PASSWORD RBL

CUSTOM BLACKLIST MANAGEMENT

ADMIN GUIDE

VERSION 2.00

Table of Contents

Introduction

This guide will provide all the information needed to take advantage of the custom blacklisting feature of Password RBL's bad password blacklisting service. This guide applies not matter which way you interact with the Password RBL's API – using Password Firewall for Windows or using Direct API Access.

Background Information

Password RBL Custom Blacklists are managed from anywhere either using direct API calls or the provided PowerShell-based utilities available from the website Downloads section. This guide covers all the options available when using the provided PowerShell utilities. If you are interested in writing your own code to manage your custom blacklists by interacting with our API, please reference our API Guide to better understand the API functions that are available.

The provided PowerShell utilities are written to interact with the Password RBL Custom Blacklist Management API. This utility implements all the API functions and can be run from anywhere with connectivity to our API (not just the systems that use the Source IPs provided to Password RBL during registration). There are no third-party libraries or plugins required. These utilities use all natively provided functionality built in to PowerShell.

Requirements

The PowerShell utilities require Windows PowerShell and should be run from a system with an Operating System of Windows 7 / Server 2008 R2 or later to ensure complete compatibility. When adding passwords to the blacklist, the supplied passwords are interactively hashed 30,000 times on the client system before being sent to the API, so the faster the computer's processor, the faster the program can submit custom entries. However, it is important to note that connections to the webservice API are throttled to an average of 4 API calls per second.

PowerShell Execution Policy

Windows PowerShell has a built-in security model that can deny the ability to run scripts. You may need to update your PowerShell Execution Policy to allow these utilities to run. To do so, utilize the built-in `Set-ExecutionPolicy` cmdlet. Password RBL recommends the policy be set to "RemoteSigned" to gain the benefit of requiring remotely located scripts to be digitally signed, but local scripts to be allowed. An example command is as follows:

```
Set-ExecutionPolicy -RemoteSigned
```

NOTE: You may need to run the PowerShell window with elevated permissions (Run As Administrator) in order to change the system level Execution Policy. This would only be necessary to change the Execution Policy. The Password RBL PowerShell utilities do not require elevated permissions.

Step 1: Create a List of Passwords to Block

Overview

When adding blocked passwords to your custom blacklist, you first need to create a list of passwords you want to add to your custom blacklist. You can do this yourself by placing each individual permutation on a separate line in an unformatted text file, or you can use the provided `prbl-permutationgenerator.ps1` utility to generate permutations for you. There are two modes: Automatic and Manual. Automatic will take an input password and produce a list of many permutations based on common character substitutions. In Manual mode, you specify a list of possible characters for each character position. In either mode, pipe the output to a file (using the built-in `out-file` cmdlet) for use in Step 2.

Usage Information: prbl-PermutationGenerator.ps1

Summary

```
prbl-permutationgenerator.ps1  -Mode [ Automatic | Manual ]                -
InputString [ 'password' | 'pP,a@,sS$,sS$,w,o0,r,d' ]
```

Required Arguments

InputString

The input string is used in both modes and should be a single-quoted string to ensure any special characters are used literally and not interpreted by PowerShell. The Format of this string is determined by the Mode.

Mode

The mode argument choses between how this script produces the permutations. The options are Automatic and Manual.

Automatic

This InputString is just a single permutation of a password that will be expanded to many permutations using upper and lower case versions of each character as well as common character substitutions, such as '3' for 'e' and '7' for 'T'. For example, to perform automatic permutation creation for the word, "secure", the InputString format would be:

```
-InputString 'secure'
```

Manual

In Manual mode, the InputString specifies the possible characters for each position in the password permutation. The syntax is a single quoted string value with the characters options for each position comma-delimited. For example, to form permutations of the word "Secure", you would provide:

```
-InputString 'sS,eE3,cC,uU,rR,eE3'
```

Examples

Below are examples of using this utility to create permutations that will later be used to inject into a specified custom blacklist.

IMPORTANT: It is recommended to "pipe" the output of this Powershell script to the "out-file" cmdlet to automatically place each generated permutation into the file.

Automatic Expansion of password "Secure"

```
PS C:\> prbl-permutationgenerator.ps1 -Mode Automatic -InputString 'secure' | out-file
outputfile.txt
```

Manual permutations of "Company"

```
PS C:\> prbl-permutationgenerator.ps1 -Mode Manual -InputString 'cC,o0,m,p,a@,n,yY' |
out-file outputfile.txt
```

Step 2: Add Permutations to your Custom Blacklist

Overview

Once you have a list of all the password permutations you would like to block, you can use the `prbl-cblmanagement.ps1` utility to add these permutations into your custom blacklist. Because hashing of these passwords is done on your system before being sent to the API, you should expect this process to take some time. A progress meter is printed in the console during the process.

Usage Information: prbl-cblmanagement.ps1

Summary

```
prbl-cblmanagement.ps1  -Mode [Quota | Count | Add | Delete | Empty]  -BlacklistID
[32 hex characters]  -InputFile [Path]  -HashType [ PBKDF2 | SHA256 | BOTH ]
```

Required Arguments

Mode

The mode is the verb that dictates what `prbl-cblmanagement.ps1` will do during this execution. The options are Quota, Count, Add, Delete, or Empty.

Quota

This mode will connect to the API and return the maximum number of allowed custom blacklist entries for the specified BlacklistID.

Count

This mode will connect to the API and return the current number of custom blacklist entries.

Note: Password RBL supports multiple hashing algorithms. Currently, PBKDF2 and SHA256 are supported, but others may be added in the future. You only need to add hashes of the passwords in your input file using the algorithm that you plan to use when you query the API during password change events. However, it is not detrimental to add both types. For ease of use, Password RBL's recommendation and this utility's default setting is to add passwords hashed with both algorithms. The Count method will return maximum number of custom blacklist entries across each hash type.

Add

The Add action uses the provided input file, hashes each password, and injects each resulting hash value into the custom blacklist, identified by the BlacklistID.

Delete

The Delete action uses the provided input file, hashes each password, and removes each resulting hash value from the custom blacklist, identified by the BlacklistID.

Empty

The Empty action deletes all entries (of any hash type) from the custom blacklist in a single command.

BlacklistID

The BlacklistID is always required and identifies what custom blacklist will be manipulated by the utility. The ID is a 32 character hexadecimal value assigned to you by Password RBL.

Path

The Path argument is only required when the mode is either Add or Delete. This argument contains a path to an unformatted text file of cleartext passwords to either Add or Delete from the custom blacklist identified by the accompanied BlacklistID argument. This file is the output file from Step 1, or alternatively, a file of

password permutations created manually. See the appendix for suggestions on making a good list of custom passwords to block.

Optional Arguments

HashType

This argument specifies which hashing algorithm is used when hashing the permuations provide in the input file. Password RBL currently supports two algorithms: PBKDF2 and SHA256. By default, this utility will produce both hashes and upload them into your custom blacklist.

Example Commands

Below are examples of using this utility to affect a custom blacklist. The BlacklistID shown below is for illustration purposes only and does not represent a valid BlacklistID.

Adding entries to the blacklist

```
PS C:\> prbl-cblmanagment.ps1  -Mode ADD  -BlacklistID "1234567890ABCDEF1234567890ABCDEF"
-InputFile C:\pwds4prbl.txt
```

Checking a blacklist's quota

```
PS C:\> prbl-cblmanagment.ps1 -Mode QUOTA -BlacklistID "1234567890ABCDEF1234567890ABCDEF"
```

Check how many entries are in the blacklist

```
PS C:\> prbl-cblmanagment.ps1 -Mode COUNT -BlacklistID "1234567890ABCDEF1234567890ABCDEF"
```

Remove entries from the custom blacklist

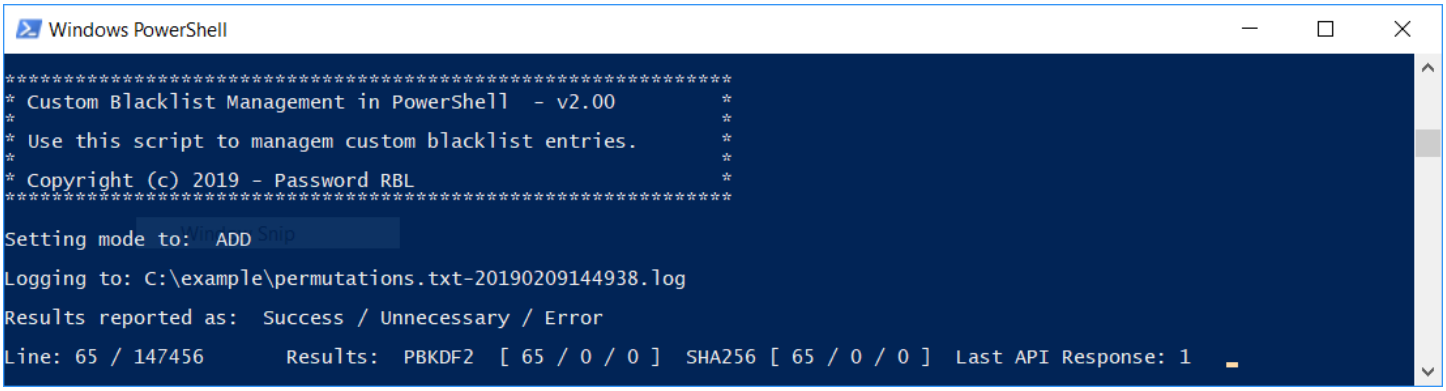
```
PS C:\> prbl-cblmanagment.ps1  -BlacklistID "1234567890ABCDEF1234567890ABCDEF"  -Mode
DELETE  -InputFile C:\pwds2del.txt
```

Remove all entries from the custom blacklist

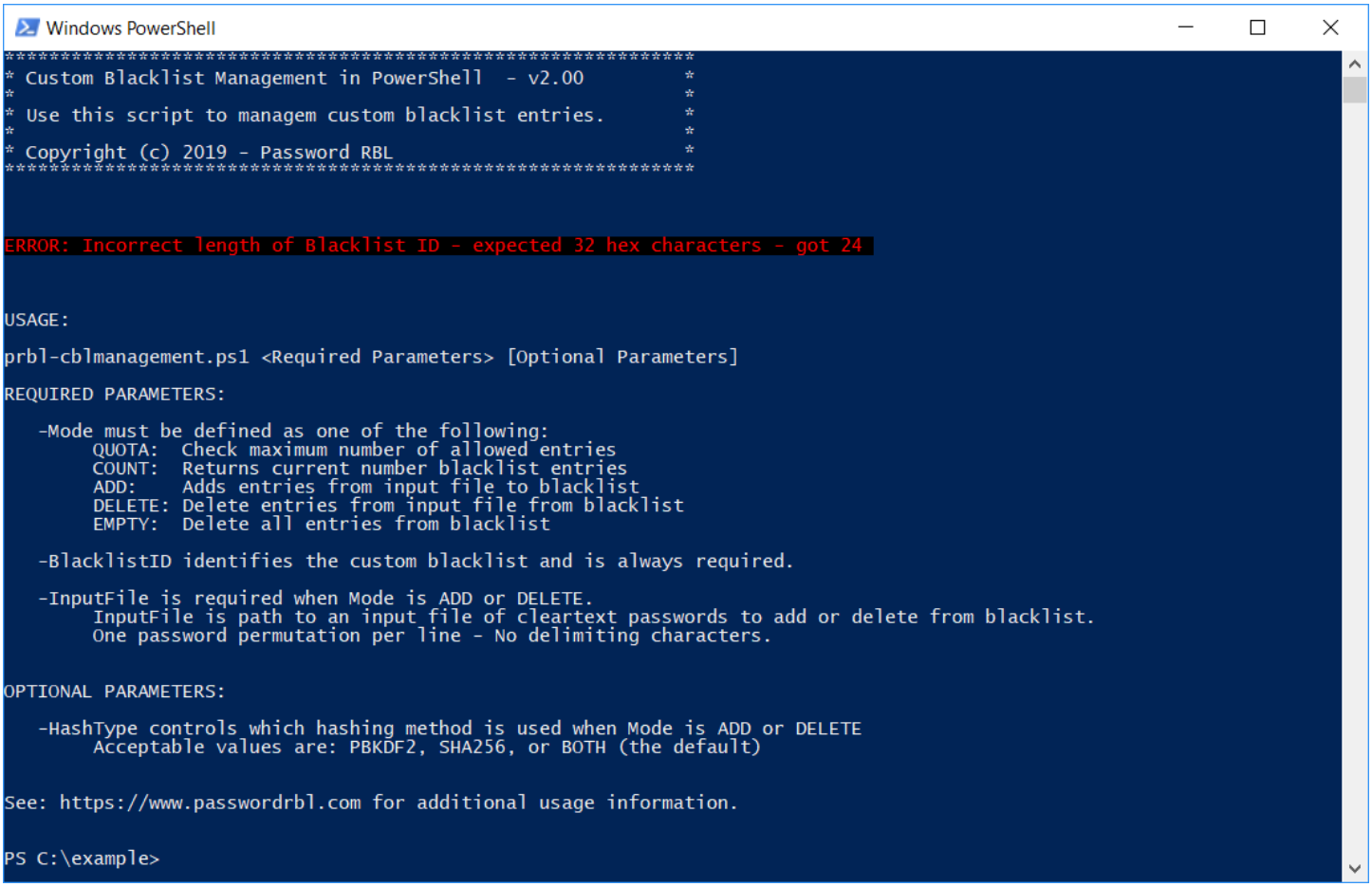
```
PS C:\> prbl-cblmanagment.ps1 -Mode EMPTY -BlacklistID "1234567890ABCDEF1234567890ABCDEF"
```

Example Output

Good Results



Bad Results



Status Line

While the `prbl-cblmanagement.ps1` script runs, it will print a current runtime status in the console window as seen in the pictures above.

Modes Quota, Count and Empty finish quickly and will only print a final output that confirms the action or shows an error message if an error was returned from the API.

When the mode is Add or Delete, hashing is involved and the status line will report the hashing activity and last API response code. The hashing status is shown for both algorithms (PBKDF2 and SHA256) and reports 3 numbers: Success, Unnecessary, and Error. Successes report API calls that completed as expected. Unnecessary means the submitted hash value was already in the list (when mode is Add) or the submitted hash value is not in the list (when mode is Delete). Errors report how often the API returns an error. If an error is returned, see the specific error code at the end of the status line. The most common errors are: connectivity issues and exceeding quota. Additional output is available in the log file (see below).

Logging

When the Mode is Add or Delete, the `prbl-cblmanagement.ps1` script logs each individual API call to a file to aid in troubleshooting in case of errors. This file will attempt to be put in the exact same folder as the input file. If this fails, then the script places this file in the home folder of the user running the script. The name of the log file will be the same name as the input file with the current date and time appended to the end of the file name. The path chosen will be printed in the console window.

Additional Runs

You can add additional passwords to your custom blacklist at any time as long as the total number of passwords added is less than your allocated quota. To speed up the process, use a new input file that only includes the new permutations to add. If you use the same input file, the status line will show Unnecessary counts for all permutations that were already in your custom blacklist.

Step 3: Use your Custom Blacklist

Password Firewall for Windows

Now that you have populated your custom blacklist with permutations you want to block, add your Custom BlacklistID to the configuration of Password Firewall for Windows to start blocking these permutations immediately. You can either re-run the Setup for Password Firewall and enter your BlacklistID when prompted by the installation wizard, or edit the passwdfw.ini file directly and add the BlacklistID to the appropriate line item. See the Password Firewall for Windows Admin Guide for further information.

Direct API Access

In order to query your custom blacklist when implementing the Password RBL API, you simply need to add your custom blacklist as a parameter in the HTTPS GET request. See the API Guide for more information.

Appendix: Tips for Custom Blacklist entries

Below you will find some tips for what types of passwords to add into your custom blacklist. These tips are based on work done by many individuals and academics, both in directed studies as well as analysis of real credential data following public data breaches.

Each Entry is [Case] Specific

Password RBL's database of bad passwords enumerates all the permutations of known bad passwords. The custom blacklist functionality works exactly the same way. This means you must add all permutations of a password you want to block. For example, you need to enter both: `example123` and `Example123`.

Incrementing Numbers Follow Base Password

It is common that end-users will add an increasing number to the end of their unchanging "base" password. This practice stems from a system or company's password policy that requires end-user password changes on a set frequency (monthly, quarterly, etc.). When this happens, end-users tend to pick one password, and just change the number at the end of their chosen password. You may want to add permutations of commonly use passwords at your company with serial numbers appended to the end. For example, `CompanyPassword1`, `CompanyPassword2`, `CompanyPassword3`, etc., etc.

IMPORTANT: Password Firewall for Windows v5.0 introduced a feature named "DoubleCheck" that will truncate common suffix characters from passwords, such as numbers, and re-query the blacklists for existence of this shorter password (if the original password was not found in the blacklists). Enabling this feature in Password Firewall means you do not have to add custom blacklist permutations that end in numbers.

Company Name and other Public Data

Many employees will choose passwords that are associated with the place of work. This helps them remember the password, but unfortunately, much of this data is publically available. You probably want to block the use of these passwords, and maybe also block them with appended numbers at the end. Common choices are:

- The company name
- The company's address or the address of the site where they work
- The company's phone number
- Company motto or slogan

Previously Used Passwords

Another good set of passwords to add to your custom password blacklist would be any previously used shared passwords. This is especially true if these passwords were used with accounts that have elevated permissions. Such practice is common amongst IT departments as IT workers have significantly higher number of credentials to remember compared to the typical employee. Adding these shared [admin] passwords to your custom password blacklist is especially important when an IT worker leaves the company.

You can then enforce that a password change happen and know that accounts are not using a password that a [now] non-employee knows and could easily guess to gain unauthorized access after they leave the company.

Custom Blacklist Admin Guide Version History

Version	Notable Changes
2.00	Update Permutation Generator to accept any length Add automatic permutation generation option Convert scripts to native PowerShell style parameters
1.10	Add simple Permutation Generator
1.02	Maintenance release, non-technical updates
1.01	Maintenance release, non-technical updates
1.00	Original Release