

AUTHENTICATION TESTER

POWERTOOLS

Watchfire Authentication Tester PowerTool
User Guide

Copyright ©1996-2006 Watchfire Corporation. All rights reserved.

No part of this manual may be reproduced in any form or by any means, without permission in writing from Watchfire Corporation.

Watchfire Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

Watchfire, AppScan, and the Flame Logo are trademarks or registered trademarks of Watchfire Corporation.

All other company or product names are trademarks or registered trademarks of their respective owners.

Spell Checking Oriented Word Lists (SCOWL) © 2000-2004 by Kevin Atkinson

Permission to use, copy, modify, distribute and sell these word lists, the associated scripts, the output created from the scripts, and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Kevin Atkinson makes no representations about the suitability of this array for any purpose. It is provided "as is" without express or implied warranty.

Contents

Authentication Tester	5
Authentication Methods	5
Form Authentication	5
Provide a Typical Login	5
Describe the Login Responses	7
Metacharacters General Information	8
HTTP Authentication	8
Running the Authentication Tests	9
Understanding Scan Results	10
Advanced Configurations	10
General Tab	11
Form Authentication Tab	12
Proxy Tab	13
Credential Generation Tab	14
Credential Generation Mode	14
Credential Generation Configuration	14

Authentication Tester

The Watchfire Authentication Tester Power Tool is a brute-force technique testing utility. It finds weak username-password combinations that could be used to gain access to your web application.

A brute force attack is an automated process of trial and error used to guess authentication credentials, causing a server to acknowledge an imposter as a legitimate user.

Using brute force, a malicious user will cycle through combinations until stumbling upon credentials that gains access to the authorized area. The malicious user, with a brute force application, can employ a dictionary file or even try all possible combinations of the accepted character set (according to the site's given format of what is valid input for a username and password).

A brute force attack can generate thousands, possibly millions, of incorrect combinations before succeeding to gain access. While brute force methods often succeed, they can take hours, weeks, or longer before finding successful credentials. If your web application enforces the use of strong passwords, you can make the brute force attack unfeasible.

Authentication Methods

Authentication Tester runs brute force tests assuming one of two authentication methods:

- **Form** - authentication is performed by a custom web-page.
- **HTTP** - authentication as defined in the protocol.

⇒ In the Authentication Tester main window, choose the authentication method used by your web application:

- **Form Authentication**
- **HTTP Authentication**

The available options for setting up the scan with the selected method change according to your selection.

Form Authentication

If you select the **Form Authentication** radio button, perform the following procedures:

- 1 Provide a Typical Login.
- 2 Describe the Login Responses.

Provide a Typical Login

If you select **Form Authentication** in the **Authentication Method** section of the main window, the **Setup** button is visible.

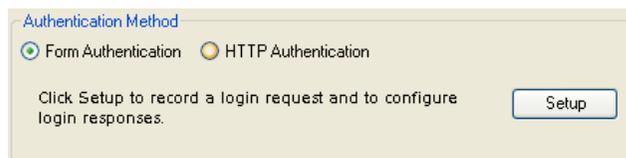


Figure 1 Authentication Method Options, Form Authentication Selected

To configure login:

- 1 Click **Setup**.
A browser opens.

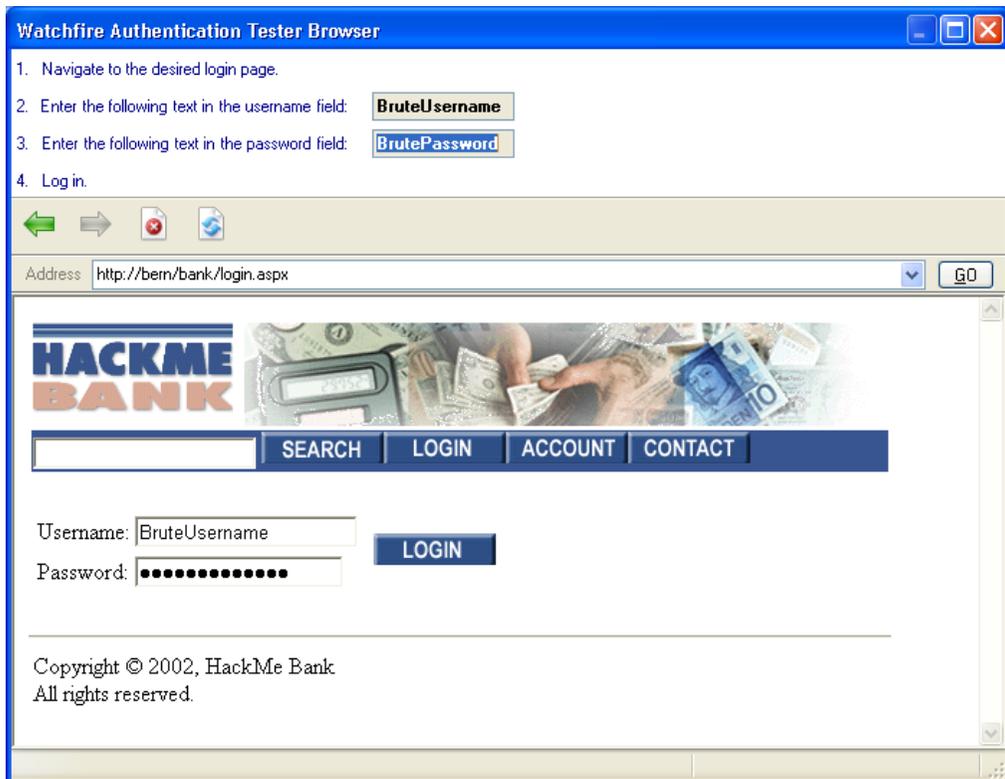


Figure 2 Authentication Tester Browser

- 2 Browse to the login page of your web application.
- 3 Log in with:

username: BruteUsername
password: BrutePassword

Authentication Tester requires that you model login of the site using these values. During the tests, the username and password strings will be replaced with the possible combinations. **If these values are not the strings that Authentication Tester is searching for, the brute force test will not operate correctly.**



You can change the placeholder values of the username and password strings. See “Form Authentication Tab” (below).

When you have successfully logged in, Authentication Tester closes the browser and opens the **Successful Login Detection** window.

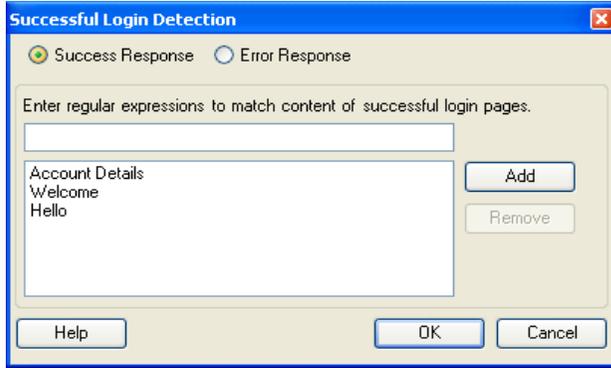


Figure 3 Successful Login Detection Window

Describe the Login Responses

In the Successful Login Detection window, you enable Authentication Tester to recognize login requests as either successful or failed. This information is necessary to know when the web application has accepted credentials as valid.

To describe login responses:

- 1 Select the type of response that you want to describe:

Success Response	describe the page content of a valid login response
Error Response	describe the page content of a response to an invalid login attempt

You may describe each type of responses. For example, if you know that invalid credentials often receive a response of: "Username and password do not match" you can use this to let Authentication Tester know the result of its tests.

- 2 Enter a regular expression that matches some content on the response page.

Be sure to match only static content, not variables.

The use of regular expressions, rather than strings, allows you to configure Authentication Tester once for multiple runs during the development stages of your web application.

For example, if the design of the successful-login page has not yet been finalized between a large, full page Welcome! note or a small welcome string at the top of the home page, you can enter

```
(?i)welcome
```

to indicate that the search word is case-insensitive.



To learn more about regular expressions, and the metacharacters that can be used to indicate more than literal strings, see "Metacharacters General Information". To test regular expressions before attempting to use them in Authentication Tester, try the **Watchfire Expression Test Power Tool**.

- 3 Click **Add**.

You can add as many regular expressions as you want. Authentication Tester uses them with an OR operator: if one or more of the regular expressions matches content on a page of your site, that page is recognized as a result page (either successful login or error page, depending on your response type selection).

- 4 Click **OK**.

The **Successful Login Detection** window closes and you are returned to the main window.

To continue:

From this point, choose one of the following operations to continue.

To learn how to:	See:
Run brute force tests using the current configuration	“Running the Authentication Tests” on page -9
Write regular expressions	“Metacharacters General Information” (below)
Set up brute force tests using HTTP authentication	“HTTP Authentication” on page -8

Metacharacters General Information

Any single character (letter, digit, or symbol) in a regular expression is matched to itself, literally; unless it is a metacharacter. A metacharacter is one or more characters that have a unique meaning and are not used literally in the regular expression match.

For example, the circumflex character (^) is a metacharacter that means “search at the beginning”. If you want to find the circumflex character, rather than the metacharacter pattern, protect it with a backslash: \^.

Metacharacter	Description	Example
\	Protect the next character.	\! finds a exclamation point (!) \. finds a period (.), rather than a character
^	Find at the beginning of a string.	^H finds “ H ome”, but not: “ this page”
.	Find any character (letter, number, symbol, whitespace), except newline.	(.*) finds any paragraph
()	Find a pattern group.	(word) finds “In this word ” ^(word) finds “ W ords in this line” Welcome ((back) (home)) finds “ W elcome back ” and “ W elcome home ”
[]	Find a pattern range.	[a-z] finds any lower-case alphabetic character
*	Find the pattern zero or more times.	<(.*> finds all HTML tags, with their content
+	Find the pattern one or more times.	<(.*>+ finds < UL >< OL >
?	Find the pattern zero or one time.	log(.?)in finds login and log in
(?i)	Find the next characters with a case-insensitive search.	(?i)word finds word , Word , woRd , WORD

Table 3-1 Common Regular Expression Metacharacters

HTTP Authentication

If you select the **HTTP Authentication** radio button in the **Authentication Method** section, the **URL Address**, with the **Show** button, and **Domain** text boxes are visible.

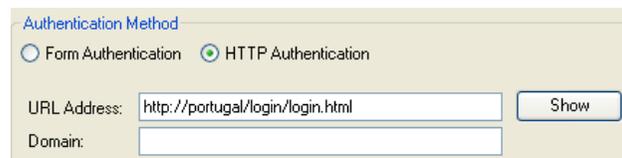


Figure 4 Authentication Method Options, HTTP Authentication Selected

To set up HTTP authentication tests:

⇒ Enter the URL of the login page in the **URL Address** text-box.

To test that this is the correct URL, click **Show**. A browser appears.

- If a standard HTTP login window appears over the browser, the URL is correct.
- If the page appears without an HTTP login window, the URL is incorrect. Try another URL.

If the HTTP login window requires a domain, enter the correct domain name in the **Domain** text box of Authentication Tester.

To continue:

From this point, choose one of the following operations to continue.

To learn how to:	See:
Run brute force tests using the current configuration	“Running the Authentication Tests” on page -9
Set up brute force tests using Form authentication	“Form Authentication” on page -5

Running the Authentication Tests

After you have chosen the authentication method and configured the basic tests, you can start the Authentication Tester scan.

To start an Authentication Tester scan:

⇒ Click **Start**.

The screenshot shows the 'Scan' window in Authentication Tester. At the top, there are 'Start', 'Pause', and 'Stop' buttons, and an 'Advanced' button. Below these is a section for 'Successful logins' with columns for 'Username' and 'Password'. At the bottom, there is a progress bar and a status indicator showing 'Scan: 552 / 720 (5 threads)'. Annotations point to various elements: 'Start is enabled after you have set up the selected method. It is disabled after the scan begins.' points to the 'Start' button; 'number credentials already tested, against total credentials to test' points to the progress bar; 'number threads used in this scan' points to the '(5 threads)' part of the status indicator; and 'if not in progress, status is also given: *paused *resumed *stopped *finished' points to the status indicator.

To pause a scan:

⇒ Click **Pause**.

If you leave Authentication Tester open (do not exit), you can resume the scan: click **Resume**. If you exit Authentication Tester after pausing, the scan is deleted.

To stop a scan:

⇒ Click **Stop**.

You cannot resume a scan after stopping it; **Stop** will stop the scan and delete the data collected so far.

Understanding Scan Results

After a scan has finished all the tests that Authentication Tester was configured to run, any username-password pair that gained a successful login is listed in the table.

These results show you exactly which usernames-passwords can be guessed with brute force methods; but more than that, they provide you with general information about the vulnerabilities of your web application.

For example, the results can help you determine the best length or length range of passwords, and the best charset:

Enter a username (should include both upper- and lower-case letters).

Enter a password (from 6 to 12 characters, containing letters and numbers).

To save results:

- 1 Click **Export results**.
- 2 Name the export file.

An XML file is created, listing the credentials that successfully gained access. You can save as many result files as you need.

This feature is convenient for authentication testing during development, as you can compare the Authentication Tester results after implementing fixes.

Advanced Configurations

You can customize the behavior of Authentication Tester for your local network and the application you are testing.

- 1 In the main window, click **Advanced**.

The **Advanced Configuration** window opens.

- 2 Make any changes you want.
- 3 Click **OK**.

The changes are applied in the next scan.

Tabs of the **Advanced Configuration** window include:

- General Tab
- Form Authentication Tab
- Proxy Tab
- Credential Generation Tab

General Tab

The **General** tab of the **Advanced Configuration** window allows you to change network configurations.

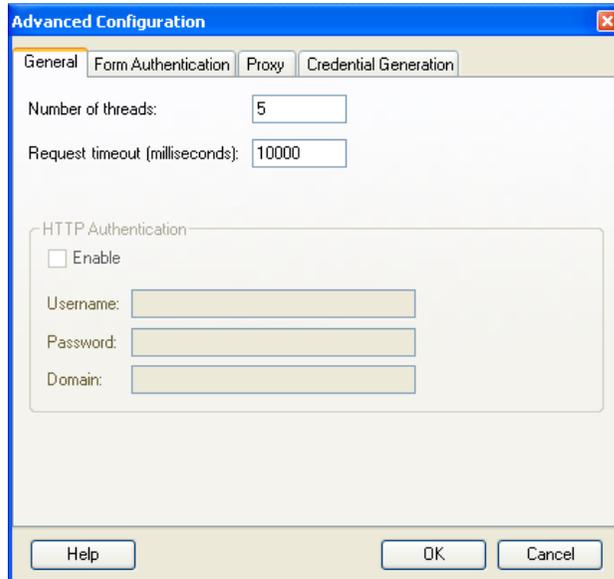


Figure 5 Advanced Configuration - “General Tab”

Option	Description
Number of threads	Set the number of connection threads Authentication Tester will use during the scan.
Request timeout (milliseconds)	Set the amount of time Authentication Tester should be given to reach the web application server before timing out the connection.

Table 5-2 General Tab - Network Configuration Options

The General tab also allows you to configure Authentication Tester for applications with multiple levels of authentication.

If the web application uses HTTP authentication on top of Form Authentication, enter valid values for the **username**, **password**, and **domain**. This will allow Authentication Tester to access the web application server to run the brute force tests on the Form Authentication mechanism.

These values do not affect the HTTP authentication that is tested in the Authentication Tester scan; they are relevant only if you select the **Form Authentication** method.

Form Authentication Tab

This tab holds the Successful Login regular expressions that you enter to describe the page that is sent in response to an accepted username-password.

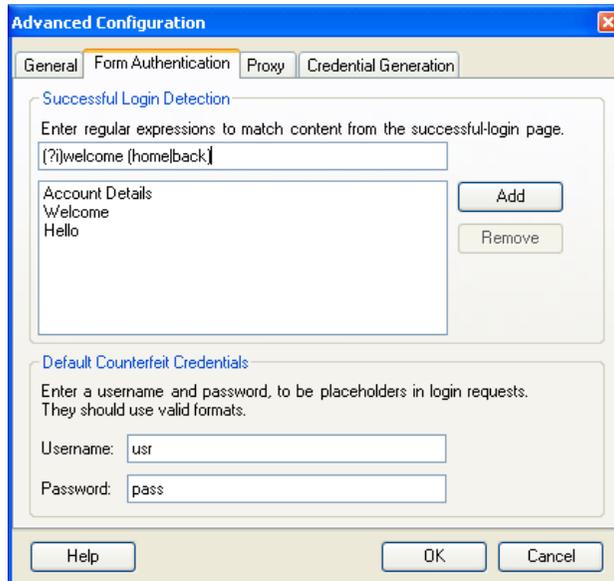


Figure 6 Advanced Configuration - “Form Authentication Tab”

See “Describe the Login Responses” on page -7 for details; or “Metacharacters General Information” on page -8 to learn more about writing regular expressions.

The options of the bottom section of this tab enable you to change the default username and password needed to log in with when configuring Authentication Tester for Form Authentication.

When you select **Form Authentication** mode, you log into the web application with a specific username and password (see “Provide a Typical Login” on page -5). These values may be valid or invalid for the application, but they must be configured in Authentication Tester as Form Authentication test values.

If you want to change these values from their default, enter a new username and password in the text-boxes of the **Default Counterfeit Credentials** section of the **Form Authentication** tab.

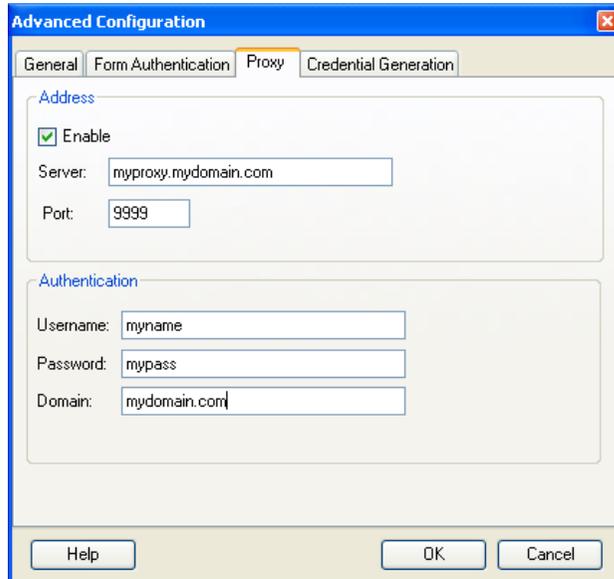


The values should follow the format that is valid for the application. For example, if the application requires an email as the username, enter a username in the email format. Be careful that neither value is a sub-string of the other value. For example, if you enter `user@email.com` as the username, you cannot use `user` as the password.

Whatever values you enter in these text-boxes are the same values that you must use to log into the web application when you are configuring a Form Authentication scan.

Proxy Tab

If your web application requires a web proxy connection, select the **Enable** checkbox on the **Proxy** tab and fill in the appropriate information.



The image shows a screenshot of the "Advanced Configuration" dialog box, specifically the "Proxy" tab. The dialog has four tabs: "General", "Form Authentication", "Proxy", and "Credential Generation". The "Proxy" tab is selected and highlighted. Under the "Address" section, the "Enable" checkbox is checked. The "Server" field contains "myproxy.mydomain.com" and the "Port" field contains "9999". Under the "Authentication" section, the "Username" field contains "myname", the "Password" field contains "mypass", and the "Domain" field contains "mydomain.com". At the bottom of the dialog are three buttons: "Help", "OK", and "Cancel".

Figure 7 Advanced Configuration - "Proxy Tab"

Credential Generation Tab

The options of this tab enable you to configure the usernames and passwords that Authentication Tester tries during the scan.

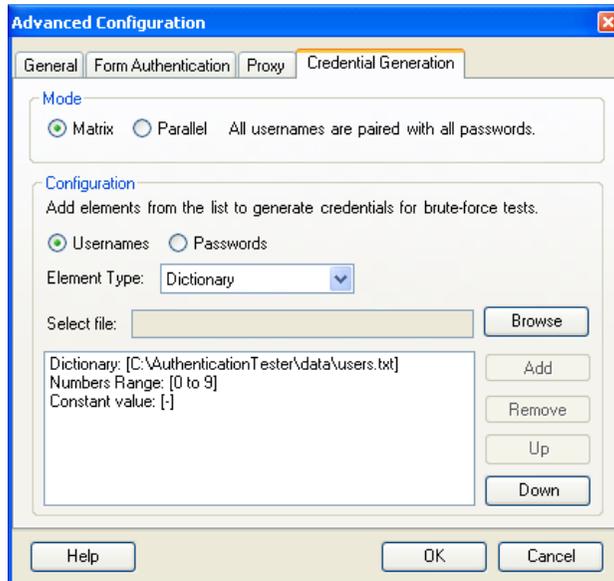


Figure 8 Advanced Configuration - Credential Generation Tab

Credential Generation Mode

Select the mode of the username-password brute force tests that Authentication Tester will run.

Mode	Description
Matrix	Each username in the list is tried against every password in the list. Select this mode for a more comprehensive scan.
Parallel	Usernames and passwords are paired by index number. Select this mode for a faster scan.

Table 8-3 Credential Generation Mode

Credential Generation Configuration

Configure the options that Authentication Tester uses to generate the usernames and/or the passwords to be tested.

- 1 Select one of the radio buttons to customize **Usernames** or **Passwords**.
All fields and data displayed in the Configuration area now apply to the selected item. For each item the drop-down list lets you configure the structure of the usernames/passwords that will be created and used in the attack.
- 2 From the drop-down list, select an element type that you want to be included in the attack, and fill in values for the element as described in the table below.

Element	Description	Valid Values
Dictionary	A “value-per-line” file used when generating usernames/passwords for the attack. Default files are provided for both Usernames and Passwords, but you can use the Browse button to browse to any other suitable file. The \data folder also includes a much larger password dictionary file (passwords_long.txt) that can be used instead of the default file.	pathname to “value-per-line” dictionary file
Numeric	A range for numbers that will be included in the usernames/passwords generated	0 - 2147483647
Constant	A string that you want included in all usernames/passwords that are generated	any string
Character Range	A range of characters and string length, for characters that will be included in the usernames/passwords that are generated	[space to tilde] - ~ and a string length

Table 8-4 List Types



The **space to tilde** range includes a-z, A-Z, 0-9, and ASCII input symbols. If you enter “ -~” (without the quotes) as a Character Range, Authentication Tester automatically inserts all included characters in the regular expression listing.

The valid value of the **length** field depends on the range. For example, if the range is 0-9 and the length is 10, the range is valid [0000000000, 0000000001, ...9999999999]; but if the range is a-z, Authentication Tester will not accept 10 as a valid length because the number of combinations would take an unreasonable amount of time and resources.

Also note: if you want to use a dash as a character, rather than to denote a range, use the backslash (\-).

- Repeat for additional elements as necessary.
- Arrange the element types (using **Up** and **Down**) so that each username or password that is tried will be built of each type, in the order listed.

Credential Generation	Resulting Credentials
Dictionary: C:\web tests\data\users.txt	user_0
Constant value: _	user_1
Numbers range: 0 to 9	user_2
	...
	user_9
Constant value: passwd	passwdaaa
Character Range: a-zA-Z0-9; length: 3	passwdaab
	...
	passwd999
Constant value: iamgod	iamgod 1900
Character Range: -~; length: 1	iamgod!1900
Numbers range: 1900 to 3000	...
	iamgod~3000

Table 8-5 Examples of Credential Generation Configuration