watchfire®

Token Analyzer

PowerTools

Watchfire Token Analyzer PowerTool
User Guide

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

Contents

Watchfire Token Analyzer PowerTool

Watchfire Token Analyzer provides various tests for your web application session tokens, to determine how well secured your application is against session theft. These tests, based on mathematical algorithms, show the strengths and weaknesses of your session tokens.

What Are Session Tokens?

Web application servers respond to client requests without linking multiple requests to any one client, according to the design of HTTP as a stateless protocol. However, the security of your web application depends on your system being able to distinguish between users and to recognize users and their permissions.

Session tokens are transmitted between the server and the client, to provide the concept of "state" to HTTP. They enable the server to respond to any one user as though in a session, throughout their activity (requests and responses) on the URLs of this web application.

Session tokens may be:

- Cookies
- URLs
- Post Data

Cookies

Typically, a cookie contains a domain name, which should ensure that the clients receive your cookies only when in your web application and that they can send your cookies only to your server. A server on another domain should not be able to receive or read the cookies of your web application.

Cookies also contain a name and a value. This pair is used to identify the client as a specific user with certain permissions on your site.

It is important to note that cookies were never designed to store sensitive information, such as usernames or passwords.

URLs

A URL can contain a parameter with the name and value information of a session token.

After the customer logs in, the application server assigns a session token to the customer and replaces variables in the linked URLs with random token IDs. The server keeps track of the URL tokens and pairs them with the customer's session token. When the customer's browser sends a request for a URL, it includes the URL session token.

For example, a link to
http://www.bank.com/account.jsp?**account=10&type=today**
will be changed to
http://www.bank.com/account.jsp?**token=1234**

Post Data

Post data information is sent in the body of an HTTP request. A customer's browser can send session token name-value pairs embedded into a request.

When Do Session Tokens Cause Security Issues?

If a malicious user is able to predict valid values of cookies or URL parameters, this user could enter the web application under the appearance of a legitimate customer: to steal a session. Session theft allows the malicious user to access information and permissions provided for a customer during a valid session. Session tokens become a security issue when their values are predictable.

In another situation, a web application cookie or URL contains user information, such as a valid username or password. This is especially true with URL session tokens that contain readable session token parameters. If a malicious user is able to collect the tokens themselves, either from system invasion or network interception, this user would have enough information to steal legitimate customer sessions.

How Does Token Analyzer Help?

You secure your web application by ensuring that session token values are distributed in a random fashion (that they are as unpredictable as possible) and that they do not contain readable strings that a malicious user could easily identify as session information (such as: "user"; "pass"; "token", "ID"). Your web application contains algorithms that provide the values. You can check these algorithms, without knowing the code or underlying algorithm, by using Watchfire Token Analyzer. If Token Analyzer finds that a session token is a security risk, you can take the necessary steps to fix the issue.

Token Analyzer assumes that a malicious user could have chosen your web application without any prior knowledge of your token-creation algorithm. In this way, it finds any level of predictability. It also means that the tests you can run are low-level, and you might find the results to be complex at first. In following sections, you will learn how to run different tests and why. Then you will learn how to use the results to analyze the security of your web application.

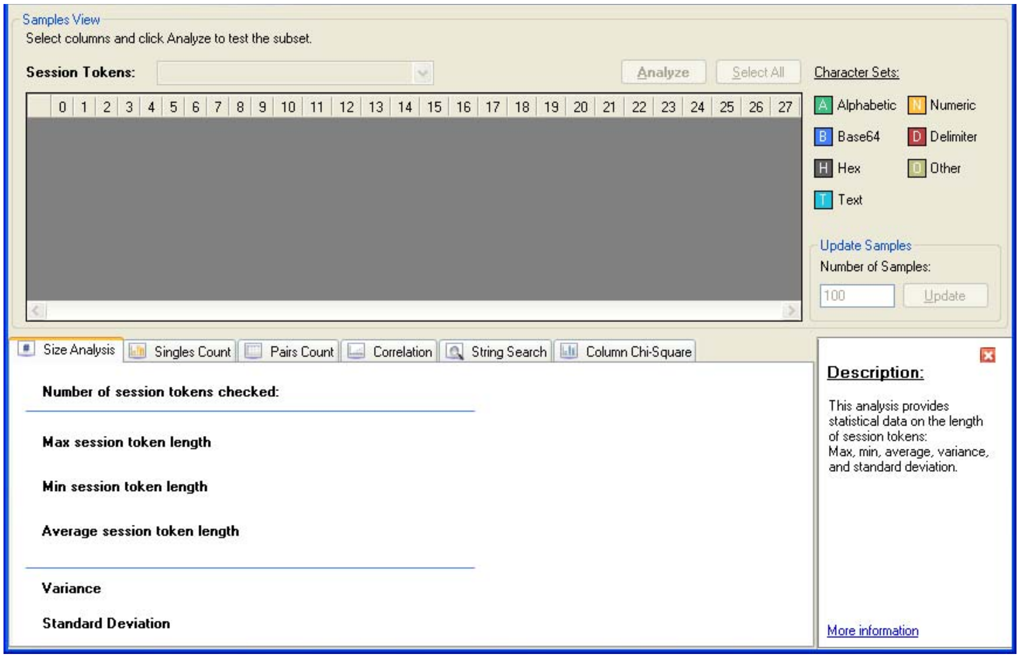
Token Analyzer Functionality

The use of Token Analyzer to analyze your web application session tokens must be interactive to ensure that the collection of data and the results that you receive are relevant for your application. The following procedure shows what Token Analyzer does automatically and where you step in to provide interactive information or analysis requests.

1. You open a new session by browsing to your web application's URL in the embedded web browser.
2. As you browse through your web application's site, Token Analyzer gathers request information.
3. Token Analyzer shows the session tokens that were sent and received in the requests of your browsing. Using this information, you can select the request that set a particular session token.
4. You specify the amount of session token samples that Token Analyzer is to collect.
5. Token Analyzer repeats the selected request until it has collected the specified number of samples.
6. Token Analyzer presents the samples to you, separating and indexing each character.
7. You select the complete sample or a subset of columns, and initiate analysis on the selection.
8. Token Analyzer analyzes the selection.
9. You view the results of the analysis and, with suggestions from Token Analyzer, evaluate the predictability of your session tokens.

Getting Started with Token Analyzer

1 Open the Watchfire Token Analyzer application. The main window appears.



Select columns and click Analyze to test the subset.

Number of Samples: 100

Number of session tokens checked:

Max session token length

Min session token length

Average session token length

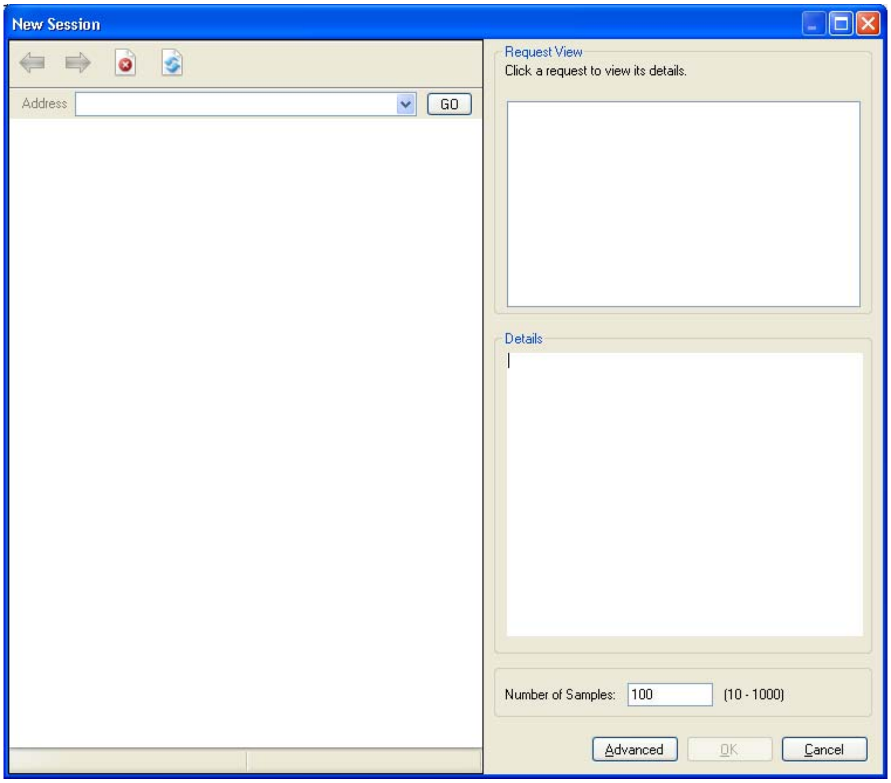
Variance

Standard Deviation

This analysis provides statistical data on the length of session tokens: Max, min, average, variance, and standard deviation.

2 Click **New**.

The **New Session** window appears.



Click a request to view its details.

Number of Samples: 100 (10 - 1000)

Starting a New Analysis

The **New Session** window is the starting point for Token Analyzer to collect session tokens and analyze your session security.

Perform the following tasks:

1 Configure Token Analyzer

2 Explore Your Web Application

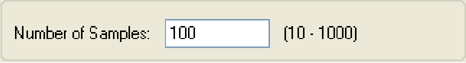
3 Select Relevant Requests

4 Collect Session Token Samples

Configure Token Analyzer

There are a few configurations you should set before starting a new analysis.

- **Samples** - Decide how well or how quickly you want Token Analyzer to evaluate your session tokens.



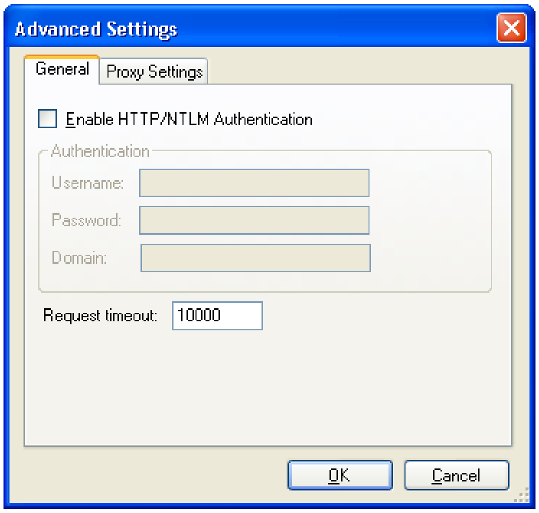
Number of Samples: 100 (10 - 1000)

Token Analyzer re-performs the actions that you modeled for as many times as needed to collect a specific number of samples. By default, this number is 100. You can change this number to any integer between 10 and 1000.

  - If 100 is good enough for your environment, you can skip this step.
  - If you want a greater accuracy in the tests, increase the value in the **Number of Samples** text field.
  - If you want Token Analyzer to work faster, decrease the value of **Number of Samples.**

- **Site Authentication** - If your web application requires HTTP or NTLM authentication, configure Token Analyzer to supply this authentication during its sample collecting.

  Click **Advanced**. In the **Advanced Settings** window, fill in the appropriate information. Consult with your system administrator if necessary.



☐ Enable HTTP/NTLM Authentication

Authentication

Username:

Password:

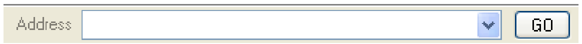
Domain:

Request timeout: 10000

Explore Your Web Application

In the embedded browser of the **New Session** window, browse through your web application as a user would. Token Analyzer will re-perform these browsing actions to collect a wide sample of session tokens.

1 In the **Address** text box of the embedded browser, enter a URL of your web application.



This URL could be your home page, the login page, or any URL where a customer could enter the application or be redirected to it.

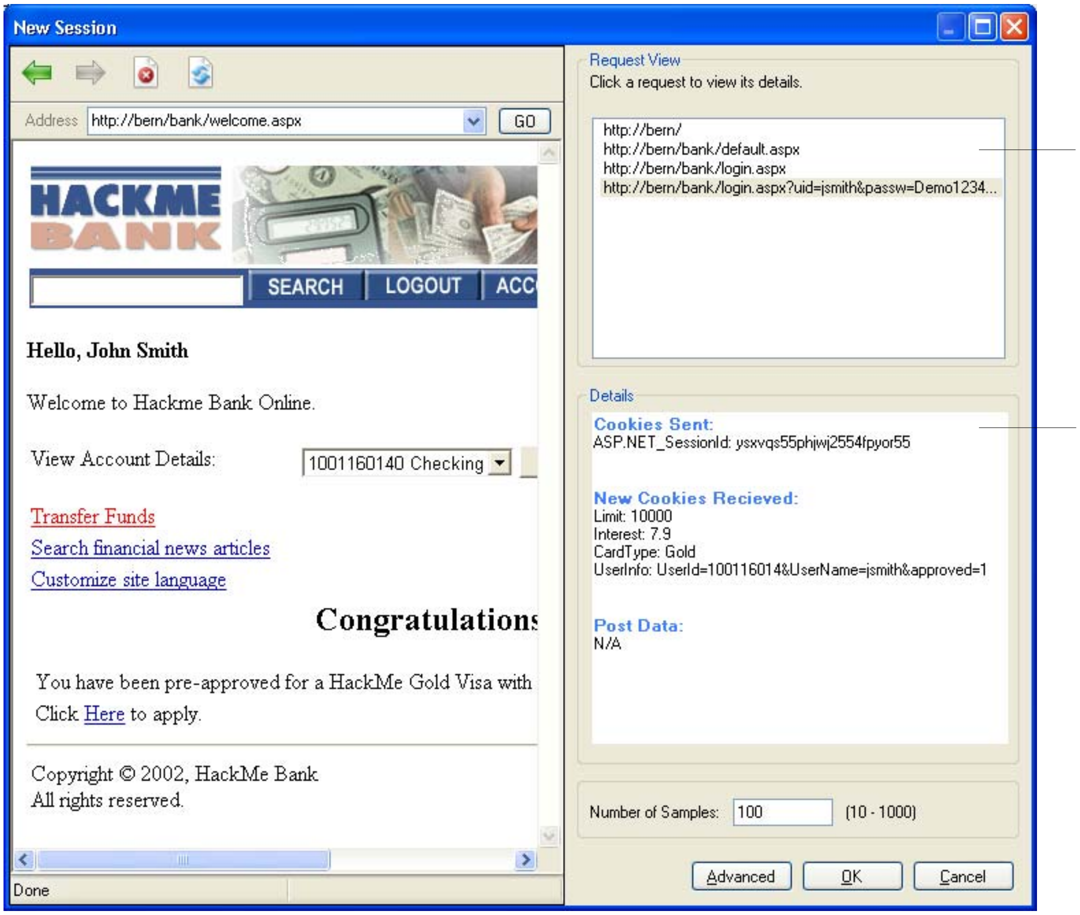
Token Analyzer connects to your web application server and downloads the page of the given URL.

2 Browse your web application, as a valid user, using typical actions.

Token Analyzer sends requests to, and receives responses from your web application server. It lists the URL requests that you sent and URLs to which Token Analyzer was redirected by the server.

Select Relevant Requests

While you explored your web application, Token Analyzer saved each request. You see that a number of URLs appear in the **Request View** list. To decide which requests are relevant for Token Analyzer, select a request in the **Request View** list and check the **Details** list.



Address http://bern/bank/welcome.aspx

Hello, John Smith

Welcome to Hackme Bank Online.

View Account Details: 1001160140 Checking

Transfer Funds

Search financial news articles

Customize site language

Congratulations

You have been pre-approved for a HackMe Gold Visa with

Click Here to apply.

Copyright © 2002, HackMe Bank
All rights reserved.

Click a request to view its details.

http://bern/
http://bern/bank/default.aspx
http://bern/bank/login.aspx
http://bern/bank/login.aspx?uid=jsmith&passw=Demo1234...

Cookies Sent:
ASP.NET_SessionId: ysxvqs55phjwj2554fpyor55

New Cookies Recieved:
Limit: 10000
Interest: 7.9
CardType: Gold
UserInfo: UserId=100116014&UserName=jsmith&approved=1

Post Data:
N/A

Number of Samples: 100 (10 - 1000)

Request view: requests sent during your explore

Details: token sent by client, tokens sent by server; post data

The **Details** list shows:

- cookie and URL parameter values that Token Analyzer sent to your application server
- new cookie and URL parameter values that Token Analyzer received from your application server
- post data the Token Analyzer sent to your application server in the body of the request

For example, you select one of the requests and it shows that your username and password were given to the server in the post data. As this request also shows, under **Cookies Received**, that it received a SessionID parameter from the server, you realize that this request is relevant for testing.

Collect Session Token Samples

⇒ After selecting a relevant request from **New Session > Request View**, click **OK**.

The **New Session** window closes. The main window appears, with a progress window over it. The progress window shows: `Collecting Session Data`

The higher the **Number of Samples** value, the longer this process takes, as this process is Token Analyzer re-performing the your site exploration and collecting the session tokens and their values that it sends and receives.

Sample Session Tokens and Subsets

The session tokens that were collected from the request that you selected in the **New Session** window are provided in the **Session Tokens** list on the main window, each by name and type (cookie or URL).

⇒ Select a session token from the list.

The **Samples View** changes when you select a different token. The **Samples View** shows each character of the session token in a separate cell, with a Character Set Classification label.

Character Set Classification

The first pattern that a malicious user would look for is: what type of input is expected? Some session tokens use only alphanumeric text, some use symbols, some use hexadecimal characters, and so on. The first evaluation that Token Analyzer makes is on each character in the session token, based on the numerous examples that it collected, classifying it as one of the following:

Class	Label	Description
Number	N	0123456789
Alphabetic	A	abcdefghijklmnopqrstuvwxyz (case insensitive)
Text	T	abcdefghijklmnopqrstuvwxyz0123456789 (case insensitive)
Base64	B	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
Hexadecimal	H	0123456789abcdef (case insensitive)
Delimiter	D	if a column includes only characters that are not in the previous sets
Other	O	if a column includes characters from Base64 and from Delimiter sets

For example:

- If the first character of the first collected value of a cookie is “w”, it might be Alphabetic, Text, Base64, Hexadecimal, or Other.
- If the first character in the fifteenth collected value is “2”, the character is either Text or Base64.

- If “+” or “/” do not appear in any of the collected samples, the character is classified as Text. However, if the 104th sample has “&” as the first character, the character should be classified as Other.

This example shows how even in the first and most basic of the value analysis, having more samples can mean the difference between true or faulty evaluations.

Subsets

Session tokens are often comprised of subsets. Each subset may be created by a different token-creator algorithm. In this case, it would not be relevant to analyze the session token as a whole; you want the tests to show whether each algorithm is producing random values. Therefore, Token Analyzer allows you select a subset of columns before running the cruncher tests.

You could select subsets at random, to see if the test results are interesting. You could select subsets of similar character classifications. Or you could examine the session token of the selected session and attempt to view it as a malicious user might; where does one subset seem to end and the next begin?

For example, delimiters are already marked for you in the character classification. You, or a malicious user, could choose the characters between delimiters as an educated guess that each subset will be separated by symbols that are different from the valid values of the subsets.

To select a subset:

1 Select a session token from the **Session Tokens** list.

2 Do one of the following:

- To select consecutive columns, drag the mouse across the column headers you want.
- To select non-consecutive columns, use the Ctrl or Shift keys.
- To select the entire session token, click **Select All**.

Testing Session Tokens

After you have selected a subset of columns in the Samples View, you can initiate analysis of the session token.

1 Click **Analyze**.

Token Analyzer runs the collected data through the tests.

2 Open the tab of the tests that are relevant for the valid values of the selected session token.

The results of the specific test are displayed in its tab.

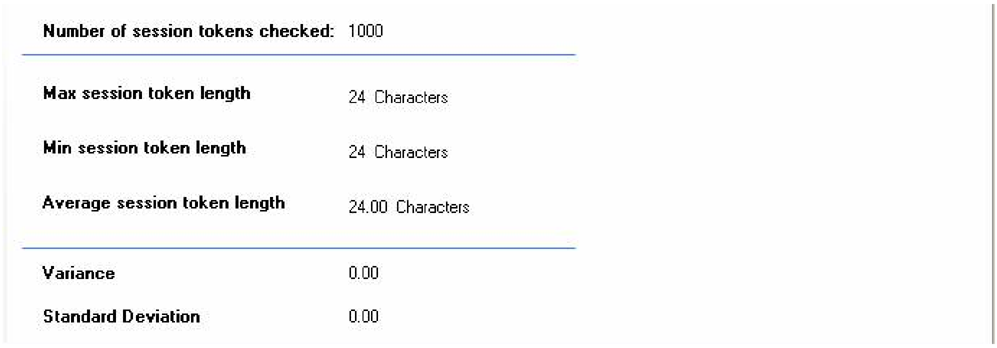
Each of the tests explained in the following topics has a different type of result and is relevant for different types of session token values.

Size Analysis

Token Analyzer provides statistics on the size of the session token: shorter values are easier to guess. Secure session tokens are long. The desired length of a token depends on the character set classification. For example, a value made of Number-type characters should be at least twelve-digits long.

If a session token, or a discernible subset, is too short, the entire session token can be more easily predicted.

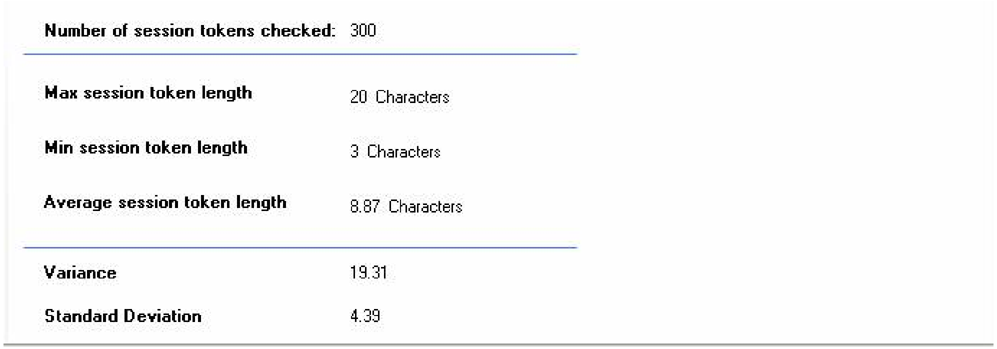
This test also provides the standard deviation and variance values, which describe the spread of session token lengths around the average length (0.00 indicates that all selected session tokens are the same length).



Number of session tokens checked:	1000
Max session token length	24 Characters
Min session token length	24 Characters
Average session token length	24.00 Characters
Variance	0.00
Standard Deviation	0.00

Figure 1 Secure Size

This token is strong, because it must always be a secure length of 24 characters.



Number of session tokens checked:	300
Max session token length	20 Characters
Min session token length	3 Characters
Average session token length	8.87 Characters
Variance	19.31
Standard Deviation	4.39

Figure 2 Weak Size

Although the maximum size of this sample session token is 20 characters, which is a secure length in most cases; the average size, less than nine characters, makes this token more often too weak. The minimum sample of three characters is a definite security issue.

Singles Count

This test counts the appearances of each character.

A secure session token will use as much of the available characters in a character set as possible, and no character will appear more often than others.

A weak session token will have a less random character distribution. For example: it might use only a few of the possible characters of a character set; or characters will appear so much more often than others, that they can be guessed with a certain probability, which makes the entire token easier to guess.

The graph shows each selected character on the X axis, and the number of times that it appeared on the Y axis. The horizontal line cutting the graph is the number of appearances that are expected of each character, assuming random probability.

The graph indicates that a session token is weak when:

- a bar crosses the Expected line by a meaningful amount
- a few characters that cross the line (even slightly) have some common characteristics (for example, close to one another in the character set).
- a character does not appear at all, but is part of the character set

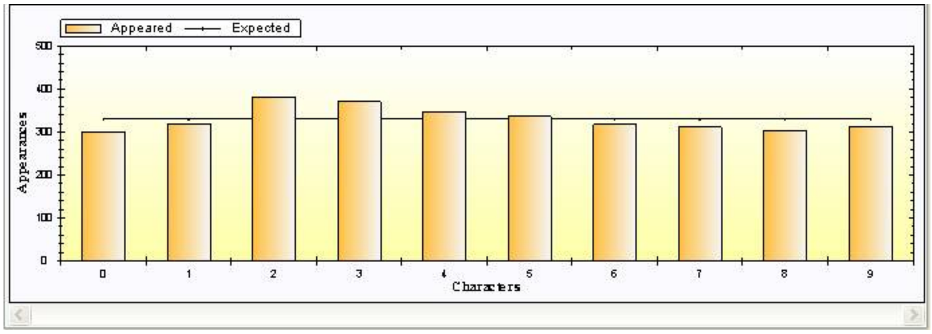


Figure 3 Secure Number of Appearances of Single Characters

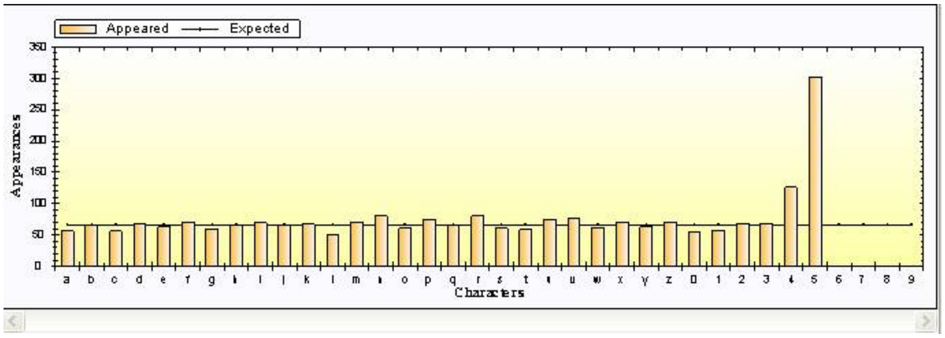


Figure 4 Weak Session Token Discovered in Singles Count

In the weak sample, the character "4" appears much more often than would be expected in a random token; and the character "5" appears so often that it may be predicted. Also notice that character 6 through 9 do not appear. This could indicate either that the session token does not use a random selection of the full character set, or that the initial analysis of which character set is used is faulty.

Pairs Count

This test counts the appearances of overlapping character pairs in each session token.

A secure session token will show a random distribution in the Pairs Count: using all available characters in random pairs, with no pair appearing more often than others.

A weak session token will have discernible patterns. For example, a specific character always appearing after another character, pairs missing from the results, or pairs that appear more often than others.

The Pairs Count test provides different views for its results: **Graph** and **Table**.

The Pairs Count graph has the first character on the X axis, the second character on the Y axis (you might have to scroll upward to see the entire character set), and the meeting point color indicates randomness of the count. The color is blue if the count of this pair is random, in relation to the other pairs; it is red if this pair appears much more often than would be expected in a random distribution. The tooltip of any meeting point lists the first character, second character, and count of the pair.

The Pairs Count table lists the first character, the second character, number of appearances that this pair was counted, and the expected number of appearances in a random distribution.

If the Singles Count results in one or two characters appearing much more than any other, try the Pairs Count. The repeated characters could be a sign of doubling.

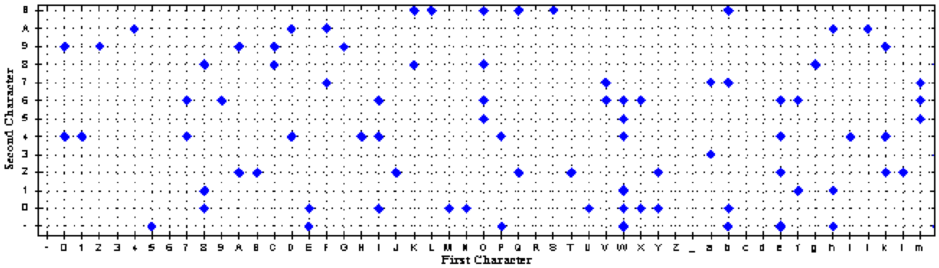


Figure 5 Secure Pairs Count

In this sample, the first thing you can see is that all pairs are colored in blue, which shows that the pairs counts are not above the expected number of appearances. Also notice that every character is used, and that there is no discernible pattern.

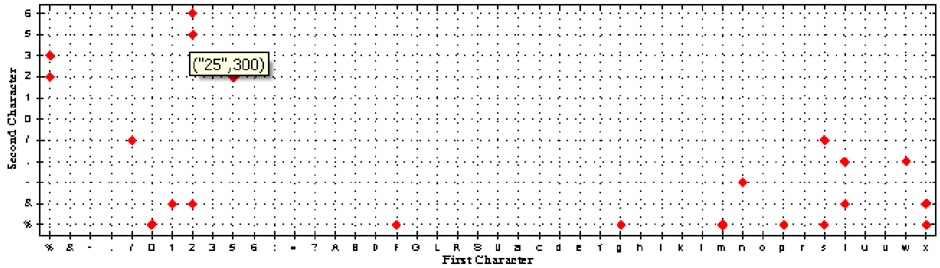


Figure 6 Pairs Count Shows Weak Session Token

In this sample, the red points on the graph show that the characters are predictable, because they appear more often than a random distribution could account for.

Correlation

The Correlation chart shows the value of a session token on the Y axis, in correlation to its index on the X axis.

Each row of a selected subset is given a numeric value based on its character set, and this value is the Y axis. The X axis is a Token Analyzer index number given to each session token as it is collected.

A secure session token will show no correlation between numeric value and time (index value). The points on the graph will be scattered.

A weak session token will have a correlation with points that appear in a pattern that is easier to see. For example, if your token-creator algorithm provides the value of a session token subset using an incremental function, the points of the Correlation test will show a discernible slope when this subset is tested.

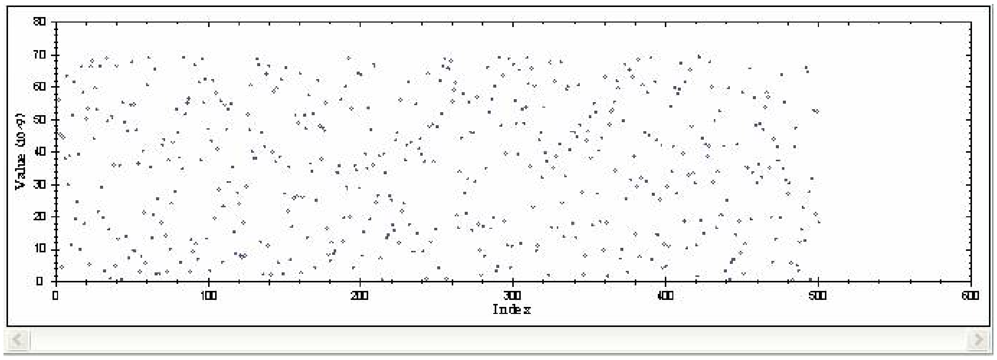


Figure 7 Secure Correlation

In a correlation analysis of a secure session token, random scattering of index-to-value correlation shows secure unpredictability.

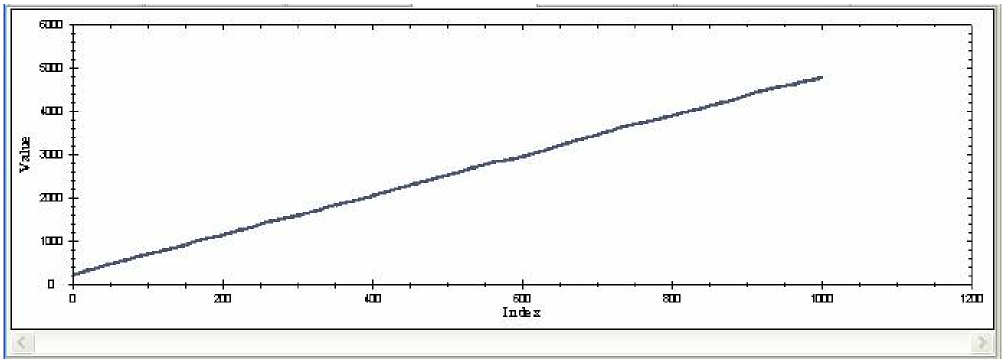


Figure 8 Correlation Shows Weak Token Values

A discernible pattern shows a weak session token.

This sample is especially predictable, as the pattern shows a non-random increment in token value.

String Search

Some servers create unique session tokens by implanting user information and credentials in the session token. This makes for easier-to-guess session tokens, also making it possible for a malicious user to obtain confidential user information. Often servers attempt to overcome this by basing a transformation algorithm on the user data. If the transformation is not complex enough and can be guessed or reverse-engineered, a malicious user is still able to hijack session tokens and obtain confidential information that is supposedly hidden by the transformation.

A secure session token will result with no dictionary or user input strings.

A weak session token will result in strings that appear under easy-to-guess transformations.

The String Search test runs automatic searches for commonly found strings and for strings in the GET and POST requests. If the results show that some value-string was found to appear, other than common strings such as **user** or **passwd**, it is because Token Analyzer found name-value pairs in the GET or POST data and performed the transformation tests on this data.

If you suspect that your session tokens are displaying specific strings, you can run a search for any string you choose.

1 Enter the string in the **Manual Search** text box and then click **Go**.

If the string is found, it is listed in the left-hand panel, with the number of appearances. The right-hand panel lists the rows of samples that contain the string.

2 To see where in the session token this string is found, click **View in Grid**.

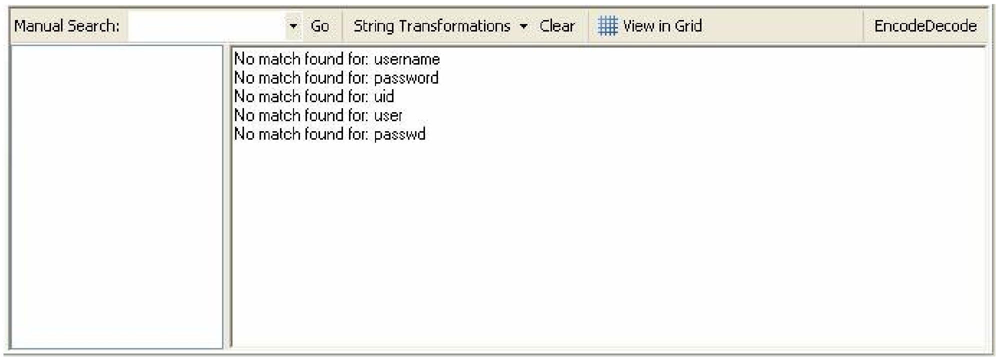
The **Session Data Grid** window opens. This window highlights the string in the samples where it is found, from within the selected subset.

The String Search test also searches for strings based on reverse transformations of different combinations of Unicode, MD5, Ascii, SHA1, Hex, and Base64; in either the user input or the session data, or both.

For example, a result of the String Search test might display the following:

```
The following transformations were done on the input string:
1. The string was transformed to bytes, assuming Ascii encoding.
2. The bytes were transformed back to string assuming Hex encoding.
```

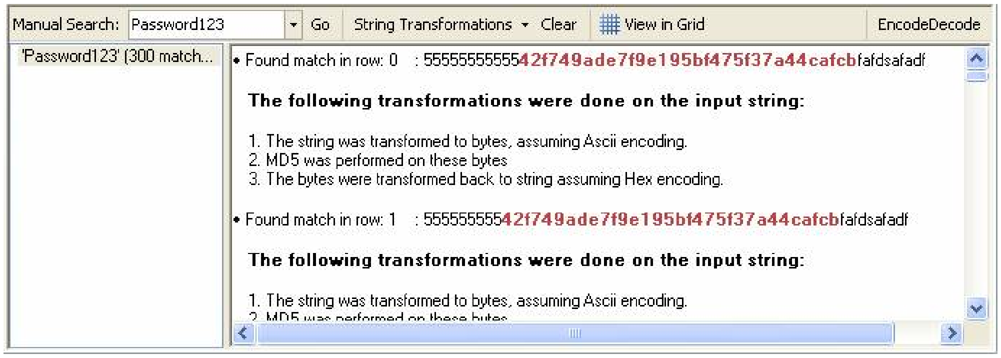
To see the list of transformation combinations that is used, click **String Transformations**.



```
No match found for: username
No match found for: password
No match found for: uid
No match found for: user
No match found for: passwd
```

Figure 9 Secure String Search

A secure session token does not contain any of the commonly used strings that malicious users exploit.



- Found match in row: 0 : 5555555555542f749ade7f9e195bf475f37a44cafcbfafdsafadf

The following transformations were done on the input string:

1. The string was transformed to bytes, assuming Ascii encoding.
2. MD5 was performed on these bytes
3. The bytes were transformed back to string assuming Hex encoding.

- Found match in row: 1 : 55555555542f749ade7f9e195bf475f37a44cafcbfafdsafadf

The following transformations were done on the input string:

1. The string was transformed to bytes, assuming Ascii encoding.
2. MD5 was performed on these bytes

Figure 10 Strings Found in Weak Session Token

The session token shown in this sample shows that Token Analyzer was able to transform the user input (string to bytes using Ascii, MD5, and then bytes to string using Hex) from an apparently random set of characters into the original string that contained the user's password.

Note: You can access the Watchfire EncodeDecode PowerTool from this test, to run more transformation samples, in various formats: click **EncodeDecode**.

Column Chi-Square

This test checks the selected columns using the Chi-Square method, which finds whether each column, over the number of samples taken, is random or not. In this test, each selected column (column number is X axis) is given a color-coded bar to show randomness levels. The higher the score (chi-square score is Y axis), the further from random the column is.

A secure session token will have all light-blue, low-scored columns.

A weak session token will have some or all columns that go into dark blue. If a column reaches red, the character is dangerously non-random. If there are multiple red-topped columns, the session token itself might be predictable.

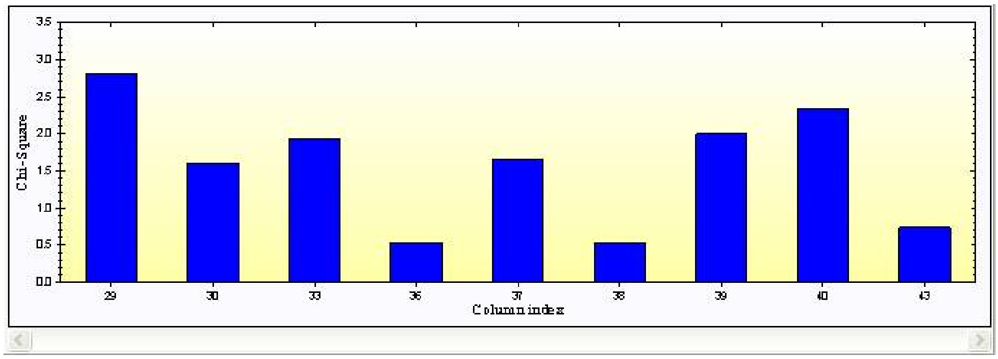


Figure 11 Secure Chi-Square

In a secure session token, the Chi-Square results are all blue, with low scores indicating secure random distribution.

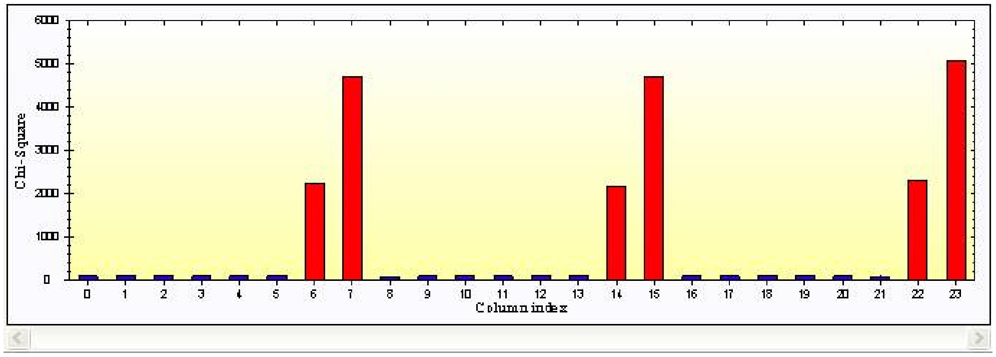


Figure 12 Chi-Square Discovers Weakness in Specific Characters

In this sample of a weak session token, the 6th, 7th, 14th, 15th, 22nd, and 23rd columns of the sample represent characters that are given high non-random scores. The security issue is emphasized by the red coloring of the columns.

Note that the Chi-Square test results are dependent upon the character set:

- Columns that are classified as Delimiters will always be given a score of 0.
- Columns of different classifications will have different score ranges; for example, the scores of a Base64 column will be higher than the scores of an Alphabetic column, even if the characters appear the same number of times.

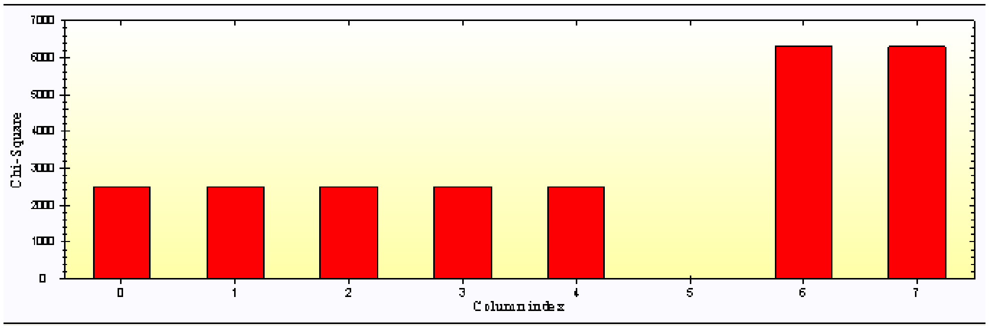


Figure 13 Chi-Square of "https://"

In this sample of a URL subset, all rows contain the same characters, so all columns should show that the values are non-random.

- Column 6 is for ":", which is a delimiter; therefore, its character set contains only ":" and its value is 0.
- The first five columns are Alphabetic and the last two are Base64; notice the difference in their non-random scores, which is due to different character set lengths.

Getting Better Results

If you are not convinced that the results of the tests are accurate, you can re-collect data, without returning to the embedded browser.

1 In the main window, increase the value given in the **Number of Samples** text box.

2 Click **Update**.

Token Analyzer re-performs the actions that you modeled in the embedded browser, for as many times as you entered in the **Number of Samples** text box. The new data is added to the previous collection and the **Sample View** is updated.

3 Click **Analyze** to retest the new data.

For any test that provides interesting results, you can check that the results are persistent, by updating the samples. If the results in the new analysis are similar to the previous results, and if the number of samples is large enough, you can assume that the results are persistent.

Index