4
PB_SV_CvarUserPulse 15
PB_SV_CvarChangedPulse 15
```

## SCREEN CAPTURE FACILITY

PunkBuster allows Server Admins to request and retrieve actual screenshot images being displayed on the gaming monitor of players currently connected to the game server. It is very easy to request a screenshot from one or more players and also to set up automatic screenshot capturing. Several advanced features are available for Admins who want to tweak their systems. The defaults are fine for most Admins and are a good place to start when getting a feel for how the Screen Capture Facility works.



There are two ways to capture screen shots: Manually and Automatically. Some Admins will do both types. To Manually request a screenshot from all connected players, simply type "PB\_SV\_GETSS" into the Game Server console and after a few seconds, you will start seeing messages about where newly retrieved screenshots were written to your hard drive and what their filenames are. PB names screenshot image files using incrementing sequential serial numbers and they always have the PNG extension. PNG graphic files are similar to GIF and JPG files - they combine the best of both worlds with respect to the type of images found in most computer games. Along with each screenshot image (PNG) file, PunkBuster also writes a "helper" HTM file with the same filename so that you can use any web browser, such as Netscape or IE, to easily view your captured screenshots. Furthermore, all screenshot requests are logged to one specific HTM file called pbsvss.htm so that the list is made available on one scrolling screen as an index for quick and easy viewing.

If you wish to take a manual screenshot of only one or a few connected players, you can specify either the player's slot number directly or you can specify a text substring and PB will request screenshots from all players whose names contain the specified search text. To find the slot # of a specific player, type PB\_SV\_PLIST into the console. Column one contains the slot # for each connected player in the list. To request screenshots from all players with the text "ABC" in their name, type PB\_SV\_GETSS "ABC" (note the quotes are necessary when specifying name substrings).

To set up Automatic Screenshot Capturing, set the PB\_SV\_AUTOSS setting to 1 (the default is 0). Then you may want to change the values of the PB\_SV\_AUTOSSFROM and PB\_SV\_AUTOSSTO settings to specify how often PunkBuster will request screenshots from each connected player. Both of these settings are measured in seconds and the defaults are 60 and 1200 respectively which means that the PunkBuster Server will request a new screenshot randomly anywhere from 1 to 20 minutes after each previous request for each connected player.

Note that regardless of the combination of manual and automatic screenshot requests submitted during gameplay, the PunkBuster system limits screen shots in two ways. First, each individual screenshot is limited to 82,000 pixels in an effort to keep image file sizes and transfer bandwidth requirements down. If you set your parameters (described below) to request images that contain more than 82,000 pixels, then PunkBuster will automatically reduce the image size until the actual capture fits within the limitation. Also, PunkBuster will not allow more than 3 screenshots to be requested from each player within any 10 minute period, nor allow a screenshot to be requested within 30 seconds of the previous request. The PB\_SV\_PLIST command used to display information about connected players has a column called "RecentSS" which shows the number of requests for each player within the last 10 minutes.

Also, Admins should keep in mind that PunkBuster will not take a screenshot from a player who has minimized his local game or who has an application active other than the game itself. In those cases, an empty black screenshot image is sent with text at the bottom explaining why the Capture failed. Also, there are a few video display setups that cannot be correctly captured at this time. These setups also do not allow the game itself to take screenshots using the built-in non-PunkBuster method.

The following settings can be used to customize the Screen Capture Facility:

PB\_SV\_SSWIDTH and PB\_SV\_SSHEIGHT are used to request a specific size image in pixels. The default is 320 x 240. If a size is requested that is larger than the actual resolution on a player's system, then PB will automatically reduce the appropriate parameter to match the size of the resolution. For example, if a screenshot of 800x600 resolution is requested from a player running the game in 640x480 resolution, then PB will take the image at 640x480 resolution automatically.

PB\_SV\_XPCT and PB\_SV\_YPCT are used to specify where on the full playing screen that the capture is to be centered. The default is 50 and 50 so that the capture comes from the center of the playing screen. Using lower numbers moves the captured area toward the left (in the case of XPCT) and upper (in the case of YPCT) portions of the screen and using higher numbers moves the area towards the right and lower portions of the screen respectively. If XPCT and/or YPCT are set to -1, then a random percentage between 0 and 100 will be used for each individual capture request - this has the effect of capturing different (randomized) portions of the playing screen each time a capture is taken.

PB\_SV\_SSSRATE sets the sample rate for the capture. The Sample Rate can be set to take less pixels from a larger area of the screen to, in effect, compact a larger picture into a smaller size by simply skipping pixels. Sample Rates of 1, 2 and 4 are available. Using a Sample Rate of 2 will reduce image file sizes to about one fourth of normal (Sample Rate 1) and using 4 will reduce to about one sixteenth. The reduction in file size does affect the clarity of the image. It is always best to use Sample Rate of 1 unless you desire to capture large portions of the playing screen and don't mind losing some clarity in the resulting image. Even with the 82,000 pixel limit, it is possible to capture images at 1280x1024 using a sample rate of 4 for the relatively few players who play in very high resolutions.

PB\_SV\_SSDELAY causes the PB Client to wait a random number of seconds (up to 60) after receiving the capture request before actually capturing the image.

PB\_SV\_SSPATH can be configured to hold an alternate location where captured screenshots



are to be written. This can be a network share if desired. The default is "" (empty) which means the pb/svss subfolder holds captured screen shots and related support files.

PB\_SV\_SSFLOOR, PB\_SV\_SSCEILING, and PB\_SV\_SSNEXT deal with the way captured screenshot images are named as they are saved to files at the PB server. The PB\_SV\_SSNEXT setting holds the serial number that PB will use for the next captured filename - for example if PB\_SV\_SSNEXT is 250, then the next captured image will have a filename of pb000250.png (and it's "helper" file will be named pb000250.htm). The PB\_SV\_SSNEXT setting is incremented and maintained automatically by the PB system but can be set manually if desired. The FLOOR and CEILING settings specify the from and to range for the screenshot filenames captured by this server. Using the PB\_SV\_SSPATH command mentioned above, Admins who run multiple servers may want to have all PB Servers send captured screenshots to a central location. The FLOOR and CEILING settings allow the admin to give each PB Server its own unique range of filenames to use so that there is never any overlap in the central archive.

PB\_SV\_SSCMD holds the name of a local script (batch) file that PB will execute automatically after each screen shot is retrieved successfully. The default value of this setting is "" (empty) which means that PB will not run a local script at all after image captures. PunkBuster passes the full image filename with path as a parameter to the script. This is for advanced Admins who wish to automate the processing, archiving, and/or publishing of screenshots (such as to a web server).

## PUNKBUSTER PLAYER POWER

This facility is not associated with the votekick system that is built into the game itself. Both systems work independently of each other and can be used together or be enabled/disabled separately depending on the desires of each PB Server Admin.

PB Player Power allows PB Server Admins to empower their trusted regular players with more weight in a new elective process used to remove an undesirable player from the game server. Frequently, troublemakers who attack teammates, block doorways, or otherwise ruin the gameplay are able to get around other methods of keeping them off of the server. This new facility aims to tackle this issue so that gameplay on good public servers doesn't suffer when troublemaking punks show up.

Basically, every player on PB Servers has a power rating. Each PB Server Admin can add specific players to a locally stored database which stores the power rating for those specific players between playing sessions. Additionally, a "catch all" setting called `pb_sv_powerdef` holds the default power rating for players who are not in the database. During gameplay, players may apply their power points against a player by requesting to have that player kicked from the server. Each application of power points is reported to all players on the server so that all players will be able to see when a player applies his/her power points against another player. Once enough power points have accumulated against a given player, then that player is removed from the game and forced to wait a number of minutes before being allowed to rejoin.

To add a specific player to the local PB Player Power database, use the `pb_sv_power` command; this command takes two arguments: the slot # of the player to be added and the desired power rating. The `pb_sv_plist` command is used to display a slot # for all connected players. The local PB Player Power database is stored in the `pbpower.dat` file. Each entry is based on the guid of the player.

The power rating assigned to players may range from 0 to 100. Assigning a power of 0 to a player means that the player's vote carries zero weight on the server and any requests by that player to kick other players using this facility are completely ignored. A setting of 0 should normally be reserved for players who have shown themselves to be untrustworthy from the standpoint of voting against their peers. Assigning a power of 100 to a player is further distinguishing that player as a "deputy". There are a couple of additional points of interest involving deputies. When at least one deputy is presently connected to a game server, then **only** deputies may apply their power points on that server. Other players must appeal to the judgment of any and all deputies presently connected. Also, deputy players cannot be removed from the game server using this facility.

The `pb_sv_powerdef` setting is used to set the number of power points allocated to players who are not in the local PB Player Power database. Use the `pb_sv_powermin` setting to specify the number of power points required to be applied to a given player before that player is removed from the game. The `pb_sv_powerkicklen` setting holds the number of minutes that players must wait before being allowed to rejoin when removed via this facility. To completely disable this facility, set `pb_sv_powerkicklen` to 0.

More information about how this facility is used from a player's perspective can be found in our



related publication: "PunkBuster for Players".

## REMOTE LOGGING FACILITY

Even Balance, Inc. will soon be offering a premium, optional hosting service for Server Admins who wish to remotely log screenshot and log file signatures in real time for automatic publication to a fully and openly accessible website. We will offer a similar service for Players. Each subscriber gets a separate page and each days' activity is further separated into web pages. When this Service is available, more information will be available on our website at [evenbalance.com](http://evenbalance.com).

## FREQUENTLY ASKED QUESTIONS

Frequently Asked Questions (FAQs) are available on our [website](#). Choose the appropriate game from the support page.